

1 INTERNET-DRAFT  
2 draft-ietf-ipp-model-v11-054.txt

3 R. deBry  
4 Utah Valley State College  
5 T. Hastings (editor)  
6 Xerox Corporation  
7 R. Herriot  
8 Xerox Corporation  
9 S. Isaacson  
10 Novell, Inc.  
11 P. Powell  
12 Astart Technologies  
13 ~~June 23, 1999~~ February 23, 2000

14  
15 Internet Printing Protocol/1.1: Model and Semantics  
16 Copyright (C) The Internet Society (~~1999~~2000). 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 [RFC2616]. It defines the  
56 encoding rules for a new Internet MIME media type called "application/ipp". This document also defines  
57 the rules for transporting over HTTP a message body whose Content-Type is "application/ipp". This  
58 document defines a new scheme named 'ipp' for identifying IPP printers and jobs.

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

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

66

**Table of Contents**

67	1. Introduction .....	10
68	1.1 Simplified Printing Model .....	11
69	2. IPP Objects .....	13
70	2.1 Printer Object .....	13
71	2.2 Job Object .....	16
72	2.3 Object Relationships .....	17
73	2.4 Object Identity .....	18
74	3. IPP Operations .....	20
75	3.1 Common Semantics .....	21
76	3.1.1 Required Parameters .....	21
77	3.1.2 Operation IDs and Request IDs .....	22
78	3.1.3 Attributes .....	22
79	3.1.4 Character Set and Natural Language Operation Attributes .....	23
80	3.1.4.1 Request Operation Attributes .....	24
81	3.1.4.2 Response Operation Attributes .....	27
82	3.1.5 Operation Targets .....	28
83	3.1.6 Operation Response Status Codes and Status Messages .....	29
84	3.1.6.1 "status-code" (type2 enum) .....	29
85	3.1.6.2 "status-message" (text(255)) .....	30
86	3.1.6.3 "detailed-status-message" (text(MAX)) .....	30
87	3.1.6.4 "document-access-error" (text(MAX)) .....	30
88	3.1.7 Unsupported Attributes .....	31
89	3.1.8 Versions .....	32
90	3.1.9 Job Creation Operations .....	34
91	3.2 Printer Operations .....	36
92	3.2.1 Print-Job Operation .....	36
93	3.2.1.1 Print-Job Request .....	36
94	3.2.1.2 Print-Job Response .....	40
95	3.2.2 Print-URI Operation .....	41
96	3.2.3 Validate-Job Operation .....	42
97	3.2.4 Create-Job Operation .....	42
98	3.2.5 Get-Printer-Attributes Operation .....	43
99	3.2.5.1 Get-Printer-Attributes Request .....	44
100	3.2.5.2 Get-Printer-Attributes Response .....	45
101	3.2.6 Get-Jobs Operation .....	46
102	3.2.6.1 Get-Jobs Request .....	46
103	3.2.6.2 Get-Jobs Response .....	48
104	3.2.7 Pause-Printer Operation .....	49
105	3.2.7.1 Pause-Printer Request .....	50
106	3.2.7.2 Pause-Printer Response .....	50
107	3.2.8 Resume-Printer Operation .....	51

108	3.2.9	Purge-Jobs Operation.....	51
109	3.3	Job Operations .....	52
110	3.3.1	Send-Document Operation.....	52
111	3.3.1.1	Send-Document Request .....	53
112	3.3.1.2	Send-Document Response.....	55
113	3.3.2	Send-URI Operation .....	55
114	3.3.3	Cancel-Job Operation.....	55
115	3.3.3.1	Cancel-Job Request .....	56
116	3.3.3.2	Cancel-Job Response.....	57
117	3.3.4	Get-Job-Attributes Operation.....	57
118	3.3.4.1	Get-Job-Attributes Request .....	58
119	3.3.4.2	Get-Job-Attributes Response.....	58
120	3.3.5	Hold-Job Operation.....	59
121	3.3.5.1	Hold-Job Request .....	60
122	3.3.5.2	Hold-Job Response.....	61
123	3.3.6	Release-Job Operation .....	61
124	3.3.7	Restart-Job Operation .....	62
125	3.3.7.1	Restart-Job Request.....	63
126	3.3.7.2	Restart-Job Response .....	64
127	4.	Object Attributes .....	64
128	4.1	Attribute Syntaxes.....	65
129	4.1.1	'text'.....	65
130	4.1.1.1	'textWithoutLanguage'.....	66
131	4.1.1.2	'textWithLanguage'.....	66
132	4.1.2	'name'.....	67
133	4.1.2.1	'nameWithoutLanguage'.....	67
134	4.1.2.2	'nameWithLanguage' .....	68
135	4.1.2.3	Matching 'name' attribute values .....	68
136	4.1.3	'keyword' .....	69
137	4.1.4	'enum' .....	70
138	4.1.5	'uri'.....	70
139	4.1.6	'uriScheme' .....	70
140	4.1.7	'charset'.....	71
141	4.1.8	'naturalLanguage'.....	71
142	4.1.9	'mimeType'.....	72
143	4.1.9.1	Application/octet-stream -- Auto-Sensing the document format .....	72
144	4.1.10	'octetString'.....	73
145	4.1.11	'boolean'.....	73
146	4.1.12	'integer' .....	73
147	4.1.13	'rangeOfInteger'.....	73
148	4.1.14	'dateTime'.....	74
149	4.1.15	'resolution'.....	74
150	4.1.16	'1setOf X'.....	74
151	4.2	Job Template Attributes.....	74

152	4.2.1	job-priority (integer(1:100)).....	78
153	4.2.2	job-hold-until (type3 keyword   name (MAX)).....	79
154	4.2.3	job-sheets (type3 keyword   name(MAX)).....	79
155	4.2.4	multiple-document-handling (type2 keyword).....	80
156	4.2.5	copies (integer(1:MAX)).....	81
157	4.2.6	finishings (1setOf type2 enum).....	81
158	4.2.7	page-ranges (1setOf rangeOfInteger (1:MAX)).....	83
159	4.2.8	sides (type2 keyword).....	84
160	4.2.9	number-up (integer(1:MAX)).....	84
161	4.2.10	orientation-requested (type2 enum).....	85
162	4.2.11	media (type3 keyword   name(MAX)).....	86
163	4.2.12	printer-resolution (resolution).....	86
164	4.2.13	print-quality (type2 enum).....	86
165	4.3	Job Description Attributes.....	87
166	4.3.1	job-uri (uri).....	89
167	4.3.2	job-id (integer(1:MAX)).....	89
168	4.3.3	job-printer-uri (uri).....	90
169	4.3.4	job-more-info (uri).....	90
170	4.3.5	job-name (name(MAX)).....	90
171	4.3.6	job-originating-user-name (name(MAX)).....	90
172	4.3.7	job-state (type1 enum).....	90
173	4.3.7.1	Forwarding Servers.....	93
174	4.3.7.2	Partitioning of Job States.....	93
175	4.3.8	job-state-reasons (1setOf type2 keyword).....	94
176	4.3.9	job-state-message (text(MAX)).....	97
177	4.3.10	job-detailed-status-messages (1setOf text(MAX)).....	97
178	4.3.11	job-document-access-errors (1setOf text(MAX)).....	97
179	4.3.12	number-of-documents (integer(0:MAX)).....	98
180	4.3.13	output-device-assigned (name(127)).....	98
181	4.3.14	Event Time Job Description Attributes.....	98
182	4.3.14.1	time-at-creation (integer(MIN:MAX)).....	99
183	4.3.14.2	time-at-processing (integer(MIN:MAX)).....	99
184	4.3.14.3	time-at-completed (integer(MIN:MAX)).....	99
185	4.3.14.4	job-printer-up-time (integer(1:MAX)).....	99
186	4.3.14.5	date-time-at-creation (dateTime).....	99
187	4.3.14.6	date-time-at-processing (dateTime).....	99
188	4.3.14.7	date-time-at-completed (dateTime).....	99
189	4.3.15	number-of-intervening-jobs (integer(0:MAX)).....	100
190	4.3.16	job-message-from-operator (text(127)).....	100
191	4.3.17	Job Size Attributes.....	100
192	4.3.17.1	job-k-octets (integer(0:MAX)).....	100
193	4.3.17.2	job-impressions (integer(0:MAX)).....	100
194	4.3.17.3	job-media-sheets (integer(0:MAX)).....	101
195	4.3.18	Job Progress Attributes.....	101
196	4.3.18.1	job-k-octets-processed (integer(0:MAX)).....	101

197	4.3.18.2	job-impressions-completed (integer(0:MAX)).....	101
198	4.3.18.3	job-media-sheets-completed (integer(0:MAX)).....	102
199	4.3.19	attributes-charset (charset).....	102
200	4.3.20	attributes-natural-language (naturalLanguage).....	102
201	4.4	Printer Description Attributes.....	102
202	4.4.1	printer-uri-supported (1setOf uri).....	104
203	4.4.2	uri-authentication-supported (1setOf type2 keyword).....	105
204	4.4.3	uri-security-supported (1setOf type2 keyword).....	105
205	4.4.4	printer-name (name(127)).....	106
206	4.4.5	printer-location (text(127)).....	107
207	4.4.6	printer-info (text(127)).....	107
208	4.4.7	printer-more-info (uri).....	107
209	4.4.8	printer-driver-installer (uri).....	107
210	4.4.9	printer-make-and-model (text(127)).....	107
211	4.4.10	printer-more-info-manufacturer (uri).....	107
212	4.4.11	printer-state (type1 enum).....	108
213	4.4.12	printer-state-reasons (1setOf type2 keyword).....	108
214	4.4.13	printer-state-message (text(MAX)).....	110
215	4.4.14	ipp-versions-supported (1setOf type2 keyword).....	110
216	4.4.15	operations-supported (1setOf type2 enum).....	111
217	4.4.16	multiple-document-jobs-supported (boolean).....	112
218	4.4.17	charset-configured (charset).....	112
219	4.4.18	charset-supported (1setOf charset).....	112
220	4.4.19	natural-language-configured (naturalLanguage).....	112
221	4.4.20	generated-natural-language-supported (1setOf naturalLanguage).....	113
222	4.4.21	document-format-default (mimeMediaType).....	113
223	4.4.22	document-format-supported (1setOf mimeMediaType).....	113
224	4.4.23	printer-is-accepting-jobs (boolean).....	113
225	4.4.24	queued-job-count (integer(0:MAX)).....	114
226	4.4.25	printer-message-from-operator (text(127)).....	114
227	4.4.26	color-supported (boolean).....	114
228	4.4.27	reference-uri-schemes-supported (1setOf uriScheme).....	114
229	4.4.28	pdl-override-supported (type2 keyword).....	114
230	4.4.29	printer-up-time (integer(1:MAX)).....	115
231	4.4.30	printer-current-time (dateTime).....	115
232	4.4.31	multiple-operation-time-out (integer(1:MAX)).....	116
233	4.4.32	compression-supported (1setOf type3 keyword).....	116
234	4.4.33	job-k-octets-supported (rangeOfInteger(0:MAX)).....	116
235	4.4.34	job-impressions-supported (rangeOfInteger(0:MAX)).....	116
236	4.4.35	job-media-sheets-supported (rangeOfInteger(0:MAX)).....	116
237	4.4.36	pages-per-minute (integer(0:MAX)).....	117
238	4.4.37	pages-per-minute-color (integer(0:MAX)).....	117
239	5.	Conformance.....	117
240	5.1	Client Conformance Requirements.....	117

241	5.2	IPP Object Conformance Requirements .....	119
242	5.2.1	Objects .....	119
243	5.2.2	Operations .....	119
244	5.2.3	IPP Object Attributes .....	120
245	5.2.4	Versions .....	120
246	5.2.5	Extensions .....	121
247	5.2.6	Attribute Syntaxes .....	121
248	5.2.7	Security .....	121
249	5.3	Charset and Natural Language Requirements .....	121
250	6.	IANA Considerations .....	122
251	6.1	Typed 'keyword' and 'enum' Extensions .....	122
252	6.2	Attribute Extensibility .....	124
253	6.3	Attribute Syntax Extensibility .....	125
254	6.4	Operation Extensibility .....	125
255	6.5	Attribute Group Extensibility .....	125
256	6.6	Status Code Extensibility .....	126
257	6.7	<u>Out-of-band Attribute Value Extensibility .....</u>	<u>126</u>
258	6.8	Registration of MIME types/sub-types for document-formats .....	127
259	6.9	Registration of charsets for use in 'charset' attribute values .....	127
260	7.	Internationalization Considerations .....	127
261	8.	Security Considerations .....	131
262	8.1	Security Scenarios .....	131
263	8.1.1	Client and Server in the Same Security Domain .....	131
264	8.1.2	Client and Server in Different Security Domains .....	132
265	8.1.3	Print by Reference .....	132
266	8.2	URIs in Operation, Job, and Printer attributes .....	132
267	8.3	URIs for each authentication mechanisms .....	132
268	8.4	Restricted Queries .....	133
269	8.5	Operations performed by operators and system administrators .....	133
270	8.6	Queries on jobs submitted using non-IPP protocols .....	133
271	9.	References .....	134
272	10.	Author's Address .....	138
273	11.	Formats for IPP Registration Proposals .....	141
274	11.1	Type2 keyword attribute values registration .....	141
275	11.2	Type3 keyword attribute values registration .....	141
276	11.3	Type2 enum attribute values registration .....	142
277	11.4	Type3 enum attribute values registration .....	142
278	11.5	Attribute registration .....	142
279	11.6	Attribute Syntax registration .....	143
280	11.7	Operation registration .....	143



281	11.8	Attribute Group registration.....	143
282	11.9	Status code registration.....	144
283		<u>11.10 Out-of-band Attribute Value registration.....</u>	<u>144</u>
284	12.	APPENDIX A: Terminology.....	144
285	12.1	Conformance Terminology.....	145
286	12.1.1	NEED NOT.....	145
287	12.2	Model Terminology.....	145
288	12.2.1	Keyword.....	145
289	12.2.2	Attributes.....	145
290	12.2.2.1	Attribute Name.....	145
291	12.2.2.2	Attribute Group Name.....	145
292	12.2.2.3	Attribute Value.....	146
293	12.2.2.4	Attribute Syntax.....	146
294	12.2.3	Supports.....	146
295	12.2.4	print-stream page.....	147
296	12.2.5	impression.....	148
297	13.	APPENDIX B: Status Codes and Suggested Status Code Messages.....	148
298	13.1	Status Codes.....	149
299	13.1.1	Informational.....	149
300	13.1.2	Successful Status Codes.....	149
301	13.1.2.1	successful-ok (0x0000).....	149
302	13.1.2.2	successful-ok-ignored-or-substituted-attributes (0x0001).....	149
303	13.1.2.3	successful-ok-conflicting-attributes (0x0002).....	149
304	13.1.3	Redirection Status Codes.....	150
305	13.1.4	Client Error Status Codes.....	150
306	13.1.4.1	client-error-bad-request (0x0400).....	150
307	13.1.4.2	client-error-forbidden (0x0401).....	150
308	13.1.4.3	client-error-not-authenticated (0x0402).....	150
309	13.1.4.4	client-error-not-authorized (0x0403).....	150
310	13.1.4.5	client-error-not-possible (0x0404).....	151
311	13.1.4.6	client-error-timeout (0x0405).....	151
312	13.1.4.7	client-error-not-found (0x0406).....	151
313	13.1.4.8	client-error-gone (0x0407).....	151
314	13.1.4.9	client-error-request-entity-too-large (0x0408).....	152
315	13.1.4.10	client-error-request-value-too-long (0x0409).....	152
316	13.1.4.11	client-error-document-format-not-supported (0x040A).....	152
317	13.1.4.12	client-error-attributes-or-values-not-supported (0x040B).....	152
318	13.1.4.13	client-error-uri-scheme-not-supported (0x040C).....	153
319	13.1.4.14	client-error-charset-not-supported (0x040D).....	153
320	13.1.4.15	client-error-conflicting-attributes (0x040E).....	153
321	13.1.4.16	client-error-compression-not-supported (0x040F).....	153
322	13.1.4.17	client-error-compression-error (0x0410).....	153
323	13.1.4.18	client-error-document-format-error (0x0411).....	154



324	13.1.4.19	client-error-document-access-error (0x0412).....	154
325	13.1.5	Server Error Status Codes.....	154
326	13.1.5.1	server-error-internal-error (0x0500).....	154
327	13.1.5.2	server-error-operation-not-supported (0x0501).....	154
328	13.1.5.3	server-error-service-unavailable (0x0502).....	154
329	13.1.5.4	server-error-version-not-supported (0x0503).....	155
330	13.1.5.5	server-error-device-error (0x0504).....	155
331	13.1.5.6	server-error-temporary-error (0x0505).....	155
332	13.1.5.7	server-error-not-accepting-jobs (0x0506).....	156
333	13.1.5.8	server-error-busy (0x0507).....	156
334	13.1.5.9	server-error-job-canceled (0x0508).....	156
335	13.1.5.10	server-error-multiple-document-jobs-not-supported (0x0509).....	156
336	13.2	Status Codes for IPP Operations.....	157
337	14.	APPENDIX C: "media" keyword values.....	159
338	15.	APPENDIX D: Processing IPP Attributes.....	177
339	15.1	Fidelity.....	178
340	15.2	Page Description Language (PDL) Override.....	179
341	15.3	Using Job Template Attributes During Document Processing.....	180
342	16.	APPENDIX E: Generic Directory Schema.....	181
343	17.	APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model and Semantics" Documents.....	184
344	18.	Full Copyright Statement.....	189
345			

## 346 1. Introduction

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

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

357

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

360 This document is laid out as follows:

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

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

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

## 387 **1.1 Simplified Printing Model**

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

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

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

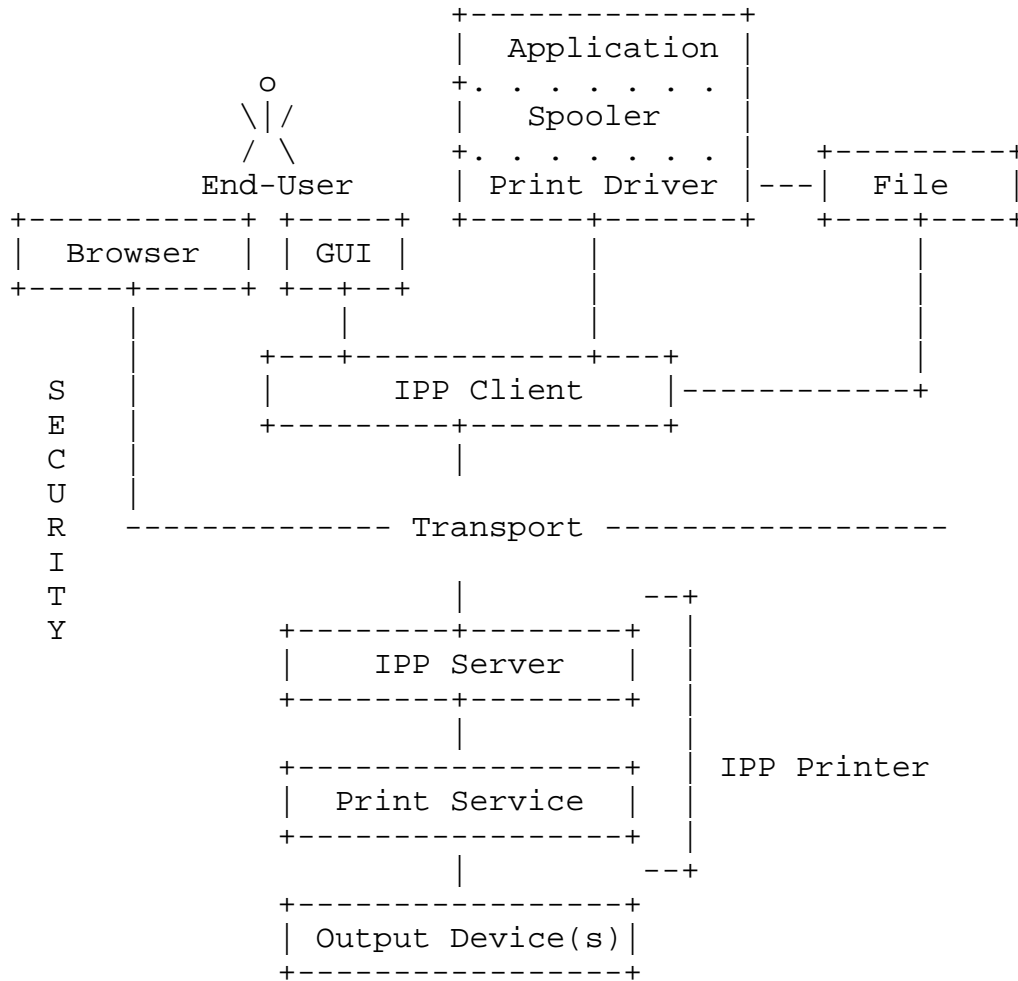
405

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

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

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  
441

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



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

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

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

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

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

## 472 2. IPP Objects

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

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

- 482 - "REQUIRED": each object MUST support the attribute.
  - 483 - "RECOMMENDED": each object SHOULD support the attribute.
  - 484 - "OPTIONAL": each object MAY support the attribute.
- 485

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

### 489 2.1 Printer Object

490 The major component of the IPP/1.1 model is the Printer object. A Printer object implements the server-  
491 side of the IPP/1.1 protocol. Using the protocol, end users may query the attributes of the Printer object and

492 submit print jobs to the Printer object. The actual implementation components behind the Printer  
493 abstraction may take on different forms and different configurations. However, the model abstraction  
494 allows the details of the configuration of real components to remain opaque to the end user. Section 3  
495 describes each of the Printer operations in detail.

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

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

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

506 Some examples of configurations supporting a Printer object include:

- 507 1) An output device with no spooling capabilities
- 508 2) An output device with a built-in spooler
- 509 3) A print server supporting IPP with one or more associated output devices
  - 510 3a) The associated output devices may or may not be capable of spooling jobs
  - 511 3b) The associated output devices may or may not support IPP

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

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

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

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

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





528 Legend:

529

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

534

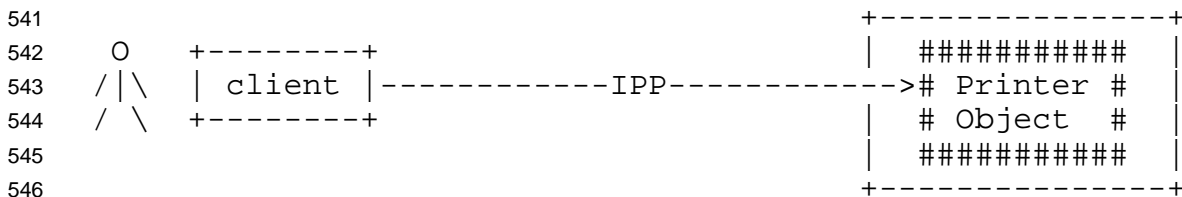
535 any indicates any network protocol or direct  
536 connect, including IPP

537

538

539 embedded printer:

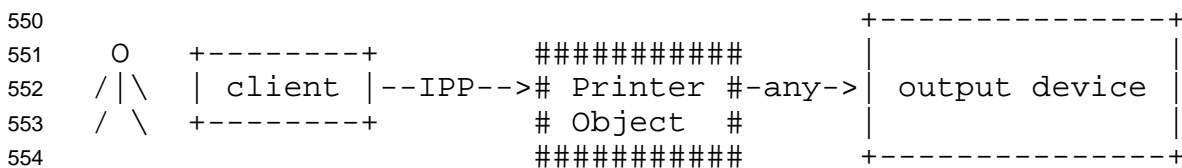
540



547

548

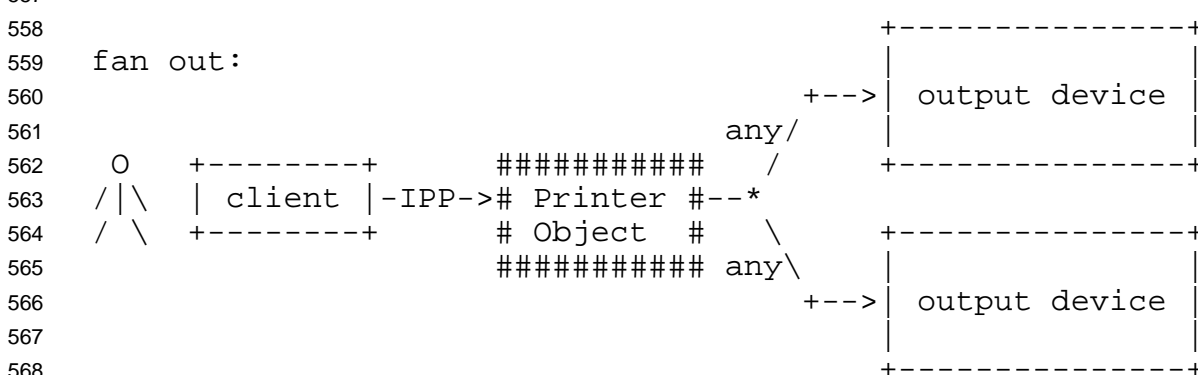
549 hosted printer:



555

556

557



567

568

569

570

## 571 2.2 Job Object

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

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

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

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

583

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

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

589

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

### 593 **2.3 Object Relationships**

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

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

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

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

## 609 2.4 Object Identity

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

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

623 If a Printer object supports more than one communication channel, some or all of those channels might  
624 support and/or require different security mechanisms. In such cases, an administrator could expose the  
625 simultaneous support for these multiple communication channels as multiple URIs for a single Printer  
626 object where each URI represents one of the communication channels to the Printer object. To support this  
627 flexibility, the IPP Printer object type defines a multi-valued identification attribute called the "printer-uri-  
628 supported" attribute. It MUST contain at least one URI. It MAY contain more than one URI. That is,  
629 every Printer object will have at least one URI that identifies at least one communication channel to the  
630 Printer object, but it may have more than one URI where each URI identifies a different communication  
631 channel to the Printer object. The "printer-uri-supported" attribute has two companion attributes, the "uri-  
632 security-supported" attribute and the "uri-authentication-supported". Both have the same cardinality as  
633 "printer-uri-supported". The purpose of the "uri-security-supported" attribute is to indicate the security  
634 mechanisms (if any) used for each URI listed in "printer-uri-supported". The purpose of the "uri-  
635 authentication-supported" attribute is to indicate the authentication mechanisms (if any) used for each URI  
636 listed in "printer-uri-supported". These three attributes are fully described in sections 4.4.1, 4.4.2, and  
637 4.4.3.

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

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

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

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

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

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

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

687 To summarize:

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

### 711 3. IPP Operations

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

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

724 The IPP/1.1 Printer operations are:

- 725 Print-Job (section 3.2.1)  
726 Print-URI (section 3.2.2)

727 Validate-Job (section 3.2.3)  
728 Create-Job (section 3.2.4)  
729 Get-Printer-Attributes (section 3.2.5)  
730 Get-Jobs (section 3.2.6)  
731 Pause-Printer (section 3.3.5)  
732 Resume-Printer (section 3.3.6)  
733 Purge-Jobs (section 3.3.7)  
734

735 The Job operations are:

736 Send-Document (section 3.3.1)  
737 Send-URI (section 3.3.2)  
738 Cancel-Job (section 3.3.3)  
739 Get-Job-Attributes (section 3.3.4)  
740 Hold-Job (section 3.3.5)  
741 Release-Job (section 3.3.6)  
742 Restart-Job (section 3.3.7)  
743

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

## 746 **3.1 Common Semantics**

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

### 749 **3.1.1 Required Parameters**

750 Every operation request contains the following REQUIRED parameters:

- 751 - a "version-number",
  - 752 - an "operation-id",
  - 753 - a "request-id", and
  - 754 - the attributes that are REQUIRED for that type of request.
- 755

756 Every operation response contains the following REQUIRED parameters:

- 757 - a "version-number",
  - 758 - a "status-code",
  - 759 - the "request-id" that was supplied in the corresponding request, and
  - 760 - the attributes that are REQUIRED for that type of response.
- 761

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



### 765 3.1.2 Operation IDs and Request IDs

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

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

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

### 781 3.1.3 Attributes

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

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



805 - Unsupported Attributes: In a create request, the client supplies a set of Operation and Job Template  
806 attributes. If any of these attributes or their values is unsupported by the Printer object, the Printer  
807 object returns the set of unsupported attributes in the response. Sections 3.1.7, 3.2.1.2, and 15 give  
808 a full description of how Job Template attributes supplied by the client in a create request are  
809 processed by the Printer object and how unsupported attributes are returned to the client. Because  
810 of extensibility, any IPP object might receive a request that contains new or unknown attributes or  
811 values for which it has no support. In such cases, the IPP object processes what it can and returns  
812 the unsupported attributes in the response. The Unsupported Attribute group is defined for all  
813 operation responses for returning unsupported attributes that the client supplied in the request.  
814

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

818 The attributes within a group MUST be unique; if an attribute with the same name occurs more than once,  
819 the group is mal-formed. Clients MUST NOT submit such malformed requests and Printers MUST NOT  
820 return such malformed responses. If such a malformed request is submitted to a Printer, the Printer MUST  
821 either (1) reject the request with the 'client-error-bad-request' status code (see section 13.1.4.1) or (2)  
822 process the request normally after selecting only one of the attribute instances, depending on  
823 implementation. Which attribute is selected when there are duplicate attributes depends on implementation.  
824 The IPP Printer MUST NOT use the values from more than one such duplicate attribute instance.

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

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

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

### 841 3.1.4 Character Set and Natural Language Operation Attributes

842 Some Job and Printer attributes have values that are text strings and names intended for human  
843 understanding rather than machine understanding (see the 'text' and 'name' attribute syntax descriptions in

844 section 4.1). The following sections describe two special Operation Attributes called "attributes-charset"  
845 and "attributes-natural-language". These attributes are always part of the Operation Attributes group. For  
846 most attribute groups, the order of the attributes within the group is not important. However, for these two  
847 attributes within the Operation Attributes group, the order is critical. The "attributes-charset" attribute  
848 MUST be the first attribute in the group and the "attributes-natural-language" attribute MUST be the second  
849 attribute in the group. In other words, these attributes MUST be supplied in every IPP request and  
850 response, they MUST come first in the group, and MUST come in the specified order. For job creation  
851 operations, the IPP Printer implementation saves these two attributes with the new Job object as Job  
852 Description attributes. For the sake of brevity in this document, these operation attribute descriptions are  
853 not repeated with every operation request and response, but have a reference back to this section instead.

#### 854 3.1.4.1 Request Operation Attributes

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

857 "attributes-charset" (charset):

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

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

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

882  
883 See the 'charset' attribute syntax description in Section 4.1.7 for the syntax and semantic  
884 interpretation of the values of this attribute and for example values.

885

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

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

891

892 There are no REQUIRED natural languages required for the Printer object to support. However, the  
893 Printer object's "generated-natural-language-supported" attribute identifies the natural languages  
894 supported by the Printer object and any contained Job objects for all text strings generated by the  
895 IPP object. A client MAY query this attribute to determine which natural language(s) are supported  
896 for generated messages.

897

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

904

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

911

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

918

919 The IPP object MUST accept any natural language and any Natural Language Override, whether the  
920 IPP object supports that natural language or not (and independent of the value of the "ipp-attribute-  
921 fidelity" Operation attribute). That is the IPP object accepts all client supplied values no matter  
922 what the values are in the Printer object's "generated-natural-language-supported" attribute. That  
923 attribute, "generated-natural-language-supported", only applies to generated messages, not client  
924 supplied messages. The IPP object MUST remember that natural language for all client-supplied  
925 attributes, and when returning those attributes in response to a query, the IPP object MUST indicate  
926 that natural language.

927

928 Each value whose attribute syntax type is 'text' or 'name' (see sections 4.1.1 and 4.1.2) has an  
929 Associated Natural-Language. This document does not specify how this association is stored in a

930 Printer or Job object. When such a value is encoded in a request or response, the natural language is  
931 either implicit or explicit:

- 932
- 933 – In the implicit case, the value contains only the text/name value, and the language is  
934 specified by the "attributes-natural-language" operation attribute in the request or response  
935 (see sections 4.1.1.1 textWithoutLanguage and 4.1.2.1 nameWithoutLanguage).
  - 936
  - 937 – In the explicit case (also known as the Natural-Language Override case), the value contains  
938 both the language and the text/name value (see sections 4.1.1.2 textWithLanguage and  
939 4.1.2.2 nameWithLanguage).

940

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

951

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

954

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

957

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

- 964
- 965 - In a create request: If the client supplies a text or name attribute (for example, the "job-name"  
966 operation attribute) that uses an apparently incompatible combination, it is a client choice that does  
967 not affect the Printer object or its correct operation. Therefore, the Printer object simply accepts the  
968 client supplied value, stores it with the Job object, and responds back with the same combination  
969 whenever the client (or any client) queries for that attribute.
  - 970 - In a query-type operation, like Get-Printer-Attributes: If the client requests an apparently incompatible  
971 combination, the Printer object responds (as described in section 3.1.4.2) using the Printer's  
972 configured natural language rather than the natural language requested by the client.

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

#### 978 **3.1.4.2 Response Operation Attributes**

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

981 "attributes-charset" (charset):

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

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

997  
998 Whether an implementation that supports more than one charset stores the data in the charset  
999 supplied by the client or code converts to one of the other supported charsets, depends on  
1000 implementation. The strategy should try to minimize loss of information during code conversion.  
1001 On each response, such an implementation converts from its internal charset to that requested.

1002  
1003 "attributes-natural-language" (naturalLanguage):

1004 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that  
1005 the IPP object is returning in this response. Unlike the "attributes-charset" operation attribute, the  
1006 IPP object **NEED NOT** return the same value as that supplied by the client in the request. The IPP  
1007 object **MAY** return the natural language of the Job object or the Printer's configured natural  
1008 language as identified by the Printer object's "natural-language-configured" attribute, rather than the  
1009 natural language supplied by the client. For any 'text' or 'name' attribute or status message in the  
1010 response that is in a different natural language than the value returned in the "attributes-natural-  
1011 language" operation attribute, the IPP object **MUST** use the Natural Language Override mechanism  
1012 (see sections 4.1.1.2 and 4.1.2.2) on each attribute value returned. The IPP object **MAY** use the  
1013 Natural Language Override mechanism redundantly, i.e., use it even when the value is in the same

1014 natural language as the value supplied in the "attributes-natural-language" operation attribute of the  
1015 response.

### 1016 **3.1.5 Operation Targets**

1017 All IPP operations are directed at IPP objects. For Printer operations, the operation is always directed at a  
1018 Printer object using one of its URIs (i.e., one of the values in the Printer object's "printer-uri-supported"  
1019 attribute). Even if the Printer object supports more than one URI, the client supplies only one URI as the  
1020 target of the operation. The client identifies the target object by supplying the correct URI in the "printer-  
1021 uri (uri)" operation attribute.

1022 For Job operations, the operation is directed at either:

- 1023 - The Job object itself using the Job object's URI. In this case, the client identifies the target object by  
1024 supplying the correct URI in the "job-uri (uri)" operation attribute.
- 1025 - The Printer object that created the Job object using both the Printer objects URI and the Job object's  
1026 Job ID. Since the Printer object that created the Job object generated the Job ID, it **MUST** be able to  
1027 correctly associate the client supplied Job ID with the correct Job object. The client supplies the  
1028 Printer object's URI in the "printer-uri (uri)" operation attribute and the Job object's Job ID in the  
1029 "job-id (integer(1:MAX))" operation attribute.

1030

1031 If the operation is directed at the Job object directly using the Job object's URI, the client **MUST NOT**  
1032 include the redundant "job-id" operation attribute.

1033 The operation target attributes are **REQUIRED** operation attributes that **MUST** be included in every  
1034 operation request. Like the charset and natural language attributes (see section 3.1.4), the operation target  
1035 attributes are specially ordered operation attributes. In all cases, the operation target attributes immediately  
1036 follow the "attributes-charset" and "attributes-natural-language" attributes within the operation attribute  
1037 group, however the specific ordering rules are:

- 1038 - In the case where there is only one operation target attribute (i.e., either only the "printer-uri" attribute  
1039 or only the "job-uri" attribute), that attribute **MUST** be the third attribute in the operation attributes  
1040 group.
- 1041 - In the case where Job operations use two operation target attributes (i.e., the "printer-uri" and "job-id"  
1042 attributes), the "printer-uri" attribute **MUST** be the third attribute and the "job-id" attribute **MUST**  
1043 be the fourth attribute.

1044

1045 In all cases, the target URIs contained within the body of IPP operation requests and responses must be in  
1046 absolute format rather than relative format (a relative URL identifies a resource with the scope of the HTTP  
1047 server, but does not include scheme, host or port).

1048 The following rules apply to the use of port numbers in URIs that identify IPP objects:

- 1049 1. If the URI scheme allows the port number to be explicitly included in the URI string, and a port  
1050 number is specified within the URI, then that port number **MUST** be used by the client to contact  
1051 the IPP object.



- 1052
- 1053 2. If the URI scheme allows the port number to be explicitly included in the URI string, and a port
- 1054 number is not specified within the URI, then default port number implied by that URI scheme
- 1055 MUST be used by the client to contact the IPP object.
- 1056
- 1057 3. If the URI scheme does not allow an explicit port number to be specified within the URI, then the
- 1058 default port number implied by that URI MUST be used by the client to contact the IPP object.
- 1059

1060 Note: The IPP "Encoding and Transport document [IPP-PRO] shows a mapping of IPP onto HTTP/1.1

1061 [RFC2616] and defines a new default port number for using IPP over HTTP/1.1.

### 1062 3.1.6 Operation Response Status Codes and Status Messages

1063 Every operation response includes a REQUIRED "status-code" parameter and an OPTIONAL "status-

1064 message" operation attribute, and an OPTIONAL "detailed-status-message" operation attribute. The Print-

1065 URI and Send-URI response MAY include an OPTIONAL "document-access-error" operation attribute.

#### 1066 3.1.6.1 "status-code" (type2 enum)

1067 The REQUIRED "status-code" parameter provides information on the processing of a request.

1068 The status code is intended for use by automata. A client implementation of IPP SHOULD convert status

1069 code values into any localized message that has semantic meaning to the end user.

1070 The "status-code" value is a numeric value that has semantic meaning. The "status-code" syntax is similar

1071 to a "type2 enum" (see section 4.1 on "Attribute Syntaxes") except that values can range only from 0x0000

1072 to 0x7FFF. Section 13 describes the status codes, assigns the numeric values, and suggests a corresponding

1073 status message for each status code for use by the client when the user's natural language is English.

1074 If the Printer performs an operation with no errors and it encounters no problems, it MUST return the status

1075 code 'successful-ok' in the response. See section 13.

1076 If the client supplies unsupported values for the following parameters or Operation attributes, the Printer

1077 object MUST reject the operation, NEED NOT return the unsupported attribute value in the Unsupported

1078 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

1079



1080 If the client supplies unsupported values for other attributes, or unsupported attributes, the Printer returns  
1081 the status code defined in section 3.1.7 on Unsupported Attributes.

### 1082 **3.1.6.2 "status-message" (text(255))**

1083 The OPTIONAL "status-message" operation attribute provides a short textual description of the status of  
1084 the operation. The "status-message" attribute's syntax is "text(255)", so the maximum length is 255 octets  
1085 (see section 4.1.1). The status message is intended for the human end user. If a response does include a  
1086 "status-message" attribute, an IPP client NEED NOT examine or display the messages, however it  
1087 SHOULD do so in some implementation specific manner. The "status-message" is especially useful for a  
1088 later version of a Printer object to return as supplemental information for the human user to accompany a  
1089 status code that an earlier version of a client might not understand.

1090 If the Printer object supports the "status-message" operation attribute, the Printer object MUST be able to  
1091 generate this message in any of the natural languages identified by the Printer object's "generated-natural-  
1092 language-supported" attribute (see the "attributes-natural-language" operation attribute specified in section  
1093 3.1.4.1. Section 13 suggests the text for the status message returned by the Printer for use with the English  
1094 natural language.

1095 As described in section 3.1.4.1 for any returned 'text' attribute, if there is a choice for generating this  
1096 message, the Printer object uses the natural language indicated by the value of the "attributes-natural-  
1097 language" in the client request if supported, otherwise the Printer object uses the value in the Printer  
1098 object's own "natural-language-configured" attribute.

1099 If the Printer object supports the "status-message" operation attribute, it SHOULD use the REQUIRED 'utf-  
1100 8' charset to return a status message for the following error status codes (see section 13): 'client-error-bad-  
1101 request', 'client-error-charset-not-supported', 'server-error-internal-error', 'server-error-operation-not-  
1102 supported', and 'server-error-version-not-supported'. In this case, it MUST set the value of the "attributes-  
1103 charset" operation attribute to 'utf-8' in the error response.

### 1104 **3.1.6.3 "detailed-status-message" (text(MAX))**

1105 The OPTIONAL "detailed-status-message" operation attribute provides additional more detailed technical  
1106 and implementation-specific information about the operation. The "detailed-status-message" attribute's  
1107 syntax is "text(MAX)", so the maximum length is 1023 octets (see section 4.1.1). If the Printer objects  
1108 supports the "detailed-status-message" operation attribute, neither the Printer nor the client localizes the  
1109 message, since it is intended for use by the system administrator or other experienced technical persons.  
1110 Clients MUST NOT attempt to parse the value of this attribute. See the "document-access-error" operation  
1111 attribute (section 3.1.6.4) for additional errors that a program can process.

### 1112 **3.1.6.4 "document-access-error" (text(MAX))**

1113 This OPTIONAL operation attribute provides additional information about any document access errors  
1114 encountered by the Printer before it returned a response to the Print-URI (section 3.2.2) or Send-URI  
1115 (section 3.3.1) operation. For errors in the protocol identified by the URI scheme in the "document-uri"

1116 operation attribute, such as 'http:' or 'ftp:', the error code is returned in parentheses, followed by the URI.  
1117 For example:

1118 (404) http://ftp.pwg.org/pub/pwg/ipp/new\_MOD/ipp-model-v11-990510.pdf  
1119

1120 Most Internet protocols use decimal error codes (unlike IPP), so the ASCII error code representation is in  
1121 decimal.

### 1122 3.1.7 Unsupported Attributes

1123 The Unsupported Attributes group contains attributes that are not supported by the operation. This group is  
1124 primarily for the job creation operations, but all operations can return this group.

1125 A Printer object MUST include an Unsupported Attributes group in a response if the status code is one of  
1126 the following: 'successful-ok-ignored-or-substituted-attributes', 'successful-ok-conflicting-attributes', 'client-  
1127 error-attributes-or-values-not-supported' or 'client-error-conflicting-attributes'.

1128 If the status code is one of the four specified in the preceding paragraph, the Unsupported Attributes group  
1129 MUST contain all of those attributes and only those attributes that are:

- 1130 a. an Operation or Job Template attribute supplied in the request, and
- 1131 b. unsupported by the printer. See below for details on the three categories "unsupported" attributes.

1132 If the status code is one of those in the table in section 3.1.6.1, the Unsupported Attributes group NEED  
1133 NOT contain the unsupported parameter or attribute indicated in that table.

1134 If the Printer object is not returning any Unsupported Attributes in the response, the Printer object  
1135 SHOULD omit Group 2 rather than sending an empty group. However, a client MUST be able to accept an  
1136 empty group.

1137 Unsupported attributes fall into three categories:

- 1138 1. The Printer object does not support the supplied attribute (no matter what the attribute syntax or  
1139 value).
- 1140 2. The Printer object does support the attribute, but does not support some or all of the particular  
1141 attribute syntaxes or values supplied by the client (i.e., the Printer object does not have those  
1142 attribute syntaxes or values in its corresponding "xxx-supported" attribute).
- 1143 3. The Printer object does support the attributes and values supplied, but the particular values are in  
1144 conflict with one another, because they violate a constraint, such as not being able to staple  
1145 transparencies.

1146 In the case of an unsupported attribute name, the Printer object returns the client-supplied attribute with a  
1147 substituted value of 'unsupported'. This value's syntax type is "out-of-band" and its encoding is defined by  
1148 special rules for "out-of-band" values in the "Encoding and Transport" document [IPP-PRO]. Its value  
1149 indicates no support for the attribute itself (see the beginning of section 4.1).

1150 In the case of a supported attribute with one or more unsupported attribute syntaxes or values, the Printer  
1151 object simply returns the client-supplied attribute with the unsupported attribute syntaxes or values as  
1152 supplied by the client. This indicates support for the attribute, but no support for that particular attribute  
1153 syntax or value. If the client supplies a multi-valued attribute with more than one value and the Printer  
1154 object supports the attribute but only supports a subset of the client-supplied attribute syntaxes or values,  
1155 the Printer object MUST return only those attribute syntaxes or values that are unsupported.

1156 In the case of two (or more) supported attribute values that are in conflict with one another (although each  
1157 is supported independently, the values conflict when requested together within the same job), the Printer  
1158 object MUST return all the values that it ignores or substitutes to resolve the conflict, but not any of the  
1159 values that it is still using. The choice for exactly how to resolve the conflict is implementation dependent.  
1160 See sections 3.2.1.2 and 15. See The Implementer's Guide [IPP-IIG] for an example.

### 1161 3.1.8 Versions

1162 Each operation request and response carries with it a "version-number" parameter. Each value of the  
1163 "version-number" is in the form "X.Y" where X is the major version number and Y is the minor version  
1164 number. By including a version number in the client request, it allows the client to identify which version  
1165 of IPP it is interested in using, i.e., the version whose conformance requirements the client may be  
1166 depending upon the Printer to meet.

1167 If the IPP object does not support that major version number supplied by the client, i.e., the major version  
1168 field of the "version-number" parameter does not match any of the values of the Printer's "ipp-versions-  
1169 supported" (see section 4.4.14), the object MUST respond with a status code of 'server-error-version-not-  
1170 supported' along with the closest version number that is supported (see section 13.1.5.4). If the major  
1171 version number is supported, but the minor version number is not, the IPP object SHOULD accept and  
1172 attempt to perform the request (or reject the request if the operation is not supported), else it rejects the  
1173 request and returns the 'server-error-version-not-supported' status code. In all cases, the IPP object MUST  
1174 return the "version-number" that it supports that is closest to the version number supplied by the client in  
1175 the request.

1176 There is no version negotiation per se. However, if after receiving a 'server-error-version-not-supported'  
1177 status code from an IPP object, a client SHOULD try again with a different version number. A client MAY  
1178 also determine the versions supported either from a directory that conforms to Appendix E (see section 16)  
1179 or by querying the Printer object's "ipp-versions-supported" attribute (see section 4.4.14) to determine  
1180 which versions are supported.

1181 An IPP object implementation MUST support version '1.1', i.e., meet the conformance requirements for  
1182 IPP/1.1 as specified in this document and [IPP-PRO]. It is recommended that IPP object implementations  
1183 accept any request with the major version '1' (or reject the request if the operation is not supported).

1184 There is only one notion of "version number" that covers both IPP Model and IPP Protocol changes. Thus  
1185 the version number MUST change when introducing a new version of the Model and Semantics document  
1186 (this document) or a new version of the "Encoding and Transport" document [IPP-PRO].

1187 Changes to the major version number of the Model and Semantics document indicate structural or syntactic  
1188 changes that make it impossible for older version of IPP clients and Printer objects to correctly parse and  
1189 correctly process the new or changed attributes, operations and responses. If the major version number  
1190 changes, the minor version numbers is set to zero. As an example, adding the REQUIRED "ipp-attribute-  
1191 fidelity" attribute to version '1.1' (if it had not been part of version '1.0'), would have required a change to  
1192 the major version number, since an IPP/1.0 Printer would not have processed a request with the correct  
1193 semantics that contained the "ipp-attribute-fidelity" attribute that it did not know about. Items that might  
1194 affect the changing of the major version number include any changes to the Model and Semantics document  
1195 (this document) or the "Encoding and Transport" document [IPP-PRO] itself, such as:

- 1196 - reordering of ordered attributes or attribute sets
- 1197 - changes to the syntax of existing attributes
- 1198 - adding REQUIRED (for an IPP object to support) operation attribute groups
- 1199 - adding values to existing REQUIRED operation attributes
- 1200 - adding REQUIRED operations

1201

1202 Changes to the minor version number indicate the addition of new features, attributes and attribute values  
1203 that may not be understood by all IPP objects, but which can be ignored if not understood. Items that might  
1204 affect the changing of the minor version number include any changes to the model objects and attributes but  
1205 not the encoding and transport rules [IPP-PRO] (except adding attribute syntaxes). Examples of such  
1206 changes are:

- 1207 - grouping all extensions not included in a previous version into a new version
- 1208 - adding new attribute values
- 1209 - adding new object attributes
- 1210 - adding OPTIONAL (for an IPP object to support) operation attributes (i.e., those attributes that an IPP  
1211 object can ignore without confusing clients)
- 1212 - adding OPTIONAL (for an IPP object to support) operation attribute groups (i.e., those attributes that  
1213 an IPP object can ignore without confusing clients)
- 1214 - adding new attribute syntaxes
- 1215 - adding OPTIONAL operations
- 1216 - changing Job Description attributes or Printer Description attributes from OPTIONAL to REQUIRED  
1217 or vice versa.
- 1218 - adding OPTIONAL attribute syntaxes to an existing attribute.

1219 The encoding of the "version-number" MUST NOT change over any version number (either major or  
1220 minor). This rule guarantees that all future versions will be backwards compatible with all previous  
1221 versions (at least for checking the "version-number"). In addition, any protocol elements (attributes, error  
1222 codes, tags, etc.) that are not carried forward from one version to the next are deprecated so that they can  
1223 never be reused with new semantics.

1224 Implementations that support a certain version NEED NOT support ALL previous versions. As each new  
1225 version is defined (through the release of a new IPP specification document), that version will specify  
1226 which previous versions MUST and which versions SHOULD be supported in compliant implementations.

### 1227 3.1.9 Job Creation Operations

1228 In order to "submit a print job" and create a new Job object, a client issues a create request. A create  
1229 request is any one of following three operation requests:

1230 - The Print-Job Request: A client that wants to submit a print job with only a single document uses the  
1231 Print-Job operation. The operation allows for the client to "push" the document data to the Printer  
1232 object by including the document data in the request itself.

1233  
1234 - The Print-URI Request: A client that wants to submit a print job with only a single document (where  
1235 the Printer object "pulls" the document data instead of the client "pushing" the data to the Printer  
1236 object) uses the Print-URI operation. In this case, the client includes in the request only a URI  
1237 reference to the document data (not the document data itself).

1238  
1239 - The Create-Job Request: A client that wants to submit a print job with multiple documents uses the  
1240 Create-Job operation. This operation is followed by an arbitrary number of Send-Document and/or  
1241 Send-URI operations (each creating another document for the newly create Job object). The Send-  
1242 Document operation includes the document data in the request (the client "pushes" the document  
1243 data to the printer), and the Send-URI operation includes only a URI reference to the document data  
1244 in the request (the Printer "pulls" the document data from the referenced location). The last Send-  
1245 Document or Send-URI request for a given Job object includes a "last-document" operation attribute  
1246 set to 'true' indicating that this is the last request.

1247

1248 Throughout this model document, the term "create request" is used to refer to any of these three operation  
1249 requests.

1250 A Create-Job operation followed by only one Send-Document operation is semantically equivalent to a  
1251 Print-Job operation, however, for performance reasons, the client SHOULD use the Print-Job operation for  
1252 all single document jobs. Also, Print-Job is a REQUIRED operation (all implementations MUST support  
1253 it) whereas Create-Job is an OPTIONAL operation, hence some implementations might not support it.

1254 Job submission time is the point in time when a client issues a create request. The initial state of every Job  
1255 object is the 'pending', 'pending-held', or 'processing' state (see section 4.3.7). When the Printer object  
1256 begins processing the print job, the Job object's state moves to 'processing'. This is known as job  
1257 processing time. There are validation checks that must be done at job submission time and others that must  
1258 be performed at job processing time.

1259 At job submission time and at the time a Validate-Job operation is received, the Printer MUST do the  
1260 following:

- 1261 1. Process the client supplied attributes and either accept or reject the request
- 1262 2. Validate the syntax of and support for the scheme of any client supplied URI

1263

1264 At job submission time the Printer object MUST validate whether or not the supplied attributes, attribute  
1265 syntaxes, and values are supported by matching them with the Printer object's corresponding "xxx-

1266 supported" attributes. See section 3.1.7 for details. [IPP-IIG] presents suggested steps for an IPP object to  
1267 either accept or reject any request and additional steps for processing create requests.

1268 At job submission time the Printer object NEED NOT perform the validation checks reserved for job  
1269 processing time such as:

- 1270 1. Validating the document data
- 1271 2. Validating the actual contents of any client supplied URI (resolve the reference and follow the link to  
1272 the document data)

1273

1274 At job submission time, these additional job processing time validation checks are essentially useless, since  
1275 they require actually parsing and interpreting the document data, are not guaranteed to be 100% accurate,  
1276 and MUST be done, yet again, at job processing time. Also, in the case of a URI, checking for availability  
1277 at job submission time does not guarantee availability at job processing time. In addition, at job processing  
1278 time, the Printer object might discover any of the following conditions that were not detectable at job  
1279 submission time:

- 1280 - runtime errors in the document data,
- 1281 - nested document data that is in an unsupported format,
- 1282 - the URI reference is no longer valid (i.e., the server hosting the document might be down), or
- 1283 - any other job processing error

1284

1285 At job submission time, a Printer object, especially a non-spooling Printer, MAY accept jobs that it does  
1286 not have enough space for. In such a situation, a Printer object MAY stop reading data from a client for an  
1287 indefinite period of time. A client MUST be prepared for a write operation to block for an indefinite period  
1288 of time (see section 5.1 on client conformance).

1289 When a Printer object has too little space for starting a new job, it MAY reject a new create request. In this  
1290 case, a Printer object MUST return a response (in reply to the rejected request) with a status-code of 'server-  
1291 error-busy' (see section 14.1.5.8) and it MAY close the connection before receiving all bytes of the  
1292 operation. A Printer SHOULD indicate that it is temporarily unable to accept jobs by setting the 'spool-  
1293 space-full' value in its "printer-state-reasons" attribute and removing the value when it can accept another  
1294 job (see section 4.4.12).

1295 When receiving a 'server-error-busy' status-code in an operation response, a client MUST be prepared for  
1296 the Printer object to close the connection before the client has sent all of the data (especially for the Print-  
1297 Job operation). A client MUST be prepared to keep submitting a create request until the IPP Printer object  
1298 accepts the create request.

1299 At job processing time, since the Printer object has already responded with a successful status code in the  
1300 response to the create request, if the Printer object detects an error, the Printer object is unable to inform the  
1301 end user of the error with an operation status code. In this case, the Printer, depending on the error, can set  
1302 the job object's "job-state", "job-state-reasons", or "job-state-message" attributes to the appropriate value(s)  
1303 so that later queries can report the correct job status.

1304 Note: Asynchronous notification of events is outside the scope of this IPP/1.1 document.



1305

## 1306 3.2 Printer Operations

1307 All Printer operations are directed at Printer objects. A client MUST always supply the "printer-uri"  
1308 operation attribute in order to identify the correct target of the operation.

### 1309 3.2.1 Print-Job Operation

1310 This REQUIRED operation allows a client to submit a print job with only one document and supply the  
1311 document data (rather than just a reference to the data). See Section 15 for the suggested steps for  
1312 processing create operations and their Operation and Job Template attributes.

#### 1313 3.2.1.1 Print-Job Request

1314 The following groups of attributes are supplied as part of the Print-Job Request:

##### 1315 Group 1: Operation Attributes

1316 Natural Language and Character Set:

1317 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.

1318 The Printer object MUST copy these values to the corresponding Job Description attributes  
1319 described in sections 4.3.19 and 4.3.20.

1320

1321 Target:

1322 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1323 section 3.1.5.

1324

1325 Requesting User Name:

1326 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1327 described in section 8.3.

1328

1329 "job-name" (name(MAX)):

1330 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1331 contains the client supplied Job name. If this attribute is supplied by the client, its value is used for  
1332 the "job-name" attribute of the newly created Job object. The client MAY automatically include any  
1333 information that will help the end-user distinguish amongst his/her jobs, such as the name of the  
1334 application program along with information from the document, such as the document name,  
1335 document subject, or source file name. If this attribute is not supplied by the client, the Printer  
1336 generates a name to use in the "job-name" attribute of the newly created Job object (see Section  
1337 4.3.5).

1338

1339 "ipp-attribute-fidelity" (boolean):

1340 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute.

1341 The value 'true' indicates that total fidelity to client supplied Job Template attributes and values is



1342 required, else the Printer object MUST reject the Print-Job request. The value 'false' indicates that a  
1343 reasonable attempt to print the Job object is acceptable and the Printer object MUST accept the  
1344 Print-Job request. If not supplied, the Printer object assumes the value is 'false'. All Printer objects  
1345 MUST support both types of job processing. See section 15 for a full description of "ipp-attribute-  
1346 fidelity" and its relationship to other attributes, especially the Printer object's "pdl-override-  
1347 supported" attribute.

1348  
1349 "document-name" (name(MAX)):

1350 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute.  
1351 It contains the client supplied document name. The document name MAY be different than the Job  
1352 name. Typically, the client software automatically supplies the document name on behalf of the end  
1353 user by using a file name or an application generated name. If this attribute is supplied, its value can  
1354 be used in a manner defined by each implementation. Examples include: printed along with the Job  
1355 (job start sheet, page adornments, etc.), used by accounting or resource tracking management tools,  
1356 or even stored along with the document as a document level attribute. IPP/1.1 does not support the  
1357 concept of document level attributes.

1358  
1359 "compression" (type3 keyword)

1360 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute  
1361 and the "compression-supported" attribute (see section 4.4.32). The client supplied "compression"  
1362 operation attribute identifies the compression algorithm used on the document data. The following  
1363 cases exist:

- 1364 a) If the client omits this attribute, the Printer object MUST assume that the data is not  
1365 compressed (i.e. the Printer follows the rules below as if the client supplied the  
1366 "compression" attribute with a value of 'none').
  - 1367 b) If the client supplies this attribute, but the value is not supported by the Printer object,  
1368 i.e., the value is not one of the values of the Printer object's "compression-supported"  
1369 attribute, the Printer object MUST reject the request, and return the 'client-error-  
1370 compression-not-supported' status code. See section 3.1.7 for returning unsupported  
1371 attributes and values.
  - 1372 c) If the client supplies the attribute and the Printer object supports the attribute value, the  
1373 Printer object uses the corresponding decompression algorithm on the document data.
  - 1374 d) If the decompression algorithm fails before the Printer returns an operation response, the  
1375 Printer object MUST reject the request and return the 'client-error-compression-error'  
1376 status code.
  - 1377 e) If the decompression algorithm fails after the Printer returns an operation response, the  
1378 Printer object MUST abort the job and add the 'compression-error' value to the job's  
1379 "job-state-reasons" attribute.
  - 1380 f) If the decompression algorithm succeeds, the document data MUST then have the format  
1381 specified by the job's "document-format" attribute, if supplied (see "document-format"  
1382 operation attribute definition below).
- 1383

1384 "document-format" (mimeMediaType) :

1385 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute.  
1386 The value of this attribute identifies the format of the supplied document data. The following cases  
1387 exist:

- 1388 a) If the client does not supply this attribute, the Printer object assumes that the document  
1389 data is in the format defined by the Printer object's "document-format-default" attribute.  
1390 (i.e. the Printer follows the rules below as if the client supplied the "document-format"  
1391 attribute with a value equal to the printer's default value).
- 1392 b) If the client supplies this attribute, but the value is not supported by the Printer object,  
1393 i.e., the value is not one of the values of the Printer object's "document-format-  
1394 supported" attribute, the Printer object MUST reject the request and return the 'client-  
1395 error-document-format-not-supported' status code.
- 1396 c) If the client supplies this attribute and its value is 'application/octet-stream' (i.e. to be  
1397 auto-sensed, see Section 4.1.9.1), and the format is not one of the document-formats that  
1398 the Printer can auto-sense, and this check occurs before the Printer returns an operation  
1399 response, then the Printer MUST reject the request and return the 'client-error-  
1400 document-format-not-supported' status code.
- 1401 d) If the client supplies this attribute, and the value is supported by the Printer object, the  
1402 document data, the Printer is capable of interpreting the document data.
- 1403 e) If interpreting of the document data fails before the Printer returns an operation response,  
1404 the Printer object MUST reject the request and return the 'client-error-document-format-  
1405 error' status code.
- 1406 f) If interpreting of the document data fails after the Printer returns an operation response,  
1407 the Printer object MUST abort the job and add the 'document-format-error' value to the  
1408 job's "job-state-reasons" attribute.

1409 "document-natural-language" (naturalLanguage):

1410 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1411 attribute. This attribute specifies the natural language of the document for those document-formats  
1412 that require a specification of the natural language in order to image the document unambiguously.  
1413 There are no particular values required for the Printer object to support.  
1414

1415 "job-k-octets" (integer(0:MAX))

1416 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1417 attribute and the "job-k-octets-supported" attribute (see section 4.4.33). The client supplied "job-k-  
1418 octets" operation attribute identifies the total size of the document(s) in K octets being submitted  
1419 (see section 4.3.17.1 for the complete semantics). If the client supplies the attribute and the Printer  
1420 object supports the attribute, the value of the attribute is used to populate the Job object's "job-k-  
1421 octets" Job Description attribute.  
1422

1423  
1424 For this attribute and the following two attributes ("job-impressions", and "job-media-sheets"), if the  
1425 client supplies the attribute, but the Printer object does not support the attribute, the Printer object  
1426 ignores the client-supplied value. If the client supplies the attribute and the Printer supports the  
1427 attribute, and the value is within the range of the corresponding Printer object's "xxx-supported"  
1428 attribute, the Printer object MUST use the value to populate the Job object's "xxx" attribute. If the

1429 client supplies the attribute and the Printer supports the attribute, but the value is outside the range  
1430 of the corresponding Printer object's "xxx-supported" attribute, the Printer object MUST copy the  
1431 attribute and its value to the Unsupported Attributes response group, reject the request, and return  
1432 the 'client-error-attributes-or-values-not-supported' status code. If the client does not supply the  
1433 attribute, the Printer object MAY choose to populate the corresponding Job object attribute  
1434 depending on whether the Printer object supports the attribute and is able to calculate or discern the  
1435 correct value.

1436  
1437 "job-impressions" (integer(0:MAX))

1438 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1439 attribute and the "job-impressions-supported" attribute (see section 4.4.34). The client supplied  
1440 "job-impressions" operation attribute identifies the total size in number of impressions of the  
1441 document(s) being submitted (see section 4.3.17.2 for the complete semantics).

1442  
1443 See last paragraph under "job-k-octets".  
1444

1445 "job-media-sheets" (integer(0:MAX))

1446 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1447 attribute and the "job-media-sheets-supported" attribute (see section 4.4.35). The client supplied  
1448 "job-media-sheets" operation attribute identifies the total number of media sheets to be produced for  
1449 this job (see section 4.3.17.3 for the complete semantics).

1450  
1451 See last paragraph under "job-k-octets".  
1452

1453 Group 2: Job Template Attributes

1454 The client OPTIONALLY supplies a set of Job Template attributes as defined in section 4.2. If the  
1455 client is not supplying any Job Template attributes in the request, the client SHOULD omit Group 2  
1456 rather than sending an empty group. However, a Printer object MUST be able to accept an empty  
1457 group.  
1458

1459 Group 3: Document Content

1460 The client MUST supply the document data to be processed.  
1461

1462 In addition to the MANDATORY parameters required for every operation request, the simplest Print-Job  
1463 Request consists of just the "attributes-charset" and "attributes-natural-language" operation attributes; the  
1464 "printer-uri" target operation attribute; the Document Content and nothing else. In this simple case, the  
1465 Printer object:

- 1466 - creates a new Job object (the Job object contains a single document),
- 1467 - stores a generated Job name in the "job-name" attribute in the natural language and charset requested  
1468 (see Section 3.1.4.1) (if those are supported, otherwise using the Printer object's default natural  
1469 language and charset), and

1470 - at job processing time, uses its corresponding default value attributes for the supported Job Template  
1471 attributes that were not supplied by the client as IPP attribute or embedded instructions in the  
1472 document data.  
1473

### 1474 3.2.1.2 Print-Job Response

1475 The Printer object MUST return to the client the following sets of attributes as part of the Print-Job  
1476 Response:

#### 1477 Group 1: Operation Attributes

##### 1478 Status Message:

1479 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
1480 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
1481 attribute as described in sections 13 and 3.1.6. If the client supplies unsupported or conflicting Job  
1482 Template attributes or values, the Printer object MUST reject or accept the Print-Job request  
1483 depending on the whether the client supplied a 'true' or 'false' value for the "ipp-attribute-fidelity"  
1484 operation attribute. See the Implementer's Guide [IPP-IIG] for a complete description of the  
1485 suggested steps for processing a create request.  
1486

##### 1487 Natural Language and Character Set:

1488 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.  
1489

#### 1490 Group 2: Unsupported Attributes

1491 See section 3.1.7 for details on returning Unsupported Attributes.  
1492

1493 The value of the "ipp-attribute-fidelity" supplied by the client does not affect what attributes the  
1494 Printer object returns in this group. The value of "ipp-attribute-fidelity" only affects whether the  
1495 Print-Job operation is accepted or rejected. If the job is accepted, the client may query the job using  
1496 the Get-Job-Attributes operation requesting the unsupported attributes that were returned in the  
1497 create response to see which attributes were ignored (not stored on the Job object) and which  
1498 attributes were stored with other (substituted) values.  
1499

#### 1500 Group 3: Job Object Attributes

##### 1501 "job-uri" (uri):

1502 The Printer object MUST return the Job object's URI by returning the contents of the REQUIRED  
1503 "job-uri" Job object attribute. The client uses the Job object's URI when directing operations at the  
1504 Job object. The Printer object always uses its configured security policy when creating the new  
1505 URI. However, if the Printer object supports more than one URI, the Printer object also uses  
1506 information about which URI was used in the Print-Job Request to generated the new URI so that  
1507 the new URI references the correct access channel. In other words, if the Print-Job Request comes  
1508 in over a secure channel, the Printer object MUST generate a Job URI that uses the secure channel  
1509 as well.

1510

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

1512

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

1515

"job-state":

1517

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

1520

1521

"job-state-reasons":

1523

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

1524

1525

"job-state-message":

1526

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

1527

1528

1529

1530

1531

"number-of-intervening-jobs":

1532

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

1533

1534

1535

1536

1537

1538

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

1539

1540

1541

1542

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

1543

1544

1545

1546

### 3.2.2 Print-URI Operation

1547

This OPTIONAL operation is identical to the Print-Job operation (section 3.2.1) except that a client supplies a URI reference to the document data using the "document-uri" (uri) operation attribute (in Group 1) rather than including the document data itself. Before returning the response, the Printer MUST validate that the Printer supports the retrieval method (e.g., http, ftp, etc.) implied by the URI, and MUST check for valid URI syntax. If the client-supplied URI scheme is not supported, i.e. the value is not in the Printer

1548

1549

1550

1551

1552 object's "referenced-uri-scheme-supported" attribute, the Printer object MUST reject the request and return  
1553 the 'client-error-uri-scheme-not-supported' status code.

1554 The IPP Printer MAY validate the accessibility of the document as part of the operation or subsequently. If  
1555 the Printer determines an accessibility problem before returning an operation response, it rejects the request  
1556 and returns the 'client-error-document-access-error' status code. The Printer MAY also return a specific  
1557 document access error code using the "document-access-error" operation attribute (see section 3.1.6.4).

1558 If the Printer determines this document accessibility problem after accepting the request and returning an  
1559 operation response with one of the successful status codes, the Printer adds the 'document-access-error'  
1560 value to the job's "job-state-reasons" attribute and MAY populate the job's "job-document-access-errors"  
1561 Job Description attribute (see section 4.3.11). See The Implementer's Guide [IPP-IIG] for suggested  
1562 additional checks.

1563 If the Printer object supports this operation, it MUST support the "reference-uri-schemes-supported" Printer  
1564 attribute (see section 4.4.27).

1565 It is up to the IPP object to interpret the URI and subsequently "pull" the document from the source  
1566 referenced by the URI string.

### 1567 **3.2.3 Validate-Job Operation**

1568 This REQUIRED operation is similar to the Print-Job operation (section 3.2.1) except that a client supplies  
1569 no document data and the Printer allocates no resources (i.e., it does not create a new Job object). This  
1570 operation is used only to verify capabilities of a printer object against whatever attributes are supplied by  
1571 the client in the Validate-Job request. By using the Validate-Job operation a client can validate that an  
1572 identical Print-Job operation (with the document data) would be accepted. The Validate-Job operation also  
1573 performs the same security negotiation as the Print-Job operation (see section 8), so that a client can check  
1574 that the client and Printer object security requirements can be met before performing a Print-Job operation.

1575 The Validate-Job operation does not accept a "document-uri" attribute in order to allow a client to check  
1576 that the same Print-URI operation will be accepted, since the client doesn't send the data with the Print-URI  
1577 operation. The client SHOULD just issue the Print-URI request.

1578 The Printer object returns the same status codes, Operation Attributes (Group 1) and Unsupported  
1579 Attributes (Group 2) as the Print-Job operation. However, no Job Object Attributes (Group 3) are returned,  
1580 since no Job object is created.

### 1581 **3.2.4 Create-Job Operation**

1582 This OPTIONAL operation is similar to the Print-Job operation (section 3.2.1) except that in the Create-Job  
1583 request, a client does not supply document data or any reference to document data. Also, the client does not  
1584 supply any of the "document-name", "document-format", "compression", or "document-natural-language"  
1585 operation attributes. This operation is followed by one or more Send-Document or Send-URI operations.



1586 In each of those operation requests, the client **OPTIONALLY** supplies the "document-name", "document-  
1587 format", and "document-natural-language" attributes for each document in the multi-document Job object.

1588 If a Printer object supports the Create-Job operation, it **MUST** also support the Send-Document operation  
1589 and also **MAY** support the Send-URI operation.

1590 If the Printer object supports this operation, it **MUST** support the "multiple-operation-time-out" Printer  
1591 attribute (see section 4.4.31).

1592 If the Printer object supports this operation, then it **MUST** support the "multiple-document-jobs-supported"  
1593 Printer Description attribute (see section 4.4.16) and indicate whether or not it supports multiple-document  
1594 jobs.

1595 If the Printer object supports this operation and supports multiple documents in a job, then it **MUST** support  
1596 the "multiple-document-handling" Job Template job attribute with at least one value (see section 4.2.4) and  
1597 the associated "multiple-document-handling-default" and "multiple-document-handling-supported" Job  
1598 Template Printer attributes (see section 4.2).

1599 After the Create-Job operation has completed, the value of the "job-state" attribute is similar to the "job-  
1600 state" after a Print-Job, even though no document-data has arrived. A Printer **MAY** set the 'job-data-  
1601 insufficient' value of the job's "job-state-reason" attribute to indicate that processing cannot begin until  
1602 sufficient data has arrived and set the "job-state" to either 'pending' or 'pending-held'. A non-spooling  
1603 printer that doesn't implement the 'pending' job state may even set the "job-state" to 'processing', even  
1604 though there is not yet any data to process. See sections 4.3.7 and 4.3.8.

### 1605 **3.2.5 Get-Printer-Attributes Operation**

1606 This **REQUIRED** operation allows a client to request the values of the attributes of a Printer object. In the  
1607 request, the client supplies the set of Printer attribute names and/or attribute group names in which the  
1608 requester is interested. In the response, the Printer object returns a corresponding attribute set with the  
1609 appropriate attribute values filled in.

1610 For Printer objects, the possible names of attribute groups are:

- 1611 - 'job-template': the subset of the Job Template attributes that apply to a Printer object (the last two  
1612 columns of the table in Section 4.2) that the implementation supports for Printer objects.
  - 1613 - 'printer-description': the subset of the attributes specified in Section 4.4 that the implementation  
1614 supports for Printer objects.
  - 1615 - 'all': the special group 'all' that includes all attributes that the implementation supports for Printer  
1616 objects.
- 1617

1618 Since a client **MAY** request specific attributes or named groups, there is a potential that there is some  
1619 overlap. For example, if a client requests, 'printer-name' and 'all', the client is actually requesting the  
1620 "printer-name" attribute twice: once by naming it explicitly, and once by inclusion in the 'all' group. In such  
1621 cases, the Printer object **NEED NOT** return each attribute only once in the response even if it is requested  
1622 multiple times. The client **SHOULD NOT** request the same attribute in multiple ways.



1623 It is NOT REQUIRED that a Printer object support all attributes belonging to a group (since some attributes  
1624 are OPTIONAL). However, it is REQUIRED that each Printer object support all group names.

### 1625 **3.2.5.1 Get-Printer-Attributes Request**

1626 The following sets of attributes are part of the Get-Printer-Attributes Request:

1627 Group 1: Operation Attributes

1628 Natural Language and Character Set:

1629 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.

1630

1631 Target:

1632 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1633 section 3.1.5.

1634

1635 Requesting User Name:

1636 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1637 described in section 8.3.

1638

1639 "requested-attributes" (1setOf keyword) :

1640 The client OPTIONALLY supplies a set of attribute names and/or attribute group names in whose  
1641 values the requester is interested. The Printer object MUST support this attribute. If the client  
1642 omits this attribute, the Printer MUST respond as if this attribute had been supplied with a value of  
1643 'all'.

1644

1645 "document-format" (mimeMediaType) :

1646 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute.  
1647 This attribute is useful for a Printer object to determine the set of supported attribute values that  
1648 relate to the requested document format. The Printer object MUST return the attributes and values  
1649 that it uses to validate a job on a create or Validate-Job operation in which this document format is  
1650 supplied. The Printer object SHOULD return only (1) those attributes that are supported for the  
1651 specified format and (2) the attribute values that are supported for the specified document format.  
1652 By specifying the document format, the client can get the Printer object to eliminate the attributes  
1653 and values that are not supported for a specific document format. For example, a Printer object  
1654 might have multiple interpreters to support both 'application/postscript' (for PostScript) and  
1655 'text/plain' (for text) documents. However, for only one of those interpreters might the Printer  
1656 object be able to support "number-up" with values of '1', '2', and '4'. For the other interpreter it  
1657 might be able to only support "number-up" with a value of '1'. Thus a client can use the Get-Printer-  
1658 Attributes operation to obtain the attributes and values that will be used to accept/reject a create job  
1659 operation.

1660

1661 If the Printer object does not distinguish between different sets of supported values for each  
1662 different document format when validating jobs in the create and Validate-Job operations, it MUST  
1663 NOT distinguish between different document formats in the Get-Printer-Attributes operation. If the

1664 Printer object does distinguish between different sets of supported values for each different  
1665 document format specified by the client, this specialization applies only to the following Printer  
1666 object attributes:

- 1667
- 1668 - Printer attributes that are Job Template attributes ("xxx-default" "xxx-supported", and "xxx-  
1669 ready" in the Table in Section 4.2),
- 1670 - "pdl-override-supported",
- 1671 - "compression-supported",
- 1672 - "job-k-octets-supported",
- 1673 - "job-impressions-supported",
- 1674 - "job-media-sheets-supported"
- 1675 - "printer-driver-installer",
- 1676 - "color-supported", and
- 1677 - "reference-uri-schemes-supported"

1678  
1679 The values of all other Printer object attributes (including "document-format-supported") remain  
1680 invariant with respect to the client supplied document format (except for new Printer description  
1681 attribute as registered according to section 6.2).

1682  
1683 If the client omits this "document-format" operation attribute, the Printer object MUST respond as if  
1684 the attribute had been supplied with the value of the Printer object's "document-format-default"  
1685 attribute. It is recommended that the client always supply a value for "document-format", since the  
1686 Printer object's "document-format-default" may be 'application/octet-stream', in which case the  
1687 returned attributes and values are for the union of the document formats that the Printer can  
1688 automatically sense. For more details, see the description of the 'mimeType' attribute syntax  
1689 in section 4.1.9.

1690  
1691 If the client supplies a value for the "document-format" Operation attribute that is not supported by  
1692 the Printer, i.e., is not among the values of the Printer object's "document-format-supported"  
1693 attribute, the Printer object MUST reject the operation and return the 'client-error-document-format-  
1694 not-supported' status code.

1695

### 1696 3.2.5.2 Get-Printer-Attributes Response

1697 The Printer object returns the following sets of attributes as part of the Get-Printer-Attributes Response:

#### 1698 Group 1: Operation Attributes

1699 Status Message:

1700 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
1701 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
1702 attribute as described in sections 13 and 3.1.6.

1703

1704 Natural Language and Character Set:

1705 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.

1706

## 1707 Group 2: Unsupported Attributes

1708 See section 3.1.7 for details on returning Unsupported Attributes.

1709

1710 The response NEED NOT contain the "requested-attributes" operation attribute with any supplied  
1711 values (attribute keywords) that were requested by the client but are not supported by the IPP object.  
1712 If the Printer object does include unsupported attributes referenced in "requested-attributes" and  
1713 such attributes include group names, such as 'all', the unsupported attributes MUST NOT include  
1714 attributes described in the standard but not supported by the implementation.

1715

## 1716 Group 3: Printer Object Attributes

1717 This is the set of requested attributes and their current values. The Printer object ignores (does not  
1718 respond with) any requested attribute which is not supported. The Printer object MAY respond with  
1719 a subset of the supported attributes and values, depending on the security policy in force. However,  
1720 the Printer object MUST respond with the 'unknown' value for any supported attribute (including all  
1721 REQUIRED attributes) for which the Printer object does not know the value. Also the Printer  
1722 object MUST respond with the 'no-value' for any supported attribute (including all REQUIRED  
1723 attributes) for which the system administrator has not configured a value. See the description of the  
1724 "out-of-band" values in the beginning of Section 4.1.

1725

1726 **3.2.6 Get-Jobs Operation**

1727 This REQUIRED operation allows a client to retrieve the list of Job objects belonging to the target Printer  
1728 object. The client may also supply a list of Job attribute names and/or attribute group names. A group of  
1729 Job object attributes will be returned for each Job object that is returned.

1730 This operation is similar to the Get-Job-Attributes operation, except that this Get-Jobs operation returns  
1731 attributes from possibly more than one object (see the description of Job attribute group names in section  
1732 3.3.4).

1733 **3.2.6.1 Get-Jobs Request**

1734 The client submits the Get-Jobs request to a Printer object.

1735 The following groups of attributes are part of the Get-Jobs Request:

## 1736 Group 1: Operation Attributes

1737 Natural Language and Character Set:

1738 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.

1739

## 1740 Target:

1741 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1742 section 3.1.5.

1743  
1744 Requesting User Name:

1745 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1746 described in section 8.3.

1747  
1748 "limit" (integer(1:MAX)):

1749 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1750 is an integer value that determines the maximum number of jobs that a client will receive from the  
1751 Printer even if "which-jobs" or "my-jobs" constrain which jobs are returned. The limit is a "stateless  
1752 limit" in that if the value supplied by the client is 'N', then only the first 'N' jobs are returned in the  
1753 Get-Jobs Response. There is no mechanism to allow for the next 'M' jobs after the first 'N' jobs. If  
1754 the client does not supply this attribute, the Printer object responds with all applicable jobs.

1755  
1756 "requested-attributes" (1setOf keyword):

1757 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1758 is a set of Job attribute names and/or attribute groups names in whose values the requester is  
1759 interested. This set of attributes is returned for each Job object that is returned. The allowed  
1760 attribute group names are the same as those defined in the Get-Job-Attributes operation in section  
1761 3.3.4. If the client does not supply this attribute, the Printer MUST respond as if the client had  
1762 supplied this attribute with two values: 'job-uri' and 'job-id'.

1763  
1764 "which-jobs" ([type2](#) keyword):

1765 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1766 indicates which Job objects MUST be returned by the Printer object. The values for this attribute  
1767 are:

1768 'completed': This includes any Job object whose state is 'completed', 'canceled', or 'aborted'.

1770 'not-completed': This includes any Job object whose state is 'pending', 'processing', 'processing-  
1771 stopped', or 'pending-held'.

1772  
1773 A Printer object MUST support both values. However, if the implementation does not keep jobs in  
1774 the 'completed', 'canceled', and 'aborted' states, then it returns no jobs when the 'completed' value is  
1775 supplied.

1776  
1777 If a client supplies some other value, the Printer object MUST copy the attribute and the  
1778 unsupported value to the Unsupported Attributes response group, reject the request, and return the  
1779 'client-error-attributes-or-values-not-supported' status code.

1780  
1781 If the client does not supply this attribute, the Printer object MUST respond as if the client had  
1782 supplied the attribute with a value of 'not-completed'.

1783

1784 "my-jobs" (boolean):

1785 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this attribute. It  
1786 indicates whether jobs from all users or just the jobs submitted by the requesting user of this request  
1787 **MUST** be returned by the Printer object. If the client does not supply this attribute, the Printer  
1788 object **MUST** respond as if the client had supplied the attribute with a value of 'false', i.e., jobs from  
1789 all users. The means for authenticating the requesting user and matching the jobs is described in  
1790 section 8.

### 1791 3.2.6.2 Get-Jobs Response

1792 The Printer object returns all of the Job objects up to the number specified by the "limit" attribute that  
1793 match the criteria as defined by the attribute values supplied by the client in the request. It is possible that  
1794 no Job objects are returned since there may literally be no Job objects at the Printer, or there may be no Job  
1795 objects that match the criteria supplied by the client. If the client requests any Job attributes at all, there is a  
1796 set of Job Object Attributes returned for each Job object.

1797 It is not an error for the Printer to return 0 jobs. If the response returns 0 jobs because there are no jobs  
1798 matching the criteria, and the request would have returned 1 or more jobs with a status code of 'successful-  
1799 ok' if there had been jobs matching the criteria, then the status code for 0 jobs **MUST** be 'successful-ok'.

#### 1800 Group 1: Operation Attributes

1801 Status Message:

1802 In addition to the **REQUIRED** status code returned in every response, the response **OPTIONALLY**  
1803 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
1804 attribute as described in sections 13 and 3.1.6.

1805  
1806 Natural Language and Character Set:

1807 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.  
1808

#### 1809 Group 2: Unsupported Attributes

1810 See section 3.1.7 for details on returning Unsupported Attributes.

1811  
1812 The response **NEED NOT** contain the "requested-attributes" operation attribute with any supplied  
1813 values (attribute keywords) that were requested by the client but are not supported by the IPP object.  
1814 If the Printer object does include unsupported attributes referenced in "requested-attributes" and  
1815 such attributes include group names, such as 'all', the unsupported attributes **MUST NOT** include  
1816 attributes described in the standard but not supported by the implementation.  
1817

#### 1818 Groups 3 to N: Job Object Attributes

1819 The Printer object responds with one set of Job Object Attributes for each returned Job object. The  
1820 Printer object ignores (does not respond with) any requested attribute or value which is not  
1821 supported or which is restricted by the security policy in force, including whether the requesting  
1822 user is the user that submitted the job (job originating user) or not (see section 8). However, the

1823 Printer object MUST respond with the 'unknown' value for any supported attribute (including all  
1824 REQUIRED attributes) for which the Printer object does not know the value, unless it would violate  
1825 the security policy. See the description of the "out-of-band" values in the beginning of Section 4.1.  
1826

1827 Jobs are returned in the following order:

- 1828 - If the client requests all 'completed' Jobs (Jobs in the 'completed', 'aborted', or 'canceled' states),  
1829 then the Jobs are returned newest to oldest (with respect to actual completion time)
- 1830 - If the client requests all 'not-completed' Jobs (Jobs in the 'pending', 'processing', 'pending-held',  
1831 and 'processing-stopped' states), then Jobs are returned in relative chronological order of  
1832 expected time to complete (based on whatever scheduling algorithm is configured for the  
1833 Printer object).

### 1834 3.2.7 Pause-Printer Operation

1835 This OPTIONAL operation allows a client to stop the Printer object from scheduling jobs on all its devices.  
1836 Depending on implementation, the Pause-Printer operation MAY also stop the Printer from processing the  
1837 current job or jobs. Any job that is currently being printed is either stopped as soon as the implementation  
1838 permits or is completed, depending on implementation. The Printer object MUST still accept create  
1839 operations to create new jobs, but MUST prevent any jobs from entering the 'processing' state.

1840 If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and  
1841 vice-versa.

1842 The IPP Printer stops the current job(s) on its device(s) that were in the 'processing' or 'processing-stopped'  
1843 states as soon as the implementation permits. If the implementation will take appreciable time to stop, the  
1844 IPP Printer adds the 'moving-to-paused' value to the Printer object's "printer-state-reasons" attribute (see  
1845 section 4.4.12). When the device(s) have all stopped, the IPP Printer transitions the Printer object to the  
1846 'stopped' state, removes the 'moving-to-paused' value, if present, and adds the 'paused' value to the Printer  
1847 object's "printer-state-reasons" attribute.

1848 When the current job(s) complete that were in the 'processing' state, the IPP Printer transitions them to the  
1849 'completed' state. When the current job(s) stop in mid processing that were in the 'processing' state, the IPP  
1850 Printer transitions them to the 'processing-stopped' state and adds the 'printer-stopped' value to the job's  
1851 "job-state-reasons" attribute.

1852 For any jobs that are 'pending' or 'pending-held', the 'printer-stopped' value of the jobs' "job-state-reasons"  
1853 attribute also applies. However, the IPP Printer NEED NOT update those jobs' "job-state-reasons"  
1854 attributes and only need return the 'printer-stopped' value when those jobs are queried (so-called "lazy  
1855 evaluation").

1856 Whether the Pause-Printer operation affects jobs that were submitted to the device from other sources than  
1857 the IPP Printer object in the same way that the Pause-Printer operation affects jobs that were submitted to  
1858 the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP protocol is being used  
1859 as a universal management protocol or just to manage IPP jobs, respectively.

1860 The IPP Printer MUST accept the request in any state and transition the Printer to the indicated new  
 1861 "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'

1862 *Access Rights:* The authenticated user (see section 8.3) performing this operation must be an operator or  
 1863 administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP Printer MUST reject the  
 1864 operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized'  
 1865 as appropriate.

### 1866 3.2.7.1 Pause-Printer Request

1867 The following groups of attributes are part of the Pause-Printer Request:

#### 1868 Group 1: Operation Attributes

##### 1869 Natural Language and Character Set:

1870 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.

##### 1871 Target:

1872 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
 1873 section 3.1.5.

##### 1874 Requesting User Name:

1875 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
 1876 described in section 8.3.

### 1879 3.2.7.2 Pause-Printer Response

1880 The following groups of attributes are part of the Pause-Printer Response:

#### 1881 Group 1: Operation Attributes

##### 1882 Status Message:

1883 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
 1884 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
 1885 attribute as described in sections 13 and 3.1.6.

1886



1887 Natural Language and Character Set:  
 1888 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.  
 1889

1890 Group 2: Unsupported Attributes

1891 See section 3.1.7 for details on returning Unsupported Attributes.  
 1892

### 1893 3.2.8 Resume-Printer Operation

1894 This operation allows a client to resume the Printer object scheduling jobs on all its devices. The Printer  
 1895 object MUST remove the 'paused' and 'moving-to-paused' values from the Printer object's "printer-state-  
 1896 reasons" attribute, if present. If there are no other reasons to keep a device paused (such as media-jam), the  
 1897 IPP Printer transitions itself to the 'processing' or 'idle' states, depending on whether there are jobs to be  
 1898 processed or not, respectively, and the device(s) resume processing jobs.

1899 If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and  
 1900 vice-versa.

1901 The IPP Printer removes the 'printer-stopped' value from any job's "job-state-reasons" attributes contained  
 1902 in that Printer.

1903 The IPP Printer MUST accept the request in any state, transition the Printer object to the indicated new state  
 1904 as follows:

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

1905 *Access Rights:* The authenticated user (see section 8.3) performing this operation must be an operator or  
 1906 administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP Printer MUST reject the  
 1907 operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized'  
 1908 as appropriate.

1909 The Resume-Printer Request and Resume-Printer Response have the same attribute groups and attributes as  
 1910 the Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

### 1911 3.2.9 Purge-Jobs Operation

1912 This OPTIONAL operation allows a client to remove all jobs from an IPP Printer object, regardless of their  
 1913 job states, including jobs in the Printer object's Job History (see Section 4.3.7.2). After a Purge-Jobs

1914 operation has been performed, a Printer object MUST return no jobs in subsequent Get-Job-Attributes and  
1915 Get-Jobs responses (until new jobs are submitted).

1916 Whether the Purge-Jobs (and Get-Jobs) operation affects jobs that were submitted to the device from other  
1917 sources than the IPP Printer object in the same way that the Purge-Jobs operation affects jobs that were  
1918 submitted to the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP protocol  
1919 is being used as a universal management protocol or just to manage IPP jobs, respectively.

1920 Note: if an operator wants to cancel all jobs without clearing out the Job History, the operator uses the  
1921 Cancel-Job operation on each job instead of using the Purge-Jobs operation.

1922 The Printer object MUST accept this operation in any state and transition the Printer object to the 'idle'  
1923 state.

1924 *Access Rights:* The authenticated user (see section 8.3) performing this operation must be an operator or  
1925 administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP object MUST reject the  
1926 operation and return: client-error-forbidden, client-error-not-authenticated, and client-error-not-authorized  
1927 as appropriate.

1928 The Purge-Jobs Request and Purge-Jobs Response have the same attribute groups and attributes as the  
1929 Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

1930

### 1931 **3.3 Job Operations**

1932 All Job operations are directed at Job objects. A client MUST always supply some means of identifying the  
1933 Job object in order to identify the correct target of the operation. That job identification MAY either be a  
1934 single Job URI or a combination of a Printer URI with a Job ID. The IPP object implementation MUST  
1935 support both forms of identification for every job.

#### 1936 **3.3.1 Send-Document Operation**

1937 This OPTIONAL operation allows a client to create a multi-document Job object that is initially "empty"  
1938 (contains no documents). In the Create-Job response, the Printer object returns the Job object's URI (the  
1939 "job-uri" attribute) and the Job object's 32-bit identifier (the "job-id" attribute). For each new document  
1940 that the client desires to add, the client uses a Send-Document operation. Each Send-Document Request  
1941 contains the entire stream of document data for one document.

1942 If the Printer supports this operation but does not support multiple documents per job, the Printer MUST  
1943 reject subsequent Send-Document operations supplied with data and return the 'server-error-multiple-  
1944 document-jobs-not-supported'. However, the Printer MUST accept the first document with a 'true' or 'false'  
1945 value for the "last-document" operation attribute (see below), so that clients MAY always submit one  
1946 document jobs with a 'false' value for "last-document" in the first Send-Document and a 'true' for "last-  
1947 document" in the second Send-Document (with no data).

1948 Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow could  
1949 occur over an arbitrarily long period of time for a particular job, a client MUST send another send operation  
1950 within an IPP Printer defined minimum time interval after the receipt of the previous request for the job. If  
1951 a Printer object supports multiple document jobs, the Printer object MUST support the "multiple-operation-  
1952 time-out" attribute (see section 4.4.31). This attribute indicates the minimum number of seconds the Printer  
1953 object will wait for the next send operation before taking some recovery action.

1954 An IPP object MUST recover from an errant client that does not supply a send operation, sometime after  
1955 the minimum time interval specified by the Printer object's "multiple-operation-time-out" attribute. Such  
1956 recovery MAY include any of the following or other recovery actions:

- 1957 1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', add the  
1958 'aborted-by-system' value to the job's "job-state-reasons" attribute (see section 4.3.8), and clean up  
1959 all resources associated with the Job. In this case, if another send operation is finally received, the  
1960 Printer responds with a "client-error-not-possible" or "client-error-not-found" depending on  
1961 whether or not the Job object is still around when the send operation finally arrives.
- 1962 2. Assume that the last send operation received was in fact the last document (as if the "last-document"  
1963 flag had been set to 'true'), close the Job object, and proceed to process it (i.e., move the Job's state  
1964 to 'pending').
- 1965 3. Assume that the last send operation received was in fact the last document, close the Job, but move it  
1966 to the 'pending-held' and add the 'submission-interrupted' value to the job's "job-state-reasons"  
1967 attribute (see section 4.3.8). This action allows the user or an operator to determine whether to  
1968 continue processing the Job by moving it back to the 'pending' state using the Release-Job operation  
1969 (see section 3.3.6) or to cancel the job using the Cancel-Job operation (see section 3.3.3).

1970  
1971 Each implementation is free to decide the "best" action to take depending on local policy, whether any  
1972 documents have been added, whether the implementation spools jobs or not, and/or any other piece of  
1973 information available to it. If the choice is to abort the Job object, it is possible that the Job object may  
1974 already have been processed to the point that some media sheet pages have been printed.

1975 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
1976 owner (as determined in the Create-Job operation) or an operator or administrator of the Printer object (see  
1977 Sections 1 and 8.5). Otherwise, the IPP object MUST reject the operation and return: 'client-error-  
1978 forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

### 1979 3.3.1.1 Send-Document Request

1980 The following attribute sets are part of the Send-Document Request:

#### 1981 Group 1: Operation Attributes

1982 Natural Language and Character Set:

1983 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.

1984

## 1985 Target:

1986 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
1987 attribute(s) which define the target for this operation as described in section 3.1.5.  
1988

## 1989 Requesting User Name:

1990 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1991 described in section 8.3.  
1992

## 1993 "document-name" (name(MAX)):

1994 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1995 contains the client supplied document name. The document name MAY be different than the Job  
1996 name. It might be helpful, but NEED NOT be unique across multiple documents in the same Job.  
1997 Typically, the client software automatically supplies the document name on behalf of the end user  
1998 by using a file name or an application generated name. See the description of the "document-name"  
1999 operation attribute in the Print-Job Request (section 3.2.1.1) for more information about this  
2000 attribute.  
2001

## 2002 "compression" (type3 keyword)

2003 See the description of "compression" for the Print-Job operation in Section 3.2.1.1.  
2004

## 2005 "document-format" (mimeMediaType) :

2006 See the description of "document-format" for the Print-Job operation in Section 3.2.1.1.  
2007

## 2008 "document-natural-language" (naturalLanguage):

2009 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
2010 attribute. This attribute specifies the natural language of the document for those document-formats  
2011 that require a specification of the natural language in order to image the document unambiguously.  
2012 There are no particular values required for the Printer object to support.  
2013

## 2014 "last-document" (boolean):

2015 The client MUST supply this attribute. The Printer object MUST support this attribute. It is a  
2016 boolean flag that is set to 'true' if this is the last document for the Job, 'false' otherwise.  
2017

## 2018 Group 2: Document Content

2019 The client MUST supply the document data if the "last-document" flag is set to 'false'. However,  
2020 since a client might not know that the previous document sent with a Send-Document (or Send-  
2021 URI) operation was the last document (i.e., the "last-document" attribute was set to 'false'), it is legal  
2022 to send a Send-Document request with no document data where the "last-document" flag is set to  
2023 'true'. Such a request MUST NOT increment the value of the Job object's "number-of-documents"  
2024 attribute, since no real document was added to the job. It is not an error for a client to submit a job  
2025 with no actual document data, i.e., only a single Create-Job and Send-Document request with a  
2026 "last-document" operation attribute set to 'true' with no document data.

### 2027 **3.3.1.2 Send-Document Response**

2028 The following sets of attributes are part of the Send-Document Response:

#### 2029 Group 1: Operation Attributes

##### 2030 Status Message:

2031 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
2032 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
2033 attribute as described in sections 13 and 3.1.6.

##### 2035 Natural Language and Character Set:

2036 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.

#### 2038 Group 2: Unsupported Attributes

2039 See section 3.1.7 for details on returning Unsupported Attributes.

#### 2040 Group 3: Job Object Attributes

2041 This is the same set of attributes as described in the Print-Job response (see section 3.2.1.2).

### 2043 **3.3.2 Send-URI Operation**

2044 This OPTIONAL operation is identical to the Send-Document operation (see section 3.3.1) except that a  
2045 client MUST supply a URI reference ("document-uri" operation attribute) rather than the document data  
2046 itself. If a Printer object supports this operation, clients can use both Send-URI or Send-Document  
2047 operations to add new documents to an existing multi-document Job object. However, if a client needs to  
2048 indicate that the previous Send-URI or Send-Document was the last document, the client MUST use the  
2049 Send-Document operation with no document data and the "last-document" flag set to 'true' (rather than  
2050 using a Send-URI operation with no "document-uri" operation attribute).

2051 If a Printer object supports this operation, it MUST also support the Print-URI operation (see section 3.2.2).

2052 The Printer object MUST validate the syntax and URI scheme of the supplied URI before returning a  
2053 response, just as in the Print-URI operation. The IPP Printer MAY validate the accessibility of the  
2054 document as part of the operation or subsequently (see section 3.2.2).

### 2055 **3.3.3 Cancel-Job Operation**

2056 This REQUIRED operation allows a client to cancel a Print Job from the time the job is created up to the  
2057 time it is completed, canceled, or aborted. Since a Job might already be printing by the time a Cancel-Job is  
2058 received, some media sheet pages might be printed before the job is actually terminated.

2059 The IPP object MUST accept or reject the request based on the job's current state and transition the job to  
2060 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'

2061 Rule 1: If the implementation requires some measurable time to cancel the job in the 'processing' or  
 2062 'processing-stopped' job states, the IPP object MUST add the 'processing-to-stop-point' value to the job's  
 2063 "job-state-reasons" attribute and then transition the job to the 'canceled' state when the processing ceases  
 2064 (see section 4.3.8).

2065 Rule 2: If the Job object already has the 'processing-to-stop-point' value in its "job-state-reasons" attribute,  
 2066 then the Printer object MUST reject a Cancel-Job operation.

2067 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
 2068 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
 2069 object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
 2070 'client-error-not-authorized' as appropriate.

### 2071 3.3.3.1 Cancel-Job Request

2072 The following groups of attributes are part of the Cancel-Job Request:

#### 2073 Group 1: Operation Attributes

2074 Natural Language and Character Set:

2075 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.  
 2076

2077 Target:

2078 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
 2079 attribute(s) which define the target for this operation as described in section 3.1.5.  
 2080

2081 Requesting User Name:

2082 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
 2083 described in section 8.3.  
 2084

2085 "message" (text(127)):

2086 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
2087 attribute. It is a message to the operator. This "message" attribute is not the same as the "job-  
2088 message-from-operator" attribute. That attribute is used to report a message from the operator to the  
2089 end user that queries that attribute. This "message" operation attribute is used to send a message  
2090 from the client to the operator along with the operation request. It is an implementation decision of  
2091 how or where to display this message to the operator (if at all).  
2092

### 2093 3.3.3.2 Cancel-Job Response

2094 The following sets of attributes are part of the Cancel-Job Response:

#### 2095 Group 1: Operation Attributes

##### 2096 Status Message:

2097 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
2098 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
2099 attribute as described in sections 13 and 3.1.6.  
2100

##### 2101 Natural Language and Character Set:

2102 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.  
2103

#### 2104 Group 2: Unsupported Attributes

2105 See section 3.1.7 for details on returning Unsupported Attributes.  
2106

2107 Once a successful response has been sent, the implementation guarantees that the Job will eventually end up  
2108 in the 'canceled' state. Between the time of the Cancel-Job operation is accepted and when the job enters the  
2109 'canceled' job-state (see section 4.3.7), the "job-state-reasons" attribute SHOULD contain the 'processing-to-  
2110 stop-point' value which indicates to later queries that although the Job might still be 'processing', it will  
2111 eventually end up in the 'canceled' state, not the 'completed' state.

### 2112 3.3.4 Get-Job-Attributes Operation

2113 This REQUIRED operation allows a client to request the values of attributes of a Job object and it is almost  
2114 identical to the Get-Printer-Attributes operation (see section 3.2.5). The only differences are that the  
2115 operation is directed at a Job object rather than a Printer object, there is no "document-format" operation  
2116 attribute used when querying a Job object, and the returned attribute group is a set of Job object attributes  
2117 rather than a set of Printer object attributes.

2118 For Jobs, the possible names of attribute groups are:

- 2119 - 'job-template': the subset of the Job Template attributes that apply to a Job object (the first column of  
2120 the table in Section 4.2) that the implementation supports for Job objects.



- 2121 - 'job-description': the subset of the Job Description attributes specified in Section 4.3 that the  
2122 implementation supports for Job objects.
- 2123 - 'all': the special group 'all' that includes all attributes that the implementation supports for Job objects.  
2124

2125 Since a client MAY request specific attributes or named groups, there is a potential that there is some  
2126 overlap. For example, if a client requests, 'job-name' and 'job-description', the client is actually requesting  
2127 the "job-name" attribute once by naming it explicitly, and once by inclusion in the 'job-description' group.  
2128 In such cases, the Printer object NEED NOT return the attribute only once in the response even if it is  
2129 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

2130 It is NOT REQUIRED that a Job object support all attributes belonging to a group (since some attributes  
2131 are OPTIONAL). However it is REQUIRED that each Job object support all group names.

### 2132 3.3.4.1 Get-Job-Attributes Request

2133 The following groups of attributes are part of the Get-Job-Attributes Request when the request is directed at  
2134 a Job object:

#### 2135 Group 1: Operation Attributes

##### 2136 Natural Language and Character Set:

2137 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.  
2138

##### 2139 Target:

2140 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
2141 attribute(s) which define the target for this operation as described in section 3.1.5.  
2142

##### 2143 Requesting User Name:

2144 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
2145 described in section 8.3.  
2146

##### 2147 "requested-attributes" (1setOf keyword) :

2148 The client OPTIONALLY supplies this attribute. The IPP object MUST support this attribute. It is  
2149 a set of attribute names and/or attribute group names in whose values the requester is interested. If  
2150 the client omits this attribute, the IPP object MUST respond as if this attribute had been supplied  
2151 with a value of 'all'.  
2152

### 2153 3.3.4.2 Get-Job-Attributes Response

2154 The Printer object returns the following sets of attributes as part of the Get-Job-Attributes Response:

#### 2155 Group 1: Operation Attributes

#### Status Message:

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

#### Natural Language and Character Set:

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

#### Group 2: Unsupported Attributes

See section 3.1.7 for details on returning Unsupported Attributes.

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

#### Group 3: Job Object Attributes

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

### 3.3.5 Hold-Job Operation

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

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

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

'pending'	'pending-held'	'successful-ok' See 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'

2190 Rule 1: If the implementation supports multiple reasons for a job to be in the 'pending-held' state, the IPP  
2191 object MUST add the 'job-hold-until-specified' value to the job's "job-state-reasons" attribute.

2192 Rule 2: If the IPP object supports the "job-hold-until" operation attribute, but the specified time period has  
2193 already started (or is the 'no-hold' value) and there are no other reasons to hold the job, the IPP object  
2194 MUST make the job be a candidate for processing immediately (see Section 4.2.2) by putting the job in the  
2195 'pending' state.

2196 Note: In order to keep the Hold-Job operation simple, such a request is rejected when the job is in the  
2197 'processing' or 'processing-stopped' states. If an operation is needed to hold jobs while in these states, it will  
2198 be added as an additional operation, rather than overloading the Hold-Job operation. Then it is clear to  
2199 clients by querying the Printer object's "operations-supported" (see Section 4.4.15) and the Job object's  
2200 "job-state" (see Section 4.3.7) attributes which operations are possible.

2201 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
2202 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
2203 object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
2204 'client-error-not-authorized' as appropriate.

### 2205 3.3.5.1 Hold-Job Request

2206 The groups and operation attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with the  
2207 addition of the following Group 1 Operation attribute:

2208 "job-hold-until" (type3 keyword | name(MAX)):

2209 The client OPTIONALLY supplies this Operation attribute. The IPP object MUST support this  
2210 operation attribute in a Hold-Job request, if it supports the "job-hold-until" Job template attribute in  
2211 create operations. See section 4.2.2. The IPP object SHOULD support the "job-hold-until" Job  
2212 Template attribute for use in job create operations with at least the 'indefinite' value, if it supports  
2213 the Hold-Job operation. Otherwise, a client cannot create a job and hold it immediately (without  
2214 picking some supported time period in the future).

2215 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP  
2216 object copies the supplied operation attribute to the Job object, replacing the job's previous "job-

2217 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied  
2218 named time period.

2219 If supplied, but either the "job-hold-until" Operation attribute itself or the value supplied is not  
2220 supported, the IPP object accepts the request, returns the unsupported attribute or value in the  
2221 Unsupported Attributes Group according to section 3.1.7, returns the 'successful-ok-ignored-or-  
2222 substituted-attributes, and holds the job indefinitely until a client performs a subsequent Release-Job  
2223 operation.

2224 If the client (1) supplies a value that specifies a time period that has already started or the 'no-hold'  
2225 value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until" operation  
2226 attribute and there are no other reasons to hold the job, the IPP object MUST accept the operation  
2227 and make the job be a candidate for processing immediately (see Section 4.2.2).

2228 If the client does not supply a "job-hold-until" Operation attribute in the request, the IPP object  
2229 MUST populate the job object with a "job-hold-until" attribute with the 'indefinite' value (if IPP  
2230 object supports the "job-hold-until" attribute) and hold the job indefinitely, until a client performs a  
2231 Release-Job operation.

### 2232 **3.3.5.2 Hold-Job Response**

2233 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

### 2234 **3.3.6 Release-Job Operation**

2235 This OPTIONAL operation allows a client to release a previously held job so that it is again eligible for  
2236 scheduling. If the Hold-Job operation is supported, then the Release-Job operation MUST be supported,  
2237 and vice-versa.

2238 This operation removes the "job-hold-until" job attribute, if present, from the job object that had been  
2239 supplied in the create or most recent Hold-Job or Restart-Job operation and removes its effect on the job.  
2240 The IPP object MUST remove the 'job-hold-until-specified' value from the job's "job-state-reasons"  
2241 attribute, if present. See section 4.3.8.

2242 The IPP object MUST accept or reject the request based on the job's current state and transition the job to  
2243 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'

2244 Rule 1: If there are other reasons to keep the job in the 'pending-held' state, such as 'resources-are-not-  
2245 ready', the job remains in the 'pending-held' state. Thus the 'pending-held' state is not just for jobs that have  
2246 the 'job-hold-until' applied to them, but are for any reason to keep the job from being a candidate for  
2247 scheduling and processing, such as 'resources-are-not-ready'. See the "job-hold-until" attribute (section  
2248 4.2.2).

2249 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
2250 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
2251 object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
2252 'client-error-not-authorized' as appropriate.

2253 The Release-Job Request and Release-Job Response have the same attribute groups and attributes as the  
2254 Cancel-Job operation (see section 3.3.3.1 and 3.3.3.2).

### 2255 3.3.7 Restart-Job Operation

2256 This OPTIONAL operation allows a client to restart a job that is retained in the queue after processing has  
2257 completed (see section 4.3.7.2).

2258 The job is moved to the 'pending' or 'pending-held' job state and restarts at the beginning on the same IPP  
2259 Printer object with the same attribute values. If any of the documents in the job were passed by reference  
2260 (Print-URI or Send-URI), the Printer MUST re-fetch the data, since the semantics of Restart-Job are to  
2261 repeat all Job processing. The Job Description attributes that accumulate job progress, such as "job-  
2262 impressions-completed", "job-media-sheets-completed", and "job-k-octets-processed", MUST be reset to 0  
2263 so that they give an accurate record of the job from its restart point. The job object MUST continue to use  
2264 the same "job-uri" and "job-id" attribute values.

2265 Note: If in the future an operation is needed that does not reset the job progress attributes, then a new  
2266 operation will be defined which makes a copy of the job, assigns a new "job-uri" and "job-id" to the copy  
2267 and resets the job progress attributes in the new copy only.

2268 The IPP object MUST accept or reject the request based on the job's current state, transition the job to the  
2269 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' <u>or</u> 'pending-held'	'successful-ok' - job is started over.
'completed'	'completed'	'client-error-not-possible' - see Rule 1
'canceled'	'pending' <u>or</u> 'pending-held'	'successful-ok' - job is started over.
'canceled'	'canceled'	'client-error-not-possible' - see Rule 1
'aborted'	'pending' <u>or</u> 'pending-held'	'successful-ok' - job is started over.
'aborted'	'aborted'	'client-error-not-possible' - see Rule 1

2270

2271 Rule 1: If the Job Retention Period has expired for the job in this state, then the IPP object rejects the  
2272 operation. See section 4.3.7.2.

2273 Note: In order to prevent a user from inadvertently restarting a job in the middle, the Restart-Job request is  
2274 rejected when the job is in the 'processing' or 'processing-stopped' states. If in the future an operation is  
2275 needed to hold or restart jobs while in these states, it will be added as an additional operation, rather than  
2276 overloading the Restart-Job operation, so that it is clear that the user intended that the current job not be  
2277 completed.

2278 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
2279 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
2280 object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
2281 'client-error-not-authorized' as appropriate.

### 2282 3.3.7.1 Restart-Job Request

2283 The groups and attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with the addition of  
2284 the following Group 1 Operation attribute:

2285 "job-hold-until" (type3 keyword | name(MAX)):

2286 The client OPTIONALLY supplies this attribute. The IPP object MUST support this Operation  
2287 attribute in a Restart-Job request, if it supports the "job-hold-until" Job Template attribute in create  
2288 operations. See section 4.2.2. Otherwise, the IPP object NEED NOT support the "job-hold-until"  
2289 Operation attribute in a Restart-Job request.

2290 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP  
2291 object copies the supplied Operation attribute to the Job object, replacing the job's previous "job-  
2292 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied  
2293 named time period. See section 4.2.2.

2294 If supplied, but the value is not supported, the IPP object accepts the request, returns the  
2295 unsupported attribute or value in the Unsupported Attributes Group according to section 3.1.7,

2296 returns the 'successful-ok-ignored-or-substituted-attributes' status code, and holds the job  
2297 indefinitely until a client performs a subsequent Release-Job operation.

2298 If supplied, but the "job-hold-until" Operation attribute itself is not supported, the IPP object accepts  
2299 the request, returns the unsupported attribute with the out-of-band 'unsupported' value in the  
2300 Unsupported Attributes Group according to section 3.1.7, returns the 'successful-ok-ignored-or-  
2301 substituted-attributes' status code, and restarts the job, i.e., ignores the "job-hold-until" attribute.

2302 If the client (1) supplies a value that specifies a time period that has already started or the 'no-hold'  
2303 value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until" operation  
2304 attribute and there are no other reasons to hold the job, the IPP object makes the job a candidate for  
2305 processing immediately (see Section 4.2.2).

2306 If the client does not supply a "job-hold-until" operation attribute in the request, the IPP object  
2307 removes the "job-hold-until" attribute, if present, from the job. If there are no other reasons to hold  
2308 the job, the Restart-Job operation makes the job a candidate for processing immediately (see Section  
2309 4.2.2).

### 2310 3.3.7.2 Restart-Job Response

2311 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

2312 Note: In the future an OPTIONAL Modify-Job or Set-Job-Attributes operation may be specified that  
2313 allows the client to modify other attributes before releasing the restarted job.

## 2314 4. Object Attributes

2315 This section describes the attributes with their corresponding attribute syntaxes and values that are part of  
2316 the IPP model. The sections below show the objects and their associated attributes which are included  
2317 within the scope of this protocol. Many of these attributes are derived from other relevant documents:

- 2318 - Document Printing Application (DPA) [ISO10175]
- 2319 - RFC 1759 Printer MIB [RFC1759]

2320 Each attribute is uniquely identified in this document using a "keyword" (see section 12.2.1) which is the  
2321 name of the attribute. The keyword is included in the section header describing that attribute.

2322 Note: Not only are keywords used to identify attributes, but one of the attribute syntaxes described below is  
2323 "keyword" so that some attributes have keyword values. Therefore, these attributes are defined as having  
2324 an attribute syntax that is a set of keywords.



## 2325 4.1 Attribute Syntaxes

2326 This section defines the basic attribute syntax types that all clients and IPP objects MUST be able to accept  
2327 in responses and accept in requests, respectively. Each attribute description in sections 3 and 4 includes the  
2328 name of attribute syntax(es) in the heading (in parentheses). A conforming implementation of an attribute  
2329 MUST include the semantics of the attribute syntax(es) so identified. Section 6.3 describes how the  
2330 protocol can be extended with new attribute syntaxes.

2331 The attribute syntaxes are specified in the following sub-sections, where the sub-section heading is the  
2332 keyword name of the attribute syntax inside the single quotes. In operation requests and responses each  
2333 attribute value MUST be represented as one of the attribute syntaxes specified in the sub-section heading  
2334 for the attribute. In addition, the value of an attribute in a response (but not in a request) MAY be one of  
2335 the "out-of-band" values whose special encoding rules are defined in the "Encoding and Transport"  
2336 document [IPP-PRO]. Standard "out-of-band" values are:

2337 'unknown': The attribute is supported by the IPP object, but the value is unknown to the IPP object for  
2338 some reason.

2339 'unsupported': The attribute is unsupported by the IPP object. This value MUST be returned only as the  
2340 value of an attribute in the Unsupported Attributes Group.

2341 'no-value': The attribute is supported by the Printer object, but the administrator has not yet configured a  
2342 value.

2343

2344 All attributes in a request MUST have one or more values as defined in Sections 4.2 to 4.4. Thus clients  
2345 MUST NOT supply attributes with "out-of-band" values. All attributes in a response MUST have one or  
2346 more values as defined in Sections 4.2 to 4.4 or a single "out-of-band" value.

2347 Most attributes are defined to have a single attribute syntax. However, a few attributes (e.g., "job-sheet",  
2348 "media", "job-hold-until") are defined to have several attribute syntaxes, depending on the value. These  
2349 multiple attribute syntaxes are separated by the "|" character in the sub-section heading to indicate the  
2350 choice. Since each value MUST be tagged as to its attribute syntax in the protocol, a single-valued attribute  
2351 instance may have any one of its attribute syntaxes and a multi-valued attribute instance may have a mixture  
2352 of its defined attribute syntaxes.

### 2353 4.1.1 'text'

2354 A text attribute is an attribute whose value is a sequence of zero or more characters encoded in a maximum  
2355 of 1023 ('MAX') octets. MAX is the maximum length for each value of any text attribute. However, if an  
2356 attribute will always contain values whose maximum length is much less than MAX, the definition of that  
2357 attribute will include a qualifier that defines the maximum length for values of that attribute. For example:  
2358 the "printer-location" attribute is specified as "printer-location (text(127))". In this case, text values for  
2359 "printer-location" MUST NOT exceed 127 octets; if supplied with a longer text string via some external  
2360 interface (other than the protocol), implementations are free to truncate to this shorter length limitation.

2361 In this document, all text attributes are defined using the 'text' syntax. However, 'text' is used only for  
2362 brevity; the formal interpretation of 'text' is: 'textWithoutLanguage | textWithLanguage'. That is, for any  
2363 attribute defined in this document using the 'text' attribute syntax, all IPP objects and clients MUST support  
2364 both the 'textWithoutLanguage' and 'textWithLanguage' attribute syntaxes. However, in actual usage and  
2365 protocol execution, objects and clients accept and return only one of the two syntax per attribute. The  
2366 syntax 'text' never appears "on-the-wire".

2367 Both 'textWithoutLanguage' and 'textWithLanguage' are needed to support the real world needs of  
2368 interoperability between sites and systems that use different natural languages as the basis for human  
2369 communication. Generally, one natural language applies to all text attributes in a given request or response.  
2370 The language is indicated by the "attributes-natural-language" operation attribute defined in section 3.1.4 or  
2371 "attributes-natural-language" job attribute defined in section 4.3.20, and there is no need to identify the  
2372 natural language for each text string on a value-by-value basis. In these cases, the attribute syntax  
2373 'textWithoutLanguage' is used for text attributes. In other cases, the client needs to supply or the Printer  
2374 object needs to return a text value in a natural language that is different from the rest of the text values in  
2375 the request or response. In these cases, the client or Printer object uses the attribute syntax  
2376 'textWithLanguage' for text attributes (this is the Natural Language Override mechanism described in  
2377 section 3.1.4).

2378 The 'textWithoutLanguage' and 'textWithLanguage' attribute syntaxes are described in more detail in the  
2379 following sections.

#### 2380 4.1.1.1 'textWithoutLanguage'

2381 The 'textWithoutLanguage' syntax indicates a value that is sequence of zero or more characters encoded in a  
2382 maximum of 1023 (MAX) octets. Text strings are encoded using the rules of some charset. The Printer  
2383 object MUST support the UTF-8 charset [RFC2279] and MAY support additional charsets to represent  
2384 'text' values, provided that the charsets are registered with IANA [IANA-CS]. See Section 4.1.7 for the  
2385 definition of the 'charset' attribute syntax, including restricted semantics and examples of charsets.

#### 2386 4.1.1.2 'textWithLanguage'

2387 The 'textWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2388 'textWithoutLanguage' part encoded in a maximum of 1023 (MAX) octets plus an additional  
2389 'naturalLanguage' (see section 4.1.8) part that overrides the natural language in force. The  
2390 'naturalLanguage' part explicitly identifies the natural language that applies to the text part of that value and  
2391 that value alone. For any give text attribute, the 'textWithoutLanguage' part is limited to the maximum  
2392 length defined for that 'text' attribute, but-and the 'naturalLanguage' part is always limited to 63 (additional)  
2393 octets. Using the 'textWithLanguage' attribute syntax rather than the normal 'textWithoutLanguage' syntax  
2394 is the so-called Natural Language Override mechanism and MUST be supported by all IPP objects and  
2395 clients.

2396 If the attribute is multi-valued (1setOf text), then the 'textWithLanguage' attribute syntax MUST be used to  
2397 explicitly specify each attribute value whose natural language needs to be overridden. Other values in a

2398 multi-valued 'text' attribute in a request or a response revert to the natural language of the operation  
2399 attribute.

2400 In a create request, the Printer object MUST accept and store with the Job object any natural language in the  
2401 "attributes-natural-language" operation attribute, whether the Printer object supports that natural language  
2402 or not. Furthermore, the Printer object MUST accept and store any 'textWithLanguage' attribute value,  
2403 whether the Printer object supports that natural language or not. These requirements are independent of the  
2404 value of the "ipp-attribute-fidelity" operation attribute that the client MAY supply.

2405 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2406 indicating English, but the value of the "job-name" attribute is in French, the client MUST use the  
2407 'textWithLanguage' attribute syntax with the following two values:

2408 'fr': Natural Language Override indicating French

2409 'Rapport Mensuel': the job name in French

2410

2411 See the "Encoding and Transport" document [IPP-PRO] for a detailed example of the 'textWithLanguage'  
2412 attribute syntax.

#### 2413 4.1.2 'name'

2414 This syntax type is used for user-friendly strings, such as a Printer name, that, for humans, are more  
2415 meaningful than identifiers. Names are never translated from one natural language to another. The 'name'  
2416 attribute syntax is essentially the same as 'text', including the REQUIRED support of UTF-8 except that the  
2417 sequence of characters is limited so that its encoded form MUST NOT exceed 255 (MAX) octets.

2418 Also like 'text', 'name' is really an abbreviated notation for either 'nameWithoutLanguage' or  
2419 'nameWithLanguage'. That is, all IPP objects and clients MUST support both the 'nameWithoutLanguage'  
2420 and 'nameWithLanguage' attribute syntaxes. However, in actual usage and protocol execution, objects and  
2421 clients accept and return only one of the two syntax per attribute. The syntax 'name' never appears "on-the-  
2422 wire".

2423 Only the 'text' and 'name' attribute syntaxes permit the Natural Language Override mechanism.

2424 Some attributes are defined as 'type3 keyword | name'. These attributes support values that are either type3  
2425 keywords or names. This dual-syntax mechanism enables a site administrator to extend these attributes to  
2426 legally include values that are locally defined by the site administrator. Such names are not registered with  
2427 IANA.

##### 2428 4.1.2.1 'nameWithoutLanguage'

2429 The nameWithoutLanguage' syntax indicates a value that is sequence of zero or more characters encoded in  
2430 a maximum of 255 (MAX) octets~~so that its encoded form does not exceed MAX octets.~~

#### 2431 4.1.2.2 'nameWithLanguage'

2432 The 'nameWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2433 'nameWithoutLanguage' part encoded in a maximum of 1023 (MAX) octets plus an additional  
2434 'naturalLanguage' (see section 4.1.8) part that overrides the natural language in force. The  
2435 'naturalLanguage' part explicitly identifies the natural language that applies to that name value and that  
2436 name value alone. For any give text attribute, the 'textWithoutLanguage' part is limited to the maximum  
2437 length defined for that 'text' attribute, and the 'naturalLanguage' part is always limited to 63 (additional)  
2438 octets. Using the 'textWithLanguage' attribute syntax rather than the normal 'textWithoutLanguage' syntax  
2439 is the so-called Natural Language Override mechanism and MUST be supported by all IPP objects and  
2440 clients.

2441 The 'nameWithLanguage' attribute syntax behaves the same as the 'textWithLanguage' syntax. Using the  
2442 'textWithLanguage' attribute syntax rather than the normal 'textWithoutLanguage' syntax is the so-called  
2443 Natural Language Override mechanism and MUST be supported by all IPP objects and clients. If a name is  
2444 in a language that is different than the rest of the object or operation, then this 'nameWithLanguage' syntax  
2445 is used rather than the generic 'nameWithoutLanguage' syntax.

2446 If the attribute is multi-valued (1setOf text), then the 'nameWithLanguage' attribute syntax MUST be used  
2447 to explicitly specify each attribute value whose natural language needs to be overridden. Other values in a  
2448 multi-valued 'name' attribute in a request or a response revert to the natural language of the operation  
2449 attribute.

2450 In a create request, the Printer object MUST accept and store with the Job object any natural language in the  
2451 "attributes-natural-language" operation attribute, whether the Printer object supports that natural language  
2452 or not. Furthermore, the Printer object MUST accept and store any 'nameWithLanguage' attribute value,  
2453 whether the Printer object supports that natural language or not. These requirements are independent of the  
2454 value of the "ipp-attribute-fidelity" operation attribute that the client MAY supply.

2455 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2456 indicating English, but the "printer-name" attribute is in German, the client MUST use the  
2457 'nameWithLanguage' attribute syntax as follows:

2458 'de': Natural Language Override indicating German

2459 'Farbdrucker': the Printer name in German

2460

#### 2461 4.1.2.3 Matching 'name' attribute values

2462 For purposes of matching two 'name' attribute values for equality, such as in job validation (where a client-  
2463 supplied value for attribute "xxx" is checked to see if the value is among the values of the Printer object's  
2464 corresponding "xxx-supported" attribute), the following match rules apply:

2465 1. 'keyword' values never match 'name' values.

2466 2. 'name' (nameWithoutLanguage and nameWithLanguage) values match if (1) the name parts  
2467 match and (2) the Associated Natural-Language parts (see section 3.1.4.1) match. The matching  
2468 rules are:

2469 a. the name parts match if the two names are identical character by character, except it is  
2470 RECOMMENDED that case be ignored. For example: 'Ajax-letter-head-white' MUST  
2471 match 'Ajax-letter-head-white' and SHOULD match 'ajax-letter-head-white' and 'AJAX-  
2472 LETTER-HEAD-WHITE'.

2473 b. the Associated Natural-Language parts match if the shorter of the two meets the syntactic  
2474 requirements of RFC 1766 [RFC1766] and matches byte for byte with the longer. For  
2475 example, 'en' matches 'en', 'en-us' and 'en-gb', but matches neither 'fr' nor 'e'.

#### 2476 4.1.3 'keyword'

2477 The 'keyword' attribute syntax is a sequence of characters, length: 1 to 255, containing only the US-ASCII  
2478 [ASCII] encoded values for lowercase letters ("a" - "z"), digits ("0" - "9"), hyphen ("-"), dot ("."), and  
2479 underscore ("\_"). The first character MUST be a lowercase letter. Furthermore, keywords MUST be in  
2480 U.S. English.

2481 This syntax type is used for enumerating semantic identifiers of entities in the abstract protocol, i.e., entities  
2482 identified in this document. Keywords are used as attribute names or values of attributes. Unlike 'text' and  
2483 'name' attribute values, 'keyword' values MUST NOT use the Natural Language Override mechanism, since  
2484 they MUST always be US-ASCII and U.S. English.

2485 Keywords are for use in the protocol. A user interface will likely provide a mapping between protocol  
2486 keywords and displayable user-friendly words and phrases which are localized to the natural language of  
2487 the user. While the keywords specified in this document MAY be displayed to users whose natural  
2488 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since  
2489 the user interface is outside the scope of this document.

2490 In the definition for each attribute of this syntax type, the full set of defined keyword values for that  
2491 attribute are listed.

2492 When a keyword is used to represent an attribute (its name), it MUST be unique within the full scope of all  
2493 IPP objects and attributes. When a keyword is used to represent a value of an attribute, it MUST be unique  
2494 just within the scope of that attribute. That is, the same keyword MUST NOT be used for two different  
2495 values within the same attribute to mean two different semantic ideas. However, the same keyword MAY  
2496 be used across two or more attributes, representing different semantic ideas for each attribute. Section 6.1  
2497 describes how the protocol can be extended with new keyword values. Examples of attribute name  
2498 keywords:

2499 "job-name"  
2500 "attributes-charset"

2501

2502 Note: This document uses "type1", "type2", and "type3" prefixes to the "keyword" basic syntax to indicate  
2503 different levels of review for extensions (see section 6.1).

#### 2504 **4.1.4 'enum'**

2505 The 'enum' attribute syntax is an enumerated integer value that is in the range from 1 to  $2^{*}31 - 1$  (MAX).  
2506 Each value has an associated 'keyword' name. In the definition for each attribute of this syntax type, the full  
2507 set of possible values for that attribute are listed. This syntax type is used for attributes for which there are  
2508 enum values assigned by other standards, such as SNMP MIBs. A number of attribute enum values in this  
2509 document are also used for corresponding attributes in other standards [RFC1759]. This syntax type is not  
2510 used for attributes to which the administrator may assign values. Section 6.1 describes how the protocol  
2511 can be extended with new enum values.

2512 Enum values are for use in the protocol. A user interface will provide a mapping between protocol enum  
2513 values and displayable user-friendly words and phrases which are localized to the natural language of the  
2514 user. While the enum symbols specified in this document MAY be displayed to users whose natural  
2515 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since  
2516 the user interface is outside the scope of this document.

2517 Note: SNMP MIBs use '2' for 'unknown' which corresponds to the IPP "out-of-band" value 'unknown'. See  
2518 the description of the "out-of-band" values at the beginning of Section 4.1. Therefore, attributes of type  
2519 'enum' start at '3'.

2520 Note: This document uses "type1", "type2", and "type3" prefixes to the "enum" basic syntax to indicate  
2521 different levels of review for extensions (see section 6.1).

#### 2522 **4.1.5 'uri'**

2523 The 'uri' attribute syntax is any valid Uniform Resource Identifier or URI [RFC2396]. Most often, URIs are  
2524 simply Uniform Resource Locators or URLs. The maximum length of URIs used as values of IPP  
2525 attributes is 1023 octets. Although most other IPP attribute syntax types allow for only lower-cased values,  
2526 this attribute syntax type conforms to the case-sensitive and case-insensitive rules specified in [RFC2396].  
2527 See also [IPP-IIG] for a discussion of case in URIs.

#### 2528 **4.1.6 'uriScheme'**

2529 The 'uriScheme' attribute syntax is a sequence of characters representing a URI scheme according to RFC  
2530 2396 [RFC2396]. Though RFC 2396 requires that the values be case-insensitive, IPP requires all lower  
2531 case values in IPP attributes to simplify comparing by IPP clients and Printer objects.

2532 Standard values for this syntax type are the following keywords:

2533 'ipp': for IPP schemed URIs (e.g., "ipp:...")  
2534 'http': for HTTP schemed URIs (e.g., "http:...")  
2535 'https': for use with HTTPS schemed URIs (e.g., "https:...") (not on IETF standards track)  
2536 'ftp': for FTP schemed URIs (e.g., "ftp:...")



2537 'mailto': for SMTP schemed URIs (e.g., "mailto:...")  
2538 'file': for file schemed URIs (e.g., "file:...")

2539

2540 A Printer object MAY support any URI 'scheme' that has been registered with IANA [IANA-MT]. The  
2541 maximum length of URI 'scheme' values used to represent IPP attribute values is 63 octets.

#### 2542 4.1.7 'charset'

2543 The 'charset' attribute syntax is a standard identifier for a charset. A charset is a coded character set and  
2544 encoding scheme. Charsets are used for labeling certain document contents and 'text' and 'name' attribute  
2545 values. The syntax and semantics of this attribute syntax are specified in RFC 2046 [RFC2046] and  
2546 contained in the IANA character-set Registry [IANA-CS] according to the IANA procedures [RFC2278].  
2547 Though RFC 2046 requires that the values be case-insensitive US-ASCII, IPP requires all lower case values  
2548 in IPP attributes to simplify comparing by IPP clients and Printer objects. When a character-set in the  
2549 IANA registry has more than one name (alias), the name labeled as "(preferred MIME name)", if present,  
2550 MUST be used.

2551 The maximum length of 'charset' values used to represent IPP attribute values is 63 octets.

2552 Some examples are:

2553 'utf-8': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as the UTF-8  
2554 [RFC2279] transfer encoding scheme in which US-ASCII is a subset charset.  
2555 'us-ascii': 7-bit American Standard Code for Information Interchange (ASCII), ANSI X3.4-1986  
2556 [ASCII]. That standard defines US-ASCII, but RFC 2045 [RFC2045] eliminates most of the control  
2557 characters from conformant usage in MIME and IPP.  
2558 'iso-8859-1': 8-bit One-Byte Coded Character Set, Latin Alphabet Nr 1 [ISO8859-1]. That standard  
2559 defines a coded character set that is used by Latin languages in the Western Hemisphere and  
2560 Western Europe. US-ASCII is a subset charset.  
2561 ~~'iso-10646-ucs-2': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as two~~  
2562 ~~octets (UCS-2), with the high order octet of each pair coming first (so-called Big-Endian integer).~~

2564 Some attribute descriptions MAY place additional requirements on charset values that may be used, such as  
2565 REQUIRED values that MUST be supported or additional restrictions, such as requiring that the charset  
2566 have US-ASCII as a subset charset.

#### 2567 4.1.8 'naturalLanguage'

2568 The 'naturalLanguage' attribute syntax is a standard identifier for a natural language and optionally a  
2569 country. The values for this syntax type are defined by RFC 1766 [RFC1766]. Though RFC 1766 requires  
2570 that the values be case-insensitive US-ASCII, IPP requires all lower case to simplify comparing by IPP  
2571 clients and Printer objects. Examples include:

2572 'en': for English  
2573 'en-us': for US English



2574 'fr': for French  
2575 'de': for German  
2576

2577 The maximum length of 'naturalLanguage' values used to represent IPP attribute values is 63 octets.

#### 2578 **4.1.9 'mimeMediaType'**

2579 The 'mimeMediaType' attribute syntax is the Internet Media Type (sometimes called MIME type) as  
2580 defined by RFC 2046 [RFC2046] and registered according to the procedures of RFC 2048 [RFC2048] for  
2581 identifying a document format. The value MAY include a charset parameter, depending on the  
2582 specification of the Media Type in the IANA Registry [IANA-MT]. Although most other IPP syntax types  
2583 allow for only lower-cased values, this syntax type allows for mixed-case values which are case-insensitive.

2584 Examples are:

2585 'text/html': An HTML document  
2586 'text/plain': A plain text document in US-ASCII (RFC 2046 indicates that in the absence of the charset  
2587 parameter MUST mean US-ASCII rather than simply unspecified) [RFC2046].  
2588 'text/plain; charset=US-ASCII': A plain text document in US-ASCII [52, 56].  
2589 'text/plain; charset=ISO-8859-1': A plain text document in ISO 8859-1 (Latin 1) [ISO8859-1].  
2590 'text/plain; charset=utf-8': A plain text document in ISO 10646 represented as UTF-8 [RFC2279]  
2591 'application/postscript': A PostScript document [RFC2046]  
2592 'application/vnd.hp-PCL': A PCL document [IANA-MT] (charset escape sequence embedded in the  
2593 document data)  
2594 'application/pdf': Portable Document Format - see IANA MIME Media Type registry  
2595 'application/octet-stream': Auto-sense - see section 4.1.9.1  
2596

##### 2597 **4.1.9.1 Application/octet-stream -- Auto-Sensing the document format**

2598 One special type is 'application/octet-stream'. If the Printer object supports this value, the Printer object  
2599 MUST be capable of auto-sensing the format of the document data, either as part of the create operation  
2600 and/or at document processing time. During auto-sensing, a Printer may determine that the document-data  
2601 has a format that the Printer doesn't recognize. If the Printer determines this problem before returning an  
2602 operation response, it rejects the request and returns the 'client-error-document-format-not-supported' status  
2603 code. If the Printer determines this problem after accepting the request and returning an operation response  
2604 with one of the successful status codes, the Printer adds the 'unsupported-document-format' value to the  
2605 job's "job-state-reasons" attribute.

2606 If the Printer object's default value attribute "document-format-default" is set to 'application/octet-stream',  
2607 the Printer object not only supports auto-sensing of the document format, but will depend on the result of  
2608 applying its auto-sensing when the client does not supply the "document-format" attribute. If the client  
2609 supplies a document format value, the Printer MUST rely on the supplied attribute, rather than trust its  
2610 auto-sensing algorithm. To summarize:

- 2611 1. If the client does not supply a document format value, the Printer MUST rely on its default value  
2612 setting (which may be 'application/octet-stream' indicating an auto-sensing mechanism).
- 2613 2. If the client supplies a value other than 'application/octet-stream', the client is supplying valid  
2614 information about the format of the document data and the Printer object MUST trust the client  
2615 supplied value more than the outcome of applying an automatic format detection mechanism. For  
2616 example, the client may be requesting the printing of a PostScript file as a 'text/plain' document.  
2617 The Printer object MUST print a text representation of the PostScript commands rather than  
2618 interpret the stream of PostScript commands and print the result.
- 2619 3. If the client supplies a value of 'application/octet-stream', the client is indicating that the Printer  
2620 object MUST use its auto-sensing mechanism on the client supplied document data whether auto-  
2621 sensing is the Printer object's default or not.  
2622

2623 Note: Since the auto-sensing algorithm is probabilistic, if the client requests both auto-sensing ("document-  
2624 format" set to 'application/octet-stream') and true fidelity ("ipp-attribute-fidelity" set to 'true'), the Printer  
2625 object might not be able to guarantee exactly what the end user intended (the auto-sensing algorithm might  
2626 mistake one document format for another), but it is able to guarantee that its auto-sensing mechanism be  
2627 used.

2628 The maximum length of a 'mimeType' value to represent IPP attribute values is 255 octets.

#### 2629 **4.1.10 'octetString'**

2630 The 'octetString' attribute syntax is a sequence of octets encoded in a maximum of 1023 octets which is  
2631 indicated in sub-section headers using the notation: octetString(MAX). This syntax type is used for opaque  
2632 data.

#### 2633 **4.1.11 'boolean'**

2634 The 'boolean' attribute syntax has only two values: 'true' and 'false'.

#### 2635 **4.1.12 'integer'**

2636 The 'integer' attribute syntax is an integer value that is in the range from  $-2^{31}$  (MIN) to  $2^{31} - 1$  (MAX).  
2637 Each individual attribute may specify the range constraint explicitly in sub-section headers if the range is  
2638 different from the full range of possible integer values. For example: job-priority (integer(1:100)) for the  
2639 "job-priority" attribute. However, the enforcement of that additional constraint is up to the IPP objects, not  
2640 the protocol.

#### 2641 **4.1.13 'rangeOfInteger'**

2642 The 'rangeOfInteger' attribute syntax is an ordered pair of integers that defines an inclusive range of integer  
2643 values. The first integer specifies the lower bound and the second specifies the upper bound. If a range  
2644 constraint is specified in the header description for an attribute in this document whose attribute syntax is

2645 'rangeOfInteger' (i.e., 'X:Y' indicating X as a minimum value and Y as a maximum value), then the  
2646 constraint applies to both integers.

#### 2647 **4.1.14 'dateTime'**

2648 The 'dateTime' attribute syntax is a standard, fixed length, 11 octet representation of the "DateAndTime"  
2649 syntax as defined in RFC 2579 [RFC2579]. RFC 2579 also identifies an 8 octet representation of a  
2650 "DateAndTime" value, but IPP objects MUST use the 11 octet representation. A user interface will provide  
2651 a mapping between protocol dateTime values and displayable user-friendly words or presentation values  
2652 and phrases which are localized to the natural language and date format of the user, including time zone.

#### 2653 **4.1.15 'resolution'**

2654 The 'resolution' attribute syntax specifies a two-dimensional resolution in the indicated units. It consists of  
2655 3 values: a cross feed direction resolution (positive integer value), a feed direction resolution (positive  
2656 integer value), and a units value. The semantics of these three components are taken from the Printer MIB  
2657 [RFC1759] suggested values. That is, the cross feed direction component resolution component is the same  
2658 as the prtMarkerAddressabilityXFeedDir object in the Printer MIB, the feed direction component resolution  
2659 component is the same as the prtMarkerAddressabilityFeedDir in the Printer MIB, and the units component  
2660 is the same as the prtMarkerAddressabilityUnit object in the Printer MIB (namely, '3' indicates dots per inch  
2661 and '4' indicates dots per centimeter). All three values MUST be present even if the first two values are the  
2662 same. Example: '300', '600', '3' indicates a 300 dpi cross-feed direction resolution, a 600 dpi feed direction  
2663 resolution, since a '3' indicates dots per inch (dpi).

#### 2664 **4.1.16 '1setOf X'**

2665 The '1setOf X' attribute syntax is 1 or more values of attribute syntax type X. This syntax type is used for  
2666 multi-valued attributes. The syntax type is called '1setOf' rather than just 'setOf' as a reminder that the set  
2667 of values MUST NOT be empty (i.e., a set of size 0). Sets are normally unordered. However each attribute  
2668 description of this type may specify that the values MUST be in a certain order for that attribute.

## 2669 **4.2 Job Template Attributes**

2670 Job Template attributes describe job processing behavior. Support for Job Template attributes by a Printer  
2671 object is OPTIONAL (see section 12.2.3 for a description of support for OPTIONAL attributes). Also,  
2672 clients OPTIONALLY supply Job Template attributes in create requests.

2673 Job Template attributes conform to the following rules. For each Job Template attribute called "xxx":

- 2674 1. If the Printer object supports "xxx" then it MUST support both a "xxx-default" attribute (unless there  
2675 is a "No" in the table below) and a "xxx-supported" attribute. If the Printer object doesn't support  
2676 "xxx", then it MUST support neither an "xxx-default" attribute nor an "xxx-supported" attribute,  
2677 and it MUST treat an attribute "xxx" supplied by a client as unsupported. An attribute "xxx" may be  
2678 supported for some document formats and not supported for other document formats. For example,

2679 it is expected that a Printer object would only support "orientation-requested" for some document  
2680 formats (such as 'text/plain' or 'text/html') but not others (such as 'application/postscript').  
2681

- 2682 2. "xxx" is OPTIONALLY supplied by the client in a create request. If "xxx" is supplied, the client is  
2683 indicating a desired job processing behavior for this Job. When "xxx" is not supplied, the client is  
2684 indicating that the Printer object apply its default job processing behavior at job processing time if  
2685 the document content does not contain an embedded instruction indicating an xxx-related behavior.  
2686

2687 Since an administrator MAY change the default value attribute after a Job object has been submitted  
2688 but before it has been processed, the default value used by the Printer object at job processing time  
2689 may be different that the default value in effect at job submission time.  
2690

- 2691 3. The "xxx-supported" attribute is a Printer object attribute that describes which job processing  
2692 behaviors are supported by that Printer object. A client can query the Printer object to find out what  
2693 xxx-related behaviors are supported by inspecting the returned values of the "xxx-supported"  
2694 attribute.  
2695

2696 Note: The "xxx" in each "xxx-supported" attribute name is singular, even though an "xxx-  
2697 supported" attribute usually has more than one value, such as "job-sheet-supported", unless the  
2698 "xxx" Job Template attribute is plural, such as "finishings" or "sides". In such cases the "xxx-  
2699 supported" attribute names are: "finishings-supported" and "sides-supported".  
2700

- 2701 4. The "xxx-default" default value attribute describes what will be done at job processing time when no  
2702 other job processing information is supplied by the client (either explicitly as an IPP attribute in the  
2703 create request or implicitly as an embedded instruction within the document data).  
2704

2705 If an application wishes to present an end user with a list of supported values from which to choose, the  
2706 application SHOULD query the Printer object for its supported value attributes. The application SHOULD  
2707 also query the default value attributes. If the application then limits selectable values to only those value  
2708 that are supported, the application can guarantee that the values supplied by the client in the create request  
2709 all fall within the set of supported values at the Printer. When querying the Printer, the client MAY  
2710 enumerate each attribute by name in the Get-Printer-Attributes Request, or the client MAY just name the  
2711 "job-template" group in order to get the complete set of supported attributes (both supported and default  
2712 attributes).

2713 The "finishings" attribute is an example of a Job Template attribute. It can take on a set of values such as  
2714 'staple', 'punch', and/or 'cover'. A client can query the Printer object for the "finishings-supported" attribute  
2715 and the "finishings-default" attribute. The supported attribute contains a set of supported values. The  
2716 default value attribute contains the finishing value(s) that will be used for a new Job if the client does not  
2717 supply a "finishings" attribute in the create request and the document data does not contain any  
2718 corresponding finishing instructions. If the client does supply the "finishings" attribute in the create  
2719 request, the IPP object validates the value or values to make sure that they are a subset of the supported  
2720 values identified in the Printer object's "finishings-supported" attribute. See section 3.1.7.

2721 The table below summarizes the names and relationships for all Job Template attributes. The first column  
2722 of the table (labeled "Job Attribute") shows the name and syntax for each Job Template attribute in the Job  
2723 object. These are the attributes that can optionally be supplied by the client in a create request. The last  
2724 two columns (labeled "Printer: Default Value Attribute" and "Printer: Supported Values Attribute") shows  
2725 the name and syntax for each Job Template attribute in the Printer object (the default value attribute and the  
2726 supported values attribute). A "No" in the table means the Printer MUST NOT support the attribute (that is,  
2727 the attribute is simply not applicable). For brevity in the table, the 'text' and 'name' entries do not show the  
2728 maximum length for each attribute.

2729	+=====+		
2730	Job Attribute	Printer: Default Value Attribute	Printer: Supported Values Attribute
2731	+=====+		
2732	job-priority	job-priority-default	job-priority-supported
2733	(integer 1:100)	(integer 1:100)	(integer 1:100)
2734	+-----+		
2735	job-hold-until	job-hold-until-	job-hold-until-
2736	(type3 keyword	default	supported
2737	name)	(type3 keyword	(1setOf (
2738		name)	type3 keyword   name))
2739	+-----+		
2740	job-sheets	job-sheets-default	job-sheets-supported
2741	(type3 keyword	(type3 keyword	(1setOf (
2742	name)	name)	type3 keyword   name))
2743	+-----+		
2744	multiple-document-	multiple-document-	multiple-document-
2745	handling	handling-default	handling-supported
2746	(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2747	+-----+		
2748	copies	copies-default	copies-supported
2749	(integer (1:MAX))	(integer (1:MAX))	(rangeOfInteger
2750			(1:MAX))
2751	+-----+		
2752	finishings	finishings-default	finishings-supported
2753	(1setOf type2 enum)	(1setOf type2 enum)	(1setOf type2 enum)
2754	+-----+		
2755	page-ranges	No	page-ranges-
2756	(1setOf		supported (boolean)
2757	rangeOfInteger		
2758	(1:MAX))		
2759	+-----+		
2760	sides	sides-default	sides-supported
2761	(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2762	+-----+		
2763	number-up	number-up-default	number-up-supported
2764	(integer (1:MAX))	(integer (1:MAX))	(1setOf ( <u>integer</u>
2765			(1:MAX)
2766			rangeOfInteger
2767			(1:MAX)) <u>)</u>
2768	+-----+		
2769	orientation-	orientation-requested-	orientation-requested-
2770	requested	default	supported
2771	(type2 enum)	(type2 enum)	(1setOf type2 enum)
2772	+-----+		
2773	media	media-default	media-supported
2774	(type3 keyword	(type3 keyword	(1setOf (
2775	name)	name)	type3 keyword   name))
2776	+-----+		
2777	+=====+		

2778			media-ready
2779			(1setOf (
2780			type3 keyword   name))
2781	+-----+-----+-----+		
2782	printer-resolution	printer-resolution-	printer-resolution-
2783	(resolution)	default	supported
2784		(resolution)	(1setOf resolution)
2785	+-----+-----+-----+		
2786	print-quality	print-quality-default	print-quality-
2787	(type2 enum)	(type2 enum)	supported
2788			(1setOf type2 enum)
2789	+-----+-----+-----+		
2790			
2791			

#### 2792 4.2.1 job-priority (integer(1:100))

2793 This attribute specifies a priority for scheduling the Job. A higher value specifies a higher priority. The  
 2794 value 1 indicates the lowest possible priority. The value 100 indicates the highest possible priority. Among  
 2795 those jobs that are ready to print, a Printer MUST print all jobs with a priority value of n before printing  
 2796 those with a priority value of n-1 for all n.

2797 If the Printer object supports this attribute, it MUST always support the full range from 1 to 100. No  
 2798 administrative restrictions are permitted. This way an end-user can always make full use of the entire range  
 2799 with any Printer object. If privileged jobs are implemented outside IPP/1.1, they MUST have priorities  
 2800 higher than 100, rather than restricting the range available to end-users.

2801 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
 2802 object MUST use the value of the Printer object's "job-priority-default" at job submission time (unlike most  
 2803 Job Template attributes that are used if necessary at job processing time).

2804 The syntax for the "job-priority-supported" is also integer(1:100). This single integer value indicates the  
 2805 number of priority levels supported. The Printer object MUST take the value supplied by the client and  
 2806 map it to the closest integer in a sequence of n integers values that are evenly distributed over the range  
 2807 from 1 to 100 using the formula:

2808 
$$\text{roundToNearestInt}((100x+50)/n)$$

2809 where n is the value of "job-priority-supported" and x ranges from 0 through n-1.

2810 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  
 2811 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,  
 2812 and 95; if n = 100, the sequence of values is: 1, 2, 3, ... 100.

2813 If the value of the Printer object's "job-priority-supported" is 10 and the client supplies values in the range 1  
 2814 to 10, the Printer object maps them to 5, in the range 11 to 20, the Printer object maps them to 15, etc.



**2815 4.2.2 job-hold-until (type3 keyword | name (MAX))**

2816 This attribute specifies the named time period during which the Job **MUST** become a candidate for printing.

2817 Standard keyword values for named time periods are:

2818 'no-hold': immediately, if there are not other reasons to hold the job

2819 'indefinite': - the job is held indefinitely, until a client performs a Release-Job (section 3.3.6)

2820 'day-time': during the day

2821 'evening': evening

2822 'night': night

2823 'weekend': weekend

2824 'second-shift': second-shift (after close of business)

2825 'third-shift': third-shift (after midnight)

2826

2827 An administrator **MUST** associate allowable print times with a named time period (by means outside the  
2828 scope of this IPP/1.1 document). An administrator is encouraged to pick names that suggest the type of  
2829 time period. An administrator **MAY** define additional values using the 'name' or 'keyword' attribute syntax,  
2830 depending on implementation.

2831 If the value of this attribute specifies a time period that is in the future, the Printer **SHOULD** add the 'job-  
2832 hold-until-specified' value to the job's "job-state-reasons" attribute, **MUST** move the job to the 'pending-  
2833 held' state, and **MUST NOT** schedule the job for printing until the specified time-period arrives.

2834 When the specified time period arrives, the Printer **MUST** remove the 'job-hold-until-specified' value from  
2835 the job's "job-state-reason" attribute, if present. If there are no other job state reasons that keep the job in  
2836 the 'pending-held' state, the Printer **MUST** consider the job as a candidate for processing by moving the job  
2837 to the 'pending' state.

2838 If this job attribute value is the named value 'no-hold', or the specified time period has already started, the  
2839 job **MUST** be a candidate for processing immediately.

2840 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
2841 object **MUST** use the value of the Printer object's "job-hold-until-default" at job submission time (unlike  
2842 most Job Template attributes that are used if necessary at job processing time).

**2843 4.2.3 job-sheets (type3 keyword | name(MAX))**

2844 This attribute determines which job start/end sheet(s), if any, **MUST** be printed with a job.

2845 Standard keyword values are:

2846 'none': no job sheet is printed

2847 'standard': one or more site specific standard job sheets are printed, e.g. a single start sheet or both start  
2848 and end sheet is printed

2849

2850 An administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending  
2851 on implementation.

2852 The effect of this attribute on jobs with multiple documents MAY be affected by the "multiple-document-  
2853 handling" job attribute (section 4.2.4), depending on the job sheet semantics.

#### 2854 **4.2.4 multiple-document-handling (type2 keyword)**

2855 This attribute is relevant only if a job consists of two or more documents. This attribute MUST be  
2856 supported if the Printer supports multiple documents per job (see sections 3.2.4 and 3.3.1). The attribute  
2857 controls finishing operations and the placement of one or more print-stream pages into impressions and  
2858 onto media sheets. When the value of the "copies" attribute exceeds 1, it also controls the order in which  
2859 the copies that result from processing the documents are produced. For the purposes of this explanations, if  
2860 "a" represents an instance of document data, then the result of processing the data in document "a" is a  
2861 sequence of media sheets represented by "a(\*)".

2862 Standard keyword values are:

2863 'single-document': If a Job object has multiple documents, say, the document data is called a and b, then  
2864 the result of processing all the document data (a and then b) MUST be treated as a single sequence  
2865 of media sheets for finishing operations; that is, finishing would be performed on the concatenation  
2866 of the sequences a(\*),b(\*). The Printer object MUST NOT force the data in each document instance  
2867 to be formatted onto a new print-stream page, nor to start a new impression on a new media sheet. If  
2868 more than one copy is made, the ordering of the sets of media sheets resulting from processing the  
2869 document data MUST be a(\*), b(\*), a(\*), b(\*), ..., and the Printer object MUST force each copy  
2870 (a(\*),b(\*)) to start on a new media sheet.

2871 'separate-documents-uncollated-copies': If a Job object has multiple documents, say, the document data  
2872 is called a and b, then the result of processing the data in each document instance MUST be treated  
2873 as a single sequence of media sheets for finishing operations; that is, the sets a(\*) and b(\*) would  
2874 each be finished separately. The Printer object MUST force each copy of the result of processing the  
2875 data in a single document to start on a new media sheet. If more than one copy is made, the ordering  
2876 of the sets of media sheets resulting from processing the document data MUST be a(\*), a(\*), ...,  
2877 b(\*), b(\*) ... .

2878 'separate-documents-collated-copies': If a Job object has multiple documents, say, the document data is  
2879 called a and b, then the result of processing the data in each document instance MUST be treated as  
2880 a single sequence of media sheets for finishing operations; that is, the sets a(\*) and b(\*) would each  
2881 be finished separately. The Printer object MUST force each copy of the result of processing the data  
2882 in a single document to start on a new media sheet. If more than one copy is made, the ordering of  
2883 the sets of media sheets resulting from processing the document data MUST be a(\*), b(\*), a(\*), b(\*),  
2884 ... .

2885 'single-document-new-sheet': Same as 'single-document', except that the Printer object MUST ensure  
2886 that the first impression of each document instance in the job is placed on a new media sheet. This  
2887 value allows multiple documents to be stapled together with a single staple where each document  
2888 starts on a new sheet.

2889

2890 The 'single-document' value is the same as 'separate-documents-collated-copies' with respect to ordering of  
 2891 print-stream pages, but not media sheet generation, since 'single-document' will put the first page of the  
 2892 next document on the back side of a sheet if an odd number of pages have been produced so far for the job,  
 2893 while 'separate-documents-collated-copies' always forces the next document or document copy on to a new  
 2894 sheet. In addition, if the "finishings" attribute specifies 'staple', then with 'single-document', documents a  
 2895 and b are stapled together as a single document with no regard to new sheets, with 'single-document-new-  
 2896 sheet', documents a and b are stapled together as a single document, but document b starts on a new sheet,  
 2897 but with 'separate-documents-uncollated-copies' and 'separate-documents-collated-copies', documents a and  
 2898 b are stapled separately.

2899 Note: None of these values provide means to produce uncollated sheets within a document, i.e., where  
 2900 multiple copies of sheet n are produced before sheet n+1 of the same document.

2901 The relationship of this attribute and the other attributes that control document processing is described in  
 2902 section 15.3.

#### 2903 **4.2.5 copies (integer(1:MAX))**

2904 This attribute specifies the number of copies to be printed.

2905 On many devices the supported number of collated copies will be limited by the number of physical output  
 2906 bins on the device, and may be different from the number of uncollated copies which can be supported.

2907 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
 2908 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
 2909 control document processing is described in section 15.3.

#### 2910 **4.2.6 finishings (1setOf type2 enum)**

2911 This attribute identifies the finishing operations that the Printer uses for each copy of each printed  
 2912 document in the Job. For Jobs with multiple documents, the "multiple-document-handling" attribute  
 2913 determines what constitutes a "copy" for purposes of finishing.

2914 Standard enum values are:

2915	Value	Symbolic Name and Description
2916		
2917	'3'	'none': Perform no finishing
2918	'4'	'staple': Bind the document(s) with one or more staples. The exact number and placement of 2919 the staples is site-defined.
2920	'5'	'punch': This value indicates that holes are required in the finished document. The exact 2921 number and placement of the holes is site-defined. The punch specification MAY be 2922 satisfied (in a site- and implementation-specific manner) either by drilling/punching, 2923 or by substituting pre-drilled media.

- 2924 '6' 'cover': This value is specified when it is desired to select a non-printed (or pre-printed)  
2925 cover for the document. This does not supplant the specification of a printed cover  
2926 (on cover stock medium) by the document itself.
- 2927 '7' 'bind': This value indicates that a binding is to be applied to the document; the type and  
2928 placement of the binding is site-defined.
- 2929
- 2930 '8' 'saddle-stitch': Bind the document(s) with one or more staples (wire stitches) along the  
2931 middle fold. The exact number and placement of the staples and the middle fold is  
2932 implementation and/or site-defined.
- 2933 '9' 'edge-stitch': Bind the document(s) with one or more staples (wire stitches) along one edge.  
2934 The exact number and placement of the staples is implementation and/or site-  
2935 defined.
- 2936 '10'-'19' reserved for future generic finishing enum values.

2937 The following values are more specific; they indicate a corner or an edge as if the document were a portrait  
2938 document (see below):

- 2939 '20' 'staple-top-left': Bind the document(s) with one or more staples in the top left corner.
- 2940 '21' 'staple-bottom-left': Bind the document(s) with one or more staples in the bottom left  
2941 corner.
- 2942 '22' 'staple-top-right': Bind the document(s) with one or more staples in the top right corner.
- 2943 '23' 'staple-bottom-right': Bind the document(s) with one or more staples in the bottom right  
2944 corner.
- 2945 '24' 'edge-stitch-left': Bind the document(s) with one or more staples (wire stitches) along the  
2946 left edge. The exact number and placement of the staples is implementation and/or  
2947 site-defined.
- 2948 '25' 'edge-stitch-top': Bind the document(s) with one or more staples (wire stitches) along the  
2949 top edge. The exact number and placement of the staples is implementation and/or  
2950 site-defined.
- 2951 '26' 'edge-stitch-right': Bind the document(s) with one or more staples (wire stitches) along the  
2952 right edge. The exact number and placement of the staples is implementation and/or  
2953 site-defined.
- 2954 '27' 'edge-stitch-bottom': Bind the document(s) with one or more staples (wire stitches) along  
2955 the bottom edge. The exact number and placement of the staples is implementation  
2956 and/or site-defined.
- 2957 '28' 'staple-dual-left': Bind the document(s) with two staples (wire stitches) along the left edge  
2958 assuming a portrait document (see above).
- 2959 '29' 'staple-dual-top': Bind the document(s) with two staples (wire stitches) along the top edge  
2960 assuming a portrait document (see above).
- 2961 '30' 'staple-dual-right': Bind the document(s) with two staples (wire stitches) along the right  
2962 edge assuming a portrait document (see above).
- 2963 '31' 'staple-dual-bottom': Bind the document(s) with two staples (wire stitches) along the bottom  
2964 edge assuming a portrait document (see above).

2965 The 'staple-xxx' values are specified with respect to the document as if the document were a portrait  
2966 document. If the document is actually a landscape or a reverse-landscape document, the client supplies the  
2967 appropriate transformed value. For example, to position a staple in the upper left hand corner of a

2968 landscape document when held for reading, the client supplies the 'staple-bottom-left' value (since  
2969 landscape is defined as a +90 degree rotation from portrait, i.e., anti-clockwise). On the other hand, to  
2970 position a staple in the upper left hand corner of a reverse-landscape document when held for reading, the  
2971 client supplies the 'staple-top-right' value (since reverse-landscape is defined as a -90 degree rotation from  
2972 portrait, i.e., clockwise).

2973 The angle (vertical, horizontal, angled) of each staple with respect to the document depends on the  
2974 implementation which may in turn depend on the value of the attribute.

2975 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
2976 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
2977 control document processing is described in section 15.3.

2978 If the client supplies a value of 'none' along with any other combination of values, it is the same as if only  
2979 that other combination of values had been supplied (that is the 'none' value has no effect).

#### 2980 **4.2.7 page-ranges (1setOf rangeOfInteger (1:MAX))**

2981 This attribute identifies the range(s) of print-stream pages that the Printer object uses for each copy of each  
2982 document which are to be printed. Nothing is printed for any pages identified that do not exist in the  
2983 document(s). Ranges MUST be in ascending order, for example: 1-3, 5-7, 15-19 and MUST NOT overlap,  
2984 so that a non-spooling Printer object can process the job in a single pass. If the ranges are not ascending or  
2985 are overlapping, the IPP object MUST reject the request and return the 'client-error-bad-request' status code.  
2986 The attribute is associated with print-stream pages not application-numbered pages (for example, the page  
2987 numbers found in the headers and or footers for certain word processing applications).

2988 For Jobs with multiple documents, the "multiple-document-handling" attribute determines what constitutes  
2989 a "copy" for purposes of the specified page range(s). When "multiple-document-handling" is 'single-  
2990 document', the Printer object MUST apply each supplied page range once to the concatenation of the print-  
2991 stream pages. For example, if there are 8 documents of 10 pages each, the page-range '41:60' prints the  
2992 pages in the 5th and 6th documents as a single document and none of the pages of the other documents are  
2993 printed. When "multiple-document-handling" is 'separate-documents-uncollated-copies' or 'separate-  
2994 documents-collated-copies', the Printer object MUST apply each supplied page range repeatedly to each  
2995 document copy. For the same job, the page-range '1:3, 10:10' would print the first 3 pages and the 10th  
2996 page of each of the 8 documents in the Job, as 8 separate documents.

2997 In most cases, the exact pages to be printed will be generated by a device driver and this attribute would not  
2998 be required. However, when printing an archived document which has already been formatted, the end user  
2999 may elect to print just a subset of the pages contained in the document. In this case, if page-range = n.m is  
3000 specified, the first page to be printed will be page n. All subsequent pages of the document will be printed  
3001 through and including page m.

3002 "page-ranges-supported" is a boolean value indicating whether or not the printer is capable of supporting  
3003 the printing of page ranges. This capability may differ from one PDL to another. There is no "page-ranges-  
3004 default" attribute. If the "page-ranges" attribute is not supplied by the client, all pages of the document will  
3005 be printed.

3006 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
3007 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
3008 control document processing is described in section 15.3.

#### 3009 **4.2.8 sides (type2 keyword)**

3010 This attribute specifies how print-stream pages are to be imposed upon the sides of an instance of a selected  
3011 medium, i.e., an impression.

3012 The standard keyword values are:

3013 'one-sided': imposes each consecutive print-stream page upon the same side of consecutive media  
3014 sheets.

3015 'two-sided-long-edge': imposes each consecutive pair of print-stream pages upon front and back sides of  
3016 consecutive media sheets, such that the orientation of each pair of print-stream pages on the medium  
3017 would be correct for the reader as if for binding on the long edge. This imposition is sometimes  
3018 called 'duplex' or 'head-to-head'.

3019 'two-sided-short-edge': imposes each consecutive pair of print-stream pages upon front and back sides  
3020 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the  
3021 medium would be correct for the reader as if for binding on the short edge. This imposition is  
3022 sometimes called 'tumble' or 'head-to-toe'.

3023

3024 'two-sided-long-edge', 'two-sided-short-edge', 'tumble', and 'duplex' all work the same for portrait or  
3025 landscape. However 'head-to-toe' is 'tumble' in portrait but 'duplex' in landscape. 'head-to-head' also  
3026 switches between 'duplex' and 'tumble' when using portrait and landscape modes.

3027 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
3028 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
3029 control document processing is described in section 15.3.

#### 3030 **4.2.9 number-up (integer(1:MAX))**

3031 This attribute specifies the number of print-stream pages to impose upon a single side of an instance of a  
3032 selected medium. For example, if the value is:

3033 Value	Description
3034 '1'	the Printer MUST place one print-stream page on a single side of an instance of the selected 3035 medium (MAY add some sort of translation, scaling, or rotation).
3036 '2'	the Printer MUST place two print-stream pages on a single side of an instance of the selected 3037 medium (MAY add some sort of translation, scaling, or rotation).
3038 '4'	the Printer MUST place four print-stream pages on a single side of an instance of the 3039 selected medium (MAY add some sort of translation, scaling, or rotation).

3040

3041 This attribute primarily controls the translation, scaling and rotation of print-stream pages.



3043 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
 3044 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
 3045 control document processing is described in section 15.3.

#### 3046 **4.2.10 orientation-requested (type2 enum)**

3047 This attribute indicates the desired orientation for printed print-stream pages; it does not describe the  
 3048 orientation of the client-supplied print-stream pages.

3049 For some document formats (such as 'application/postscript'), the desired orientation of the print-stream  
 3050 pages is specified within the document data. This information is generated by a device driver prior to the  
 3051 submission of the print job. Other document formats (such as 'text/plain') do not include the notion of  
 3052 desired orientation within the document data. In the latter case it is possible for the Printer object to bind  
 3053 the desired orientation to the document data after it has been submitted. It is expected that a Printer object  
 3054 would only support "orientations-requested" for some document formats (e.g., 'text/plain' or 'text/html') but  
 3055 not others (e.g., 'application/postscript'). This is no different than any other Job Template attribute since  
 3056 section 4.2, item 1, points out that a Printer object may support or not support any Job Template attribute  
 3057 based on the document format supplied by the client. However, a special mention is made here since it is  
 3058 very likely that a Printer object will support "orientation-requested" for only a subset of the supported  
 3059 document formats.

3060 Standard enum values are:

3061	Value	Symbolic Name and Description
3062		
3063	'3'	'portrait': The content will be imaged across the short edge of the medium.
3064	'4'	'landscape': The content will be imaged across the long edge of the medium. Landscape is
3065		defined to be a rotation of the print-stream page to be imaged by +90 degrees with
3066		respect to the medium (i.e. anti-clockwise) from the portrait orientation. Note: The
3067		+90 direction was chosen because simple finishing on the long edge is the same edge
3068		whether portrait or landscape
3069	'5'	'reverse-landscape': The content will be imaged across the long edge of the medium.
3070		Reverse-landscape is defined to be a rotation of the print-stream page to be imaged
3071		by -90 degrees with respect to the medium (i.e. clockwise) from the portrait
3072		orientation. Note: The 'reverse-landscape' value was added because some
3073		applications rotate landscape -90 degrees from portrait, rather than +90 degrees.
3074	'6'	'reverse-portrait': The content will be imaged across the short edge of the medium. Reverse-
3075		portrait is defined to be a rotation of the print-stream page to be imaged by 180
3076		degrees with respect to the medium from the portrait orientation. Note: The 'reverse-
3077		portrait' value was added for use with the "finishings" attribute in cases where the
3078		opposite edge is desired for finishing a portrait document on simple finishing devices
3079		that have only one finishing position. Thus a 'text/plain' portrait document can be
3080		stapled "on the right" by a simple finishing device as is common use with some
3081		middle eastern languages such as Hebrew.
3082		



3083 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
3084 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
3085 control document processing is described in section 15.3.

#### 3086 **4.2.11 media (type3 keyword | name(MAX))**

3087 This attribute identifies the medium that the Printer uses for all impressions of the Job.

3088 The values for "media" include medium-names, medium-sizes, input-trays and electronic forms so that one  
3089 attribute specifies the media. If a Printer object supports a medium name as a value of this attribute, such a  
3090 medium name implicitly selects an input-tray that contains the specified medium. If a Printer object  
3091 supports a medium size as a value of this attribute, such a medium size implicitly selects a medium name  
3092 that in turn implicitly selects an input-tray that contains the medium with the specified size. If a Printer  
3093 object supports an input-tray as the value of this attribute, such an input-tray implicitly selects the medium  
3094 that is in that input-tray at the time the job prints. This case includes manual-feed input-trays. If a Printer  
3095 object supports an electronic form as the value of this attribute, such an electronic form implicitly selects a  
3096 medium-name that in turn implicitly selects an input-tray that contains the medium specified by the  
3097 electronic form. The electronic form also implicitly selects an image that the Printer MUST merge with the  
3098 document data as it prints each page.

3099 Standard keyword values are (taken from ISO DPA [\[ISO10175\]](#), ~~and~~ the Printer MIB [\[RFC1759\]](#), and  
3100 [ASME-Y14.1M \[ASME-Y14.1M\]](#)) and are listed in section 14. An administrator MAY define additional  
3101 values using the 'name' or 'keyword' attribute syntax, depending on implementation.

3102 There is also an additional Printer attribute named "media-ready" which differs from "media-supported" in  
3103 that legal values only include the subset of "media-supported" values that are physically loaded and ready  
3104 for printing with no operator intervention required. If an IPP object supports "media-supported", it NEED  
3105 NOT support "media-ready".

3106 The relationship of this attribute and the other attributes that control document processing is described in  
3107 section 15.3.

#### 3108 **4.2.12 printer-resolution (resolution)**

3109 This attribute identifies the resolution that Printer uses for the Job.

#### 3110 **4.2.13 print-quality (type2 enum)**

3111 This attribute specifies the print quality that the Printer uses for the Job.

3112 The standard enum values are:

3113	Value	Symbolic Name and Description
3114		
3115	'3'	'draft': lowest quality available on the printer
3116	'4'	'normal': normal or intermediate quality on the printer

3117 '5' 'high': highest quality available on the printer  
3118

### 3119 **4.3 Job Description Attributes**

3120 The attributes in this section form the attribute group called "job-description". The following table  
3121 summarizes these attributes. The third column indicates whether the attribute is a REQUIRED attribute  
3122 that MUST be supported by Printer objects. If it is not indicated as REQUIRED, then it is OPTIONAL.  
3123 The maximum size in octets for 'text' and 'name' attributes is indicated in parentheses.

	Attribute	Syntax	REQUIRED?
3124			
3125			
3126			
3127	job-uri	uri	REQUIRED
3128			
3129	job-id	integer(1:MAX)	REQUIRED
3130			
3131	job-printer-uri	uri	REQUIRED
3132			
3133	job-more-info	uri	
3134			
3135	job-name	name (MAX)	REQUIRED
3136			
3137	job-originating-user-name	name (MAX)	REQUIRED
3138			
3139	job-state	type1 enum	REQUIRED
3140			
3141	job-state-reasons	1setOf type2 keyword	REQUIRED
3142			
3143	job-state-message	text (MAX)	
3144			
3145	job-detailed-status-	1setOf text (MAX)	
3146	messages		
3147			
3148	job-document-access-errors	1setOf text (MAX)	
3149			
3150	number-of-documents	integer (0:MAX)	
3151			
3152	output-device-assigned	name (127)	
3153			
3154	time-at-creation	integer (MIN:MAX)	REQUIRED
3155			
3156	time-at-processing	integer (MIN:MAX)	REQUIRED
3157			
3158	time-at-completed	integer (MIN:MAX)	REQUIRED
3159			
3160	job-printer-up-time	integer (1:MAX)	REQUIRED
3161			
3162	date-time-at-creation	dateTime	OPTIONAL
3163			
3164			
3165	date-time-at-processing	dateTime	OPTIONAL
3166			
3167			
3168	date-time-at-completed	dateTime	OPTIONAL
3169			
3170			
3171	number-of-intervening-jobs	integer (0:MAX)	
3172			

3173	job-message-from-operator	text (127)		
3174	+-----+	+-----+	+-----+	+-----+
3175	job-k-octets	integer (0:MAX)		
3176	+-----+	+-----+	+-----+	+-----+
3177	job-impressions	integer (0:MAX)		
3178	+-----+	+-----+	+-----+	+-----+
3179	job-media-sheets	integer (0:MAX)		
3180	+-----+	+-----+	+-----+	+-----+
3181	job-k-octets-processed	integer (0:MAX)		
3182	+-----+	+-----+	+-----+	+-----+
3183	job-impressions-completed	integer (0:MAX)		
3184	+-----+	+-----+	+-----+	+-----+
3185	job-media-sheets-completed	integer (0:MAX)		
3186	+-----+	+-----+	+-----+	+-----+
3187	attributes-charset	charset		REQUIRED
3188	+-----+	+-----+	+-----+	+-----+
3189	attributes-natural-language	naturalLanguage		REQUIRED
3190	+-----+	+-----+	+-----+	+-----+

3191  
3192

### 3193 4.3.1 job-uri (uri)

3194 This REQUIRED attribute contains the URI for the job. The Printer object, on receipt of a new job,  
3195 generates a URI which identifies the new Job. The Printer object returns the value of the "job-uri" attribute  
3196 as part of the response to a create request. The precise format of a Job URI is implementation dependent.  
3197 If the Printer object supports more than one URI and there is some relationship between the newly formed  
3198 Job URI and the Printer object's URI, the Printer object uses the Printer URI supplied by the client in the  
3199 create request. For example, if the create request comes in over a secure channel, the new Job URI MUST  
3200 use the same secure channel. This can be guaranteed because the Printer object is responsible for  
3201 generating the Job URI and the Printer object is aware of its security configuration and policy as well as the  
3202 Printer URI used in the create request.

3203 For a description of this attribute and its relationship to "job-id" and "job-printer-uri" attribute, see the  
3204 discussion in section 2.4 on "Object Identity".

### 3205 4.3.2 job-id (integer(1:MAX))

3206 This REQUIRED attribute contains the ID of the job. The Printer, on receipt of a new job, generates an ID  
3207 which identifies the new Job on that Printer. The Printer returns the value of the "job-id" attribute as part of  
3208 the response to a create request. The 0 value is not included to allow for compatibility with SNMP index  
3209 values which also cannot be 0.

3210 For a description of this attribute and its relationship to "job-uri" and "job-printer-uri" attribute, see the  
3211 discussion in section 2.4 on "Object Identity".

### 3212 4.3.3 job-printer-uri (uri)

3213 This REQUIRED attribute identifies the Printer object that created this Job object. When a Printer object  
3214 creates a Job object, it populates this attribute with the Printer object URI that was used in the create  
3215 request. This attribute permits a client to identify the Printer object that created this Job object when only  
3216 the Job object's URI is available to the client. The client queries the creating Printer object to determine  
3217 which languages, charsets, operations, are supported for this Job.

3218 For a description of this attribute and its relationship to "job-uri" and "job-id" attribute, see the discussion in  
3219 section 2.4 on "Object Identity".

### 3220 4.3.4 job-more-info (uri)

3221 Similar to "printer-more-info", this attribute contains the URI referencing some resource with more  
3222 information about this Job object, perhaps an HTML page containing information about the Job.

### 3223 4.3.5 job-name (name(MAX))

3224 This REQUIRED attribute is the name of the job. It is a name that is more user friendly than the "job-uri"  
3225 attribute value. It does not need to be unique between Jobs. The Job's "job-name" attribute is set to the  
3226 value supplied by the client in the "job-name" operation attribute in the create request (see Section 3.2.1.1).  
3227 If, however, the "job-name" operation attribute is not supplied by the client in the create request, the Printer  
3228 object, on creation of the Job, MUST generate a name. The printer SHOULD generate the value of the  
3229 Job's "job-name" attribute from the first of the following sources that produces a value: 1) the "document-  
3230 name" operation attribute of the first (or only) document, 2) the "document-URI" attribute of the first (or  
3231 only) document, or 3) any other piece of Job specific and/or Document Content information.

### 3232 4.3.6 job-originating-user-name (name(MAX))

3233 This REQUIRED attribute contains the name of the end user that submitted the print job. The Printer  
3234 object sets this attribute to the most authenticated printable name that it can obtain from the authentication  
3235 service over which the IPP operation was received. Only if such is not available, does the Printer object use  
3236 the value supplied by the client in the "requesting-user-name" operation attribute of the create operation  
3237 (see Sections 4.4.2, 4.4.3, and 8).

3238 Note: The Printer object needs to keep an internal originating user id of some form, typically as a credential  
3239 of a principal, with the Job object. Since such an internal attribute is implementation-dependent and not of  
3240 interest to clients, it is not specified as a Job Description attribute. This originating user id is used for  
3241 authorization checks (if any) on all subsequent operations.

### 3242 4.3.7 job-state (type1 enum)

3243 This REQUIRED attribute identifies the current state of the job. Even though the IPP protocol defines  
3244 seven values for job states (plus the out-of-band 'unknown' value - see Section 4.1), implementations only  
3245 need to support those states which are appropriate for the particular implementation. In other words, a

3246 Printer supports only those job states implemented by the output device and available to the Printer object  
3247 implementation.

3248 Standard enum values are:

3249 Values Symbolic Name and Description

3250

3251 '3' 'pending': The job is a candidate to start processing, but is not yet processing.

3252

3253 '4' 'pending-held': The job is not a candidate for processing for any number of reasons but will  
3254 return to the 'pending' state as soon as the reasons are no longer present. The job's  
3255 "job-state-reason" attribute MUST indicate why the job is no longer a candidate for  
3256 processing.

3257

3258 '5' 'processing': One or more of:

3259

3260 1. the job is using, or is attempting to use, one or more purely software processes  
3261 that are analyzing, creating, or interpreting a PDL, etc.,

3262 2. the job is using, or is attempting to use, one or more hardware devices that are  
3263 interpreting a PDL, making marks on a medium, and/or performing finishing, such as  
3264 stapling, etc.,

3265 3. the Printer object has made the job ready for printing, but the output device is not  
3266 yet printing it, either because the job hasn't reached the output device or because the  
3267 job is queued in the output device or some other spooler, awaiting the output device  
3268 to print it.

3269

3270 When the job is in the 'processing' state, the entire job state includes the detailed  
3271 status represented in the Printer object's "printer-state", "printer-state-reasons", and  
3272 "printer-state-message" attributes.

3273 Implementations MAY, though they NEED NOT, include additional values in the  
3274 job's "job-state-reasons" attribute to indicate the progress of the job, such as adding  
3275 the 'job-printing' value to indicate when the output device is actually making marks  
3276 on paper and/or the 'processing-to-stop-point' value to indicate that the IPP object is  
3277 in the process of canceling or aborting the job. Most implementations won't bother  
3278 with this nuance.

3279

3280 '6' 'processing-stopped': The job has stopped while processing for any number of reasons and  
3281 will return to the 'processing' state as soon as the reasons are no longer present.

3282

3283 The job's "job-state-reason" attribute MAY indicate why the job has stopped  
3284 processing. For example, if the output device is stopped, the 'printer-stopped' value  
3285 MAY be included in the job's "job-state-reasons" attribute.

3286

3287 Note: When an output device is stopped, the device usually indicates its condition in  
3288 human readable form locally at the device. A client can obtain more complete device  
3289 status remotely by querying the Printer object's "printer-state", "printer-state-reasons"  
3290 and "printer-state-message" attributes.

3291

3292 '7' 'canceled': The job has been canceled by a Cancel-Job operation and the Printer object has  
3293 completed canceling the job and all job status attributes have reached their final  
3294 values for the job. While the Printer object is canceling the job, the job remains in its  
3295 current state, but the job's "job-state-reasons" attribute SHOULD contain the  
3296 'processing-to-stop-point' value and one of the 'canceled-by-user', 'canceled-by-  
3297 operator', or 'canceled-at-device' value. When the job moves to the 'canceled' state,  
3298 the 'processing-to-stop-point' value, if present, MUST be removed, but the 'canceled-  
3299 by-xxx', if present, MUST remain.

3300

3301 '8' 'aborted': The job has been aborted by the system, usually while the job was in the  
3302 'processing' or 'processing-stopped' state and the Printer has completed aborting the  
3303 job and all job status attributes have reached their final values for the job. While the  
3304 Printer object is aborting the job, the job remains in its current state, but the job's  
3305 "job-state-reasons" attribute SHOULD contain the 'processing-to-stop-point' and  
3306 'aborted-by-system' values. When the job moves to the 'aborted' state, the  
3307 'processing-to-stop-point' value, if present, MUST be removed, but the 'aborted-by-  
3308 system' value, if present, MUST remain.

3309

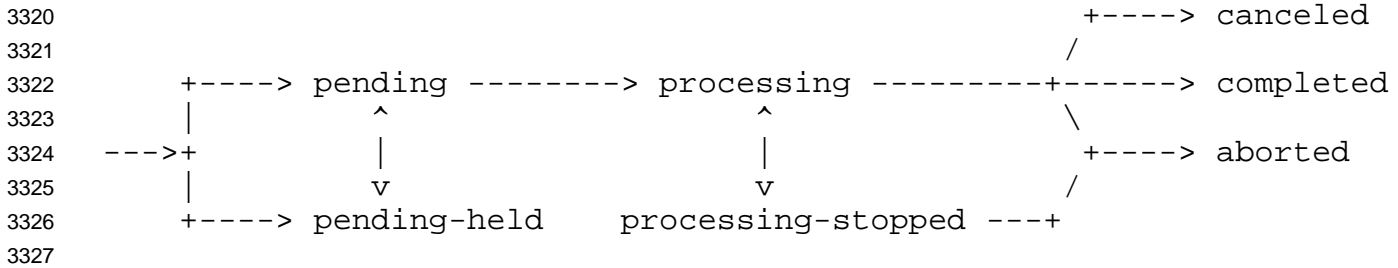
3310 '9' 'completed': The job has completed successfully or with warnings or errors after processing  
3311 and all of the job media sheets have been successfully stacked in the appropriate  
3312 output bin(s) and all job status attributes have reached their final values for the job.  
3313 The job's "job-state-reasons" attribute SHOULD contain one of: 'completed-  
3314 successfully', 'completed-with-warnings', or 'completed-with-errors' values.

3315

3316 The final value for this attribute MUST be one of: 'completed', 'canceled', or 'aborted' before the Printer  
3317 removes the job altogether. The length of time that jobs remain in the 'canceled', 'aborted', and 'completed'  
3318 states depends on implementation. See section 4.3.7.2.

3319 The following figure shows the normal job state transitions.





3328 Normally a job progresses from left to right. Other state transitions are unlikely, but are not forbidden. Not  
 3329 shown are the transitions to the 'canceled' state from the 'pending', 'pending-held', and 'processing-stopped'  
 3330 states.

3331 Jobs reach one of the three terminal states: 'completed', 'canceled', or 'aborted', after the jobs have  
 3332 completed all activity, including stacking output media, after the jobs have completed all activity, and all  
 3333 job status attributes have reached their final values for the job.

#### 3334 4.3.7.1 Forwarding Servers

3335 As with all other IPP attributes, if the implementation cannot determine the correct value for this attribute,  
 3336 it SHOULD respond with the out-of-band value 'unknown' (see section 4.1) rather than try to guess at some  
 3337 possibly incorrect value and give the end user the wrong impression about the state of the Job object. For  
 3338 example, if the implementation is just a gateway into some printing system from which it can normally get  
 3339 status, but temporarily is unable, then the implementation should return the 'unknown' value. However, if  
 3340 the implementation is a gateway to a printing system that never provides detailed status about the print job,  
 3341 the implementation MAY set the IPP Job object's state to 'completed', provided that it also sets the 'queued-  
 3342 in-device' value in the job's "job-state-reasons" attribute (see section 4.3.8).

#### 3343 4.3.7.2 Partitioning of Job States

3344 This section partitions the 7 job states into phases: Job Not Completed, Job Retention, Job History, and Job  
 3345 Removal. This section also explains the 'job-restartable' value of the "job-state-reasons" Job Description  
 3346 attribute for use with the Restart-Job operation.

3347 Job Not Completed: When a job is in the 'pending', 'pending-held', 'processing', or 'processing-stopped'  
 3348 states, the job is not completed.

3349 Job Retention: When a job enters one of the three terminal job states: 'completed', 'canceled', or 'aborted',  
 3350 the IPP Printer object MAY "retain" the job in a restartable condition for an implementation-defined time  
 3351 period. This time period MAY be zero seconds and MAY depend on the terminal job state. This phase is  
 3352 called Job Retention. While in the Job Retention phase, the job's document data is retained and a client  
 3353 may restart the job using the Restart-Job operation. If the IPP object supports the Restart-Job operation,  
 3354 then it SHOULD indicate that the job is restartable by adding the 'job-restartable' value to the job's "job-  
 3355 state-reasons" attribute (see Section 4.3.8) during the Job Retention phase.

3356 Job History: After the Job Retention phase expires for a job, the Printer object deletes the document data  
 3357 for the job and the job becomes part of the Job History. The Printer object MAY also delete any number of

3358 the job attributes. Since the job is no longer restartable, the Printer object MUST remove the 'job-  
3359 restartable' value from the job's "job-state-reasons" attribute, if present.

3360 Job Removal: After the job has remained in the Job History for an implementation-defined time, such as  
3361 when the number of jobs exceeds a fixed number or after a fixed time period (which MAY be zero  
3362 seconds), the IPP Printer removes the job from the system.

3363 Using the Get-Jobs operation and supplying the 'not-completed' value for the "which-jobs" operation  
3364 attribute, a client is requesting jobs in the Job Not Completed phase. Using the Get-Jobs operation and  
3365 supplying the 'completed' value for the "which-jobs" operation attribute, a client is requesting jobs in the  
3366 Job Retention and Job History phases. Using the Get-Job-Attributes operation, a client is requesting a job  
3367 in any phase except Job Removal. After Job Removal, the Get-Job-Attributes and Get-Jobs operations no  
3368 longer are capable of returning any information about a job.

#### 3369 **4.3.8 job-state-reasons (1setOf type2 keyword)**

3370 This REQUIRED attribute provides additional information about the job's current state, i.e., information  
3371 that augments the value of the job's "job-state" attribute.

3372 These values MAY be used with any job state or states for which the reason makes sense. Some of these  
3373 value definitions indicate conformance requirements; the rest are OPTIONAL. Furthermore, when  
3374 implemented, the Printer MUST return these values when the reason applies and MUST NOT return them  
3375 when the reason no longer applies whether the value of the Job's "job-state" attribute changed or not. When  
3376 the Job does not have any reasons for being in its current state, the value of the Job's "job-state-reasons"  
3377 attribute MUST be 'none'.

3378 Note: While values cannot be added to the 'job-state' attribute without impacting deployed clients that take  
3379 actions upon receiving "job-state" values, it is the intent that additional "job-state-reasons" values can be  
3380 defined and registered without impacting such deployed clients. In other words, the "job-state-reasons"  
3381 attribute is intended to be extensible.

3382 The following standard keyword values are defined. For ease of understanding, the values are presented in  
3383 the order in which the reasons are likely to occur (if implemented), starting with the 'job-incoming' value:

3384 'none': There are no reasons for the job's current state. This state reason is semantically equivalent to  
3385 "job-state-reasons" without any value and MUST be used when there is no other value, since the  
3386 1setOf attribute syntax requires at least one value.

3387 'job-incoming': The Create-Job operation has been accepted by the Printer, but the Printer is expecting  
3388 additional Send-Document and/or Send-URI operations and/or is accessing/accepting document  
3389 data.

3390 'job-data-insufficient': The Create-Job operation has been accepted by the Printer, but the Printer is  
3391 expecting additional document data before it can move the job into the 'processing' state. If a Printer  
3392 starts processing before it has received all data, the Printer removes the 'job-data-insufficient'  
3393 reason, but the 'job-incoming' remains. If a Printer starts processing after it has received all data, the  
3394 Printer removes the 'job-data-insufficient' reason and the 'job-incoming' at the same time.

3395 'document-access-error': After accepting a Print-URI or Send-URI request, the Printer could not access  
3396 one or more documents passed by reference. This reason is intended to cover any file access  
3397 problem, including file does not exist and access denied because of an access control problem. The  
3398 Printer MAY also indicate the document access error using the "job-document-access-errors" Job  
3399 Description attribute (see section 4.3.11). Whether the Printer aborts the job and moves the job to  
3400 the 'aborted' job state or prints all documents that are accessible and moves the job to the 'completed'  
3401 job state and adds the 'completed-with-errors' value in the job's "job-state-reasons" attribute depends  
3402 on implementation and/or site policy. This value SHOULD be supported if the Print-URI or Send-  
3403 URI operations are supported.

3404 'submission-interrupted': The job was not completely submitted for some unforeseen reason, such as:  
3405 (1) the Printer has crashed before the job was closed by the client, (2) the Printer or the document  
3406 transfer method has crashed in some non-recoverable way before the document data was entirely  
3407 transferred to the Printer, (3) the client crashed or failed to close the job before the time-out period.  
3408 See section 4.4.31.

3409 'job-outgoing': The Printer is transmitting the job to the output device.

3410 'job-hold-until-specified': The value of the job's "job-hold-until" attribute was specified with a time  
3411 period that is still in the future. The job MUST NOT be a candidate for processing until this reason  
3412 is removed and there are no other reasons to hold the job. This value SHOULD be supported if the  
3413 "job-hold-until" Job Template attribute is supported.

3414 'resources-are-not-ready': At least one of the resources needed by the job, such as media, fonts, resource  
3415 objects, etc., is not ready on any of the physical printer's for which the job is a candidate. This  
3416 condition MAY be detected when the job is accepted, or subsequently while the job is pending or  
3417 processing, depending on implementation. The job may remain in its current state or be moved to  
3418 the 'pending-held' state, depending on implementation and/or job scheduling policy.

3419 'printer-stopped-partly': The value of the Printer's "printer-state-reasons" attribute contains the value  
3420 'stopped-partly'.

3421 'printer-stopped': The value of the Printer's "printer-state" attribute is 'stopped'.

3422 'job-interpreting': Job is in the 'processing' state, but more specifically, the Printer is interpreting the  
3423 document data.

3424 'job-queued': Job is in the 'processing' state, but more specifically, the Printer has queued the document  
3425 data.

3426 'job-transforming': Job is in the 'processing' state, but more specifically, the Printer is interpreting  
3427 document data and producing another electronic representation.

3428 'job-queued-for-marker': Job is in any of the 'pending-held', 'pending', or 'processing' states, but more  
3429 specifically, the Printer has completed enough processing of the document to be able to start  
3430 marking and the job is waiting for the marker. Systems that require human intervention to release  
3431 jobs using the Release-Job operation, put the job into the 'pending-held' job state. Systems that  
3432 automatically select a job to use the marker put the job into the 'pending' job state or keep the job in  
3433 the 'processing' job state while waiting for the marker, depending on implementation. All  
3434 implementations put the job into (or back into) the 'processing' state when marking does begin.

3435 'job-printing': The output device is marking media. This value is useful for Printers which spend a great  
3436 deal of time processing (1) when no marking is happening and then want to show that marking is  
3437 now happening or (2) when the job is in the process of being canceled or aborted while the job  
3438 remains in the 'processing' state, but the marking has not yet stopped so that impression or sheet  
3439 counts are still increasing for the job.

- 3440 'job-canceled-by-user': The job was canceled by the owner of the job using the Cancel-Job request, i.e.,  
3441 by a user whose authenticated identity is the same as the value of the originating user that created  
3442 the Job object, or by some other authorized end-user, such as a member of the job owner's security  
3443 group. This value SHOULD be supported.
- 3444 'job-canceled-by-operator': The job was canceled by the operator using the Cancel-Job request, i.e., by a  
3445 user who has been authenticated as having operator privileges (whether local or remote). If the  
3446 security policy is to allow anyone to cancel anyone's job, then this value may be used when the job  
3447 is canceled by other than the owner of the job. For such a security policy, in effect, everyone is an  
3448 operator as far as canceling jobs with IPP is concerned. This value SHOULD be supported if the  
3449 implementation permits canceling by other than the owner of the job.
- 3450 'job-canceled-at-device': The job was canceled by an unidentified local user, i.e., a user at a console at  
3451 the device. This value SHOULD be supported if the implementation supports canceling jobs at the  
3452 console.
- 3453 'aborted-by-system': The job (1) is in the process of being aborted, (2) has been aborted by the system  
3454 and placed in the 'aborted' state, or (3) has been aborted by the system and placed in the 'pending-  
3455 held' state, so that a user or operator can manually try the job again. This value SHOULD be  
3456 supported.
- 3457 'unsupported-compression': The job was aborted by the system because the Printer determined while  
3458 attempting to decompress the document-data's that the compression is actually not among those  
3459 supported by the Printer. This value MUST be supported, since "compressions is a REQUIRED  
3460 operation attribute.
- 3461 'compression-error': The job was aborted by the system because the Printer encountered an error in the  
3462 document-data while decompressing it. If the Printer posts this reason, the document-data has  
3463 already passed any tests that would have led to the 'unsupported-compression' job-state-reason.
- 3464 'unsupported-document-format': The job was aborted by the system because the document-data's  
3465 document-format is not among those supported by the Printer. If the client specifies the document-  
3466 format as 'application/octet-stream', the printer MAY abort the job and post this reason even though  
3467 the format is a member of the "document-format-supported" printer attribute, but not among the  
3468 auto-sensed document-formats. This value MUST be supported, since "document-format" is a  
3469 REQUIRED operation attribute.
- 3470 'document-format-error': The job was aborted by the system because the Printer encountered an error in  
3471 the document-data while processing it. If the Printer posts this reason, the document-data has  
3472 already passed any tests that would have led to the 'unsupported-document-format' job-state-reason.
- 3473 'processing-to-stop-point': The requester has issued a Cancel-Job operation or the Printer object has  
3474 aborted the job, but is still performing some actions on the job until a specified stop point occurs or  
3475 job termination/cleanup is completed.
- 3476 If the implementation requires some measurable time to cancel the job in the 'processing' or  
3477 'processing-stopped' job states, the IPP object MUST use this value to indicate that the Printer object  
3478 is still performing some actions on the job while the job remains in the 'processing' or 'processing-  
3479 stopped' state. After all the job's job description attributes have stopped incrementing, the Printer  
3480 object moves the job from the 'processing' state to the 'canceled' or 'aborted' job states.
- 3481 'service-off-line': The Printer is off-line and accepting no jobs. All 'pending' jobs are put into the  
3482 'pending-held' state. This situation could be true if the service's or document transform's input is  
3483 impaired or broken.

3484 'job-completed-successfully': The job completed successfully. This value SHOULD be supported.  
3485 'job-completed-with-warnings': The job completed with warnings. This value SHOULD be supported  
3486 if the implementation detects warnings.  
3487 'job-completed-with-errors': The job completed with errors (and possibly warnings too). This value  
3488 SHOULD be supported if the implementation detects errors.  
3489 'job-restartable' - This job is retained (see section 4.3.7.2) and is currently able to be restarted using the  
3490 Restart-Job operation (see section 3.3.7). If 'job-restartable' is a value of the job's 'job-state-reasons'  
3491 attribute, then the IPP object MUST accept a Restart-Job operation for that job. This value  
3492 SHOULD be supported if the Restart-Job operation is supported.  
3493 'queued-in-device': The job has been forwarded to a device or print system that is unable to send back  
3494 status. The Printer sets the job's "job-state" attribute to 'completed' and adds the 'queued-in-device'  
3495 value to the job's "job-state-reasons" attribute to indicate that the Printer has no additional  
3496 information about the job and never will have any better information. See section 4.3.7.1.

### 3497 **4.3.9 job-state-message (text(MAX))**

3498 This attribute specifies information about the "job-state" and "job-state-reasons" attributes in human  
3499 readable text. If the Printer object supports this attribute, the Printer object MUST be able to generate this  
3500 message in any of the natural languages identified by the Printer's "generated-natural-language-supported"  
3501 attribute (see the "attributes-natural-language" operation attribute specified in Section 3.1.4.1).

3502 The value SHOULD NOT contain additional information not contained in the values of the "job-state" and  
3503 "job-states-reasons" attributes, such as interpreter error information. Otherwise, application programs  
3504 might attempt to parse the (localized text). For such additional information such as interpreter errors for  
3505 application program consumption or specific document access errors, new attributes with keyword values,  
3506 needs to be developed and registered.

### 3507 **4.3.10 job-detailed-status-messages (1setOf text(MAX))**

3508 This attribute specifies additional detailed and technical information about the job. Neither the Printer nor  
3509 the client localizes the message(s), since they are intended for use by the system administrator or other  
3510 experienced technical persons. Clients MUST NOT attempt to parse the value of this attribute. See "job-  
3511 document-access-errors" (section 4.3.11) for additional errors that a program can process.

### 3512 **4.3.11 job-document-access-errors (1setOf text(MAX))**

3513 This attribute provides additional information about each document access error for this job encountered by  
3514 the Printer after it returned a response to the Print-URI or Send-URI operation and subsequently attempted  
3515 to access document(s) supplied in the Print-URI or Send-URI operation. For errors in the protocol that is  
3516 identified by the URI scheme in the "document-uri" operation attribute, such as 'http:' or 'ftp:', the error code  
3517 is returned in parentheses, followed by the URI. For example:

3518 (404) http://ftp.pwg.org/pub/pwg/ipp/new\_MOD/ipp-model-v11-990510.pdf  
3519

3520 Most Internet protocols use decimal error codes (unlike IPP), so the ASCII error code representation is in  
3521 decimal.



3522 **4.3.12 number-of-documents (integer(0:MAX))**

3523 This attribute indicates the number of documents in the job, i.e., the number of Send-Document, Send-URI,  
3524 Print-Job, or Print-URI operations that the Printer has accepted for this job, regardless of whether the  
3525 document data has reached the Printer object or not.

3526 Implementations supporting the OPTIONAL Create-Job/Send-Document/Send-URI operations SHOULD  
3527 support this attribute so that clients can query the number of documents in each job.

3528 **4.3.13 output-device-assigned (name(127))**

3529 This attribute identifies the output device to which the Printer object has assigned this job. If an output  
3530 device implements an embedded Printer object, the Printer object NEED NOT set this attribute. If a print  
3531 server implements a Printer object, the value MAY be empty (zero-length string) or not returned until the  
3532 Printer object assigns an output device to the job. This attribute is particularly useful when a single Printer  
3533 object supports multiple devices (so called "fan-out" - [see section 2.1](#)).

3534 **4.3.14 Event Time Job Description Attributes**

3535 This section defines the Job Description attributes that indicate the time at which certain events occur for a  
3536 job. If the job event has not yet occurred, then the IPP object MUST return the 'no-value' out-of-band value  
3537 (see the beginning of Section 4.1). The "time-at-xxx(integer)" attributes represent time as an 'integer'  
3538 representing the number of seconds since the device was powered up (informally called "time ticks"). The  
3539 "date-time-at-xxx(dateTime)" attributes represent time as 'dateTime' representing date and time (including  
3540 an offset from UTC).

3541 In order to populate these attributes, the Printer object copies the value(s) of the following Printer  
3542 Description attributes at the time the event occurs:

- 3543 1. the value in the Printer's "printer-up-time" attribute for the "time-at-xxx(integer)" attributes  
3544 2. the value in the Printer's "printer-current-time" attribute for the "date-time-at-xxx(dateTime)"  
3545 attributes.

3546 If the Printer resets its "printer-up-time" attribute to 1 on power-up (see section 4.4.29) and has persistent  
3547 jobs, then it MUST change all of jobs' "time-at-xxx(integer)" (time tick) job attributes whose events have  
3548 occurred either to:

- 3549 1. 0 to indicate that the event happened before the most recent power up OR  
3550 2. the negative of the number of seconds before the most recent power-up that the event took place,  
3551 though the negative number NEED NOT reflect the exact number of seconds.

3552 If a client queries a "time-at-xxx(integer)" time tick Job attribute and finds the value to be 0 or negative, the  
3553 client MUST assume that the event occurred in some life other than the Printer's current life.

3554 Note: A Printer does not change the values of any "date-time-at-xxx(dateTime)" job attributes on power-up.

3555 **4.3.14.1 time-at-creation (integer(MIN:MAX))**

3556 This REQUIRED attribute indicates the time at which the Job object was created.

3557 **4.3.14.2 time-at-processing (integer(MIN:MAX))**

3558 This REQUIRED attribute indicates the time at which the Job object first began processing after the create  
3559 operation or the most recent Restart-Job operation. The out-of-band 'no-value' value is returned if the job  
3560 has not yet been in the 'processing' state (see the beginning of Section 4.1).

3561 **4.3.14.3 time-at-completed (integer(MIN:MAX))**

3562 This REQUIRED attribute indicates the time at which the Job object completed (or was cancelled or  
3563 aborted). The out-of-band 'no-value' value is returned if the job has not yet completed, been canceled, or  
3564 aborted (see the beginning of Section 4.1).

3565 **4.3.14.4 job-printer-up-time (integer(1:MAX))**

3566 This REQUIRED Job Description attribute indicates the amount of time (in seconds) that the Printer  
3567 implementation has been up and running. This attribute is an alias for the "printer-up-time" Printer  
3568 Description attribute (see Section 4.4.29).

3569 A client MAY request this attribute in a Get-Job-Attributes or Get-Jobs request and use the value returned  
3570 in combination with other requested Event Time Job Description Attributes in order to display time  
3571 attributes to a user. The difference between this attribute and the 'integer' value of a "time-at-xxx" attribute  
3572 is the number of seconds ago that the "time-at-xxx" event occurred. A client can compute the wall-clock  
3573 time at which the "time-at-xxx" event occurred by subtracting this difference from the client's wall-clock  
3574 time.

3575 **4.3.14.5 date-time-at-creation (dateTime)**

3576 This attribute indicates the date and time at which the Job object was created.

3577 **4.3.14.6 date-time-at-processing (dateTime)**

3578 This attribute indicates the date and time at which the Job object first began processing after the create  
3579 operation or the most recent Restart-Job operation.

3580 **4.3.14.7 date-time-at-completed (dateTime)**

3581 This attribute indicates the date and time at which the Job object completed (or was cancelled or aborted).



3582

**3583 4.3.15 number-of-intervening-jobs (integer(0:MAX))**

3584 This attribute indicates the number of jobs that are "ahead" of this job in the relative chronological order of  
3585 expected time to complete (i.e., the current scheduled order). For efficiency, it is only necessary to calculate  
3586 this value when an operation is performed that requests this attribute.

**3587 4.3.16 job-message-from-operator (text(127))**

3588 This attribute provides a message from an operator, system administrator or "intelligent" process to indicate  
3589 to the end user the reasons for modification or other management action taken on a job.

**3590 4.3.17 Job Size Attributes**

3591 This sub-section defines job attributes that describe the size of the job. These attributes are not intended to  
3592 be counters; they are intended to be useful routing and scheduling information if known. For these  
3593 attributes, the Printer object may try to compute the value if it is not supplied in the create request. Even if  
3594 the client does supply a value for these three attributes in the create request, the Printer object MAY choose  
3595 to change the value if the Printer object is able to compute a value which is more accurate than the client  
3596 supplied value. The Printer object may be able to determine the correct value for these attributes either  
3597 right at job submission time or at any later point in time.

**3598 4.3.17.1 job-k-octets (integer(0:MAX))**

3599 This attribute specifies the total size of the document(s) in K octets, i.e., in units of 1024 octets requested to  
3600 be processed in the job. The value MUST be rounded up, so that a job between 1 and 1024 octets MUST  
3601 be indicated as being 1, 1025 to 2048 MUST be 2, etc.

3602 This value MUST NOT include the multiplicative factors contributed by the number of copies specified by  
3603 the "copies" attribute, independent of whether the device can process multiple copies without making  
3604 multiple passes over the job or document data and independent of whether the output is collated or not.  
3605 Thus the value is independent of the implementation and indicates the size of the document(s) measured in  
3606 K octets independent of the number of copies.

3607 This value MUST also not include the multiplicative factor due to a copies instruction embedded in the  
3608 document data. If the document data actually includes replications of the document data, this value will  
3609 include such replication. In other words, this value is always the size of the source document data, rather  
3610 than a measure of the hardcopy output to be produced.

**3611 4.3.17.2 job-impressions (integer(0:MAX))**

3612 This attribute specifies the total size in number of impressions of the document(s) being submitted (see the  
3613 definition of impression in section 12.2.5).

3614 As with "job-k-octets", this value MUST NOT include the multiplicative factors contributed by the number  
3615 of copies specified by the "copies" attribute, independent of whether the device can process multiple copies  
3616 without making multiple passes over the job or document data and independent of whether the output is  
3617 collated or not. Thus the value is independent of the implementation and reflects the size of the  
3618 document(s) measured in impressions independent of the number of copies.

3619 As with "job-k-octets", this value MUST also not include the multiplicative factor due to a copies  
3620 instruction embedded in the document data. If the document data actually includes replications of the  
3621 document data, this value will include such replication. In other words, this value is always the number of  
3622 impressions in the source document data, rather than a measure of the number of impressions to be  
3623 produced by the job.

#### 3624 **4.3.17.3 job-media-sheets (integer(0:MAX))**

3625 This attribute specifies the total number of media sheets to be produced for this job.

3626 Unlike the "job-k-octets" and the "job-impressions" attributes, this value MUST include the multiplicative  
3627 factors contributed by the number of copies specified by the "copies" attribute and a 'number of copies'  
3628 instruction embedded in the document data, if any. This difference allows the system administrator to  
3629 control the lower and upper bounds of both (1) the size of the document(s) with "job-k-octets-supported"  
3630 and "job-impressions-supported" and (2) the size of the job with "job-media-sheets-supported".

#### 3631 **4.3.18 Job Progress Attributes**

3632 This sub-section defines job attributes that describe the progress of the job. These attributes are intended to  
3633 be counters. That is, the value for a job that has not started processing MUST be 0. When the job's "job-  
3634 state" is 'processing' or 'processing-stopped', this value is intended to contain the amount of the job that has  
3635 been processed to the time at which the attributes are requested. When the job enters the 'completed',  
3636 'canceled', or 'aborted' states, these values are the final values for the job.

##### 3637 **4.3.18.1 job-k-octets-processed (integer(0:MAX))**

3638 This attribute specifies the total number of octets processed in K octets, i.e., in units of 1024 octets so far.  
3639 The value MUST be rounded up, so that a job between 1 and 1024 octets inclusive MUST be indicated as  
3640 being 1, 1025 to 2048 inclusive MUST be 2, etc.

3641 For implementations where multiple copies are produced by the interpreter with only a single pass over the  
3642 data, the final value MUST be equal to the value of the "job-k-octets" attribute. For implementations where  
3643 multiple copies are produced by the interpreter by processing the data for each copy, the final value MUST  
3644 be a multiple of the value of the "job-k-octets" attribute.

##### 3645 **4.3.18.2 job-impressions-completed (integer(0:MAX))**

3646 This job attribute specifies the number of impressions completed for the job so far. For printing devices,  
3647 the impressions completed includes interpreting, marking, and stacking the output.

3648 **4.3.18.3 job-media-sheets-completed (integer(0:MAX))**

3649 This job attribute specifies the media-sheets completed marking and stacking for the entire job so far  
3650 whether those sheets have been processed on one side or on both.

3651 **4.3.19 attributes-charset (charset)**

3652 This REQUIRED attribute is populated using the value in the client supplied "attributes-charset" attribute in  
3653 the create request. It identifies the charset (coded character set and encoding method) used by any Job  
3654 attributes with attribute syntax 'text' and 'name' that were supplied by the client in the create request. See  
3655 Section 3.1.4 for a complete description of the "attributes-charset" operation attribute.

3656 This attribute does not indicate the charset in which the 'text' and 'name' values are stored internally in the  
3657 Job object. The internal charset is implementation-defined. The IPP object MUST convert from whatever  
3658 the internal charset is to that being requested in an operation as specified in Section 3.1.4.

3659 **4.3.20 attributes-natural-language (naturalLanguage)**

3660 This REQUIRED attribute is populated using the value in the client supplied "attributes-natural-language"  
3661 attribute in the create request. It identifies the natural language used for any Job attributes with attribute  
3662 syntax 'text' and 'name' that were supplied by the client in the create request. See Section 3.1.4 for a  
3663 complete description of the "attributes-natural-language" operation attribute. See Sections 4.1.1.2 and  
3664 4.1.2.2 for how a Natural Language Override may be supplied explicitly for each 'text' and 'name' attribute  
3665 value that differs from the value identified by the "attributes-natural-language" attribute.

3666 **4.4 Printer Description Attributes**

3667 These attributes form the attribute group called "printer-description". The following table summarizes  
3668 these attributes, their syntax, and whether or not they are REQUIRED for a Printer object to support. If  
3669 they are not indicated as REQUIRED, they are OPTIONAL. The maximum size in octets for 'text' and  
3670 'name' attributes is indicated in parentheses.

3671 Note: How these attributes are set by an Administrator is outside the scope of this IPP/1.1 document.

3672	+-----+-----+-----+
3673	Attribute   Syntax   REQUIRED?
3674	+-----+-----+-----+
3675	printer-uri-supported   1setOf uri   REQUIRED
3676	+-----+-----+-----+
3677	uri-security-supported   1setOf type2 keyword   REQUIRED
3678	+-----+-----+-----+
3679	uri-authentication- supported   1setOf type2 keyword   REQUIRED
3680	+-----+-----+-----+
3681	+-----+-----+-----+
3682	printer-name   name (127)   REQUIRED
3683	+-----+-----+-----+
3684	printer-location   text (127)
3685	+-----+-----+-----+
3686	printer-info   text (127)
3687	+-----+-----+-----+
3688	printer-more-info   uri
3689	+-----+-----+-----+
3690	printer-driver-installer   uri
3691	+-----+-----+-----+
3692	printer-make-and-model   text (127)
3693	+-----+-----+-----+
3694	printer-more-info- manufacturer   uri
3695	+-----+-----+-----+
3696	+-----+-----+-----+
3697	printer-state   type1 enum   REQUIRED
3698	+-----+-----+-----+
3699	printer-state-reasons   1setOf type2 keyword   REQUIRED
3700	+-----+-----+-----+
3701	printer-state-message   text (MAX)
3702	+-----+-----+-----+
3703	ipp-versions-supported   1setOf type2 keyword   REQUIRED
3704	+-----+-----+-----+
3705	operations-supported   1setOf type2 enum   REQUIRED
3706	+-----+-----+-----+
3707	<del>ipp</del> -multiple-document-jobs- <u>      </u>   boolean
3708	
3709	supported
3710	+-----+-----+-----+
3711	charset-configured   charset   REQUIRED
3712	+-----+-----+-----+
3713	charset-supported   1setOf charset   REQUIRED
3714	+-----+-----+-----+
3715	natural-language-configured   naturalLanguage   REQUIRED
3716	+-----+-----+-----+
3717	generated-natural-language- supported   1setOf naturalLanguage   REQUIRED
3718	+-----+-----+-----+
3719	+-----+-----+-----+
3720	document-format-default   mimeType   REQUIRED

3721	+-----+-----+-----+
3722	document-format-supported   1setOf mimeTypeType   REQUIRED
3723	+-----+-----+-----+
3724	printer-is-accepting-jobs   boolean   REQUIRED
3725	+-----+-----+-----+
3726	queued-job-count   integer (0:MAX)   REQUIRED
3727	+-----+-----+-----+
3728	printer-message-from-   text (127)
3729	operator
3730	+-----+-----+-----+
3731	color-supported   boolean
3732	+-----+-----+-----+
3733	reference-uri-schemes-   1setOf uriScheme
3734	supported
3735	+-----+-----+-----+
3736	pdl-override-supported   type2 keyword   REQUIRED
3737	+-----+-----+-----+
3738	printer-up-time   integer (1:MAX)   REQUIRED
3739	+-----+-----+-----+
3740	printer-current-time   dateTime
3741	+-----+-----+-----+
3742	multiple-operation-time-out   integer (1:MAX)
3743	+-----+-----+-----+
3744	compression-supported   1setOf type3 keyword   REQUIRED
3745	+-----+-----+-----+
3746	job-k-octets-supported   rangeOfInteger (0:MAX)
3747	+-----+-----+-----+
3748	job-impressions-supported   rangeOfInteger (0:MAX)
3749	+-----+-----+-----+
3750	job-media-sheets-supported   rangeOfInteger (0:MAX)
3751	+-----+-----+-----+
3752	pages-per-minute   integer(0:MAX)
3753	+-----+-----+-----+
3754	pages-per-minute-color   integer(0:MAX)
3755	+-----+-----+-----+
3756	

#### 3757 4.4.1 printer-uri-supported (1setOf uri)

3758 This REQUIRED Printer attribute contains at least one URI for the Printer object. It OPTIONALLY  
3759 contains more than one URI for the Printer object. An administrator determines a Printer object's URI(s)  
3760 and configures this attribute to contain those URIs by some means outside the scope of this IPP/1.1  
3761 document. The precise format of this URI is implementation dependent and depends on the protocol. See  
3762 the next two sections for a description of the "uri-security-supported" and "uri-authentication-supported"  
3763 attributes, both of which are the REQUIRED companion attributes to this "printer-uri-supported" attribute.  
3764 See section 2.4 on Printer object identity and section 8.2 on security and URIs for more information.

**3765 4.4.2 uri-authentication-supported (1setOf type2 keyword)**

3766 This REQUIRED Printer attribute MUST have the same cardinality (contain the same number of values) as  
3767 the "printer-uri-supported" attribute. This attribute identifies the Client Authentication mechanism  
3768 associated with each URI listed in the "printer-uri-supported" attribute. The Printer object uses the specified  
3769 mechanism to identify the authenticated user (see section 8.3) . The "i th" value in "uri-authentication-  
3770 supported" corresponds to the "i th" value in "printer-uri-supported" and it describes the authentication  
3771 mechanisms used by the Printer when accessed via that URI. See [IPP-PRO] for more details on Client  
3772 Authentication.

3773 The following standard keyword values are defined:

- 3774 'none': There is no authentication mechanism associated with the URI. The Printer object assumes that  
3775 the authenticated user is "anonymous".
- 3776 'requesting-user-name': When a client performs an operation whose target is the associated URI, the  
3777 Printer object assumes that the authenticated user is specified by the "requesting-user-name"  
3778 Operation attribute (see section 8.3). If the "requesting-user-name" attribute is absent in a request,  
3779 the Printer object assumes that the authenticated user is "anonymous".
- 3780 'basic': When a client performs an operation whose target is the associated URI, the Printer object  
3781 challenges the client with HTTP basic authentication [RFC2617]. The Printer object assumes that  
3782 the authenticated user is the name received via the basic authentication mechanism.
- 3783 'digest': When a client performs an operation whose target is the associated URI, the Printer object  
3784 challenges the client with HTTP digest authentication [RFC2617]. The Printer object assumes that  
3785 the authenticated user is the name received via the digest authentication mechanism.
- 3786 'certificate': When a client performs an operation whose target is the associated URI, the Printer object  
3787 expects the client to provide a certificate. The Printer object assumes that the authenticated user is  
3788 the textual name contained within the certificate.

**3789 4.4.3 uri-security-supported (1setOf type2 keyword)**

3790 This REQUIRED Printer attribute MUST have the same cardinality (contain the same number of values) as  
3791 the "printer-uri-supported" attribute. This attribute identifies the security mechanisms used for each URI  
3792 listed in the "printer-uri-supported" attribute. The "i th" value in "uri-security-supported" corresponds to  
3793 the "i th" value in "printer-uri-supported" and it describes the security mechanisms used for accessing the  
3794 Printer object via that URI. See [IPP-PRO] for more details on security mechanisms.

3795 The following standard keyword values are defined:

- 3796 'none': There are no secure communication channel protocols in use for the given URI.
- 3797 'ssl3': SSL3 [SSL] is the secure communications channel protocol in use for the given URI.
- 3798 'tls': TLS [RFC2246] is the secure communications channel protocol in use for the given URI.
- 3799

3800 This attribute is orthogonal to the definition of a Client Authentication mechanism. Specifically, 'none'  
3801 does not exclude Client Authentication. See section 4.4.2.

3802 Consider the following example. For a single Printer object, an administrator configures the "printer-uri-  
3803 supported", "uri-authentication-supported" and "uri-security-supported" attributes as follows:

```
3804 "printer-uri-supported": 'xxx://acme.com/open-use-printer', 'xxx://acme.com/restricted-use-printer',  
3805 'xxx://acme.com/private-printer'  
3806 "uri-authentication-supported": 'none', 'digest', 'basic'  
3807 "uri-security-supported": 'none', 'none', 'tls'  
3808
```

3809 Note: 'xxx' is not a valid scheme. See the IPP/1.1 "Transport and Encoding" document [IPP-PRO] for the  
3810 actual URI schemes to be used in object target attributes.

3811 In this case, one Printer object has three URIs.

- 3812 - For the first URI, 'xxx://acme.com/open-use-printer', the value 'none' in "uri-security-supported"  
3813 indicates that there is no secure channel protocol configured to run under HTTP. The value of 'none'  
3814 in "uri-authentication-supported" indicates that all users are 'anonymous'. There will be no  
3815 challenge and the Printer will ignore "requesting-user-name".
- 3816 - For the second URI, 'xxx://acme.com/restricted-use-printer', the value 'none' in "uri-security-  
3817 supported" indicates that there is no secure channel protocol configured to run under HTTP. The  
3818 value of 'digest' in "uri-authentication-supported" indicates that the Printer will issue a challenge and  
3819 that the Printer will use the name supplied by the digest mechanism to determine the authenticated  
3820 user (see section 8.3).
- 3821 - For the third URI, 'xxx://acme.com/private-printer', the value 'tls' in "uri-security-supported" indicates  
3822 that TLS is being used to secure the channel. The client SHOULD be prepared to use TLS framing  
3823 to negotiate an acceptable ciphersuite to use while communicating with the Printer object. In this  
3824 case, the name implies the use of a secure communications channel, but the fact is made explicit by  
3825 the presence of the 'tls' value in "uri-security-supported". The client does not need to resort to  
3826 understanding which security it must use by following naming conventions or by parsing the URI to  
3827 determine which security mechanisms are implied. The value of 'basic' in "uri-authentication-  
3828 supported" indicates that the Printer will issue a challenge and that the Printer will use the name  
3829 supplied by the digest mechanism to determine the authenticated user (see section 8.3) . Because  
3830 this challenge occurs in a tls session, the channel is secure.

3831

3832 It is expected that many IPP Printer objects will be configured to support only one channel (either  
3833 configured to use TLS access or not) and only one authentication mechanism. Such Printer objects only  
3834 have one URI listed in the "printer-uri-supported" attribute. No matter the configuration of the Printer  
3835 object (whether it has only one URI or more than one URI), a client MUST supply only one URI in the  
3836 target "printer-uri" operation attribute.

#### 3837 4.4.4 printer-name (name(127))

3838 This REQUIRED Printer attribute contains the name of the Printer object. It is a name that is more end-  
3839 user friendly than a URI. An administrator determines a printer's name and sets this attribute to that name.  
3840 This name may be the last part of the printer's URI or it may be unrelated. In non-US-English locales, a  
3841 name may contain characters that are not allowed in a URI.



**3842 4.4.5 printer-location (text(127))**

3843 This Printer attribute identifies the location of the device. This could include things like: "in Room 123A,  
3844 second floor of building XYZ".

**3845 4.4.6 printer-info (text(127))**

3846 This Printer attribute identifies the descriptive information about this Printer object. This could include  
3847 things like: "This printer can be used for printing color transparencies for HR presentations", or "Out of  
3848 courtesy for others, please print only small (1-5 page) jobs at this printer", or even "This printer is going  
3849 away on July 1, 1997, please find a new printer".

**3850 4.4.7 printer-more-info (uri)**

3851 This Printer attribute contains a URI used to obtain more information about this specific Printer object. For  
3852 example, this could be an HTTP type URI referencing an HTML page accessible to a Web Browser. The  
3853 information obtained from this URI is intended for end user consumption. Features outside the scope of IPP  
3854 can be accessed from this URI. The information is intended to be specific to this printer instance and site  
3855 specific services (e.g. job pricing, services offered, end user assistance). The device manufacturer may  
3856 initially populate this attribute.

**3857 4.4.8 printer-driver-installer (uri)**

3858 This Printer attribute contains a URI to use to locate the driver installer for this Printer object. This  
3859 attribute is intended for consumption by automata. The mechanics of print driver installation is outside the  
3860 scope of this IPP/1.1 document. The device manufacturer may initially populate this attribute.

**3861 4.4.9 printer-make-and-model (text(127))**

3862 This Printer attribute identifies the make and model of the device. The device manufacturer may initially  
3863 populate this attribute.

**3864 4.4.10 printer-more-info-manufacturer (uri)**

3865 This Printer attribute contains a URI used to obtain more information about this type of device. The  
3866 information obtained from this URI is intended for end user consumption. Features outside the scope of  
3867 IPP can be accessed from this URI (e.g., latest firmware, upgrades, print drivers, optional features available,  
3868 details on color support). The information is intended to be germane to this printer without regard to site  
3869 specific modifications or services. The device manufacturer may initially populate this attribute.

#### 3870 **4.4.11 printer-state (type1 enum)**

3871 This REQUIRED Printer attribute identifies the current state of the device. The "printer-state reasons"  
3872 attribute augments the "printer-state" attribute to give more detailed information about the Printer in the  
3873 given printer state.

3874 A Printer object need only update this attribute before responding to an operation which requests the  
3875 attribute; the Printer object NEED NOT update this attribute continually, since asynchronous event  
3876 notification is not part of IPP/1.1. A Printer NEED NOT implement all values if they are not applicable to  
3877 a given implementation.

3878 The following standard enum values are defined:

3879 Value	Symbolic Name and Description
------------	-------------------------------

3880

3881 '3'	'idle': Indicates that new jobs can start processing without waiting.
----------	---

3882 '4'	'processing': Indicates that jobs are processing; new jobs will wait before processing.
----------	---

3883 '5'	'stopped': Indicates that no jobs can be processed and intervention is required.
----------	--

3884 Values of "printer-state-reasons", such as 'spool-area-full' and 'stopped-partly', MAY be used to provide  
3885 further information.

#### 3886 **4.4.12 printer-state-reasons (1setOf type2 keyword)**

3887 This REQUIRED Printer attribute supplies additional detail about the device's state. Some of the these  
3888 value definitions indicate conformance requirements; the rest are OPTIONAL.

3889 Each keyword value MAY have a suffix to indicate its level of severity. The three levels are: report (least  
3890 severe), warning, and error (most severe).

3891 - '-report': This suffix indicates that the reason is a "report". An implementation may choose to omit  
3892 some or all reports. Some reports specify finer granularity about the printer state; others serve as a  
3893 precursor to a warning. A report MUST contain nothing that could affect the printed output.

3894 - '-warning': This suffix indicates that the reason is a "warning". An implementation may choose to omit  
3895 some or all warnings. Warnings serve as a precursor to an error. A warning MUST contain nothing  
3896 that prevents a job from completing, though in some cases the output may be of lower quality.

3897 - '-error': This suffix indicates that the reason is an "error". An implementation MUST include all  
3898 errors. If this attribute contains one or more errors, printer MUST be in the stopped state.  
3899

3900 If the implementation does not add any one of the three suffixes, all parties MUST assume that the reason is  
3901 an "error".

3902 If a Printer object controls more than one output device, each value of this attribute MAY apply to one or  
3903 more of the output devices. An error on one output device that does not stop the Printer object as a whole  
3904 MAY appear as a warning in the Printer's "printer-state-reasons attribute". If the "printer-state" for such a  
3905 Printer has a value of 'stopped', then there MUST be an error reason among the values in the "printer-state-  
3906 reasons" attribute.

3907 The following standard keyword values are defined:

3908 'other': The device has detected an error other than one listed in this document.

3909 'none': There are not reasons. This state reason is semantically equivalent to "printer-state-reasons"  
3910 without any value and MUST be used, since the 1setOf attribute syntax requires at least one value.

3911 'media-needed': A tray has run out of media.

3912 'media-jam': The device has a media jam.

3913 'moving-to-paused': Someone has paused the Printer object using the Pause-Printer operation (see  
3914 section 3.2.7) or other means, but the device(s) are taking an appreciable time to stop. Later, when  
3915 all output has stopped, the "printer-state" becomes 'stopped', and the 'paused' value replaces the  
3916 'moving-to-paused' value in the "printer-state-reasons" attribute. This value MUST be supported, if  
3917 the Pause-Printer operation is supported and the implementation takes significant time to pause a  
3918 device in certain circumstances.

3919 'paused': Someone has paused the Printer object using the Pause-Printer operation (see section 3.2.7) or  
3920 other means and the Printer object's "printer-state" is 'stopped'. In this state, a Printer MUST NOT  
3921 produce printed output, but it MUST perform other operations requested by a client. If a Printer had  
3922 been printing a job when the Printer was paused, the Printer MUST resume printing that job when  
3923 the Printer is no longer paused and leave no evidence in the printed output of such a pause. This  
3924 value MUST be supported, if the Pause-Printer operation is supported.

3925 'shutdown': Someone has removed a Printer object from service, and the device may be powered down  
3926 or physically removed. In this state, a Printer object MUST NOT produce printed output, and unless  
3927 the Printer object is realized by a print server that is still active, the Printer object MUST perform no  
3928 other operations requested by a client, including returning this value. If a Printer object had been  
3929 printing a job when it was shutdown, the Printer NEED NOT resume printing that job when the  
3930 Printer is no longer shutdown. If the Printer resumes printing such a job, it may leave evidence in  
3931 the printed output of such a shutdown, e.g. the part printed before the shutdown may be printed a  
3932 second time after the shutdown.

3933 'connecting-to-device': The Printer object has scheduled a job on the output device and is in the process  
3934 of connecting to a shared network output device (and might not be able to actually start printing the  
3935 job for an arbitrarily long time depending on the usage of the output device by other servers on the  
3936 network).

3937 'timed-out': The server was able to connect to the output device (or is always connected), but was unable  
3938 to get a response from the output device.

3939 'stopping': The Printer object is in the process of stopping the device and will be stopped in a while.  
3940 When the device is stopped, the Printer object will change the Printer object's state to 'stopped'. The  
3941 'stopping-warning' reason is never an error, even for a Printer with a single output device. When an  
3942 output-device ceases accepting jobs, the Printer will have this reason while the output device  
3943 completes printing.

3944 'stopped-partly': When a Printer object controls more than one output device, this reason indicates that  
3945 one or more output devices are stopped. If the reason is a report, fewer than half of the output  
3946 devices are stopped. If the reason is a warning, fewer than all of the output devices are stopped.

3947 'toner-low': The device is low on toner.

3948 'toner-empty': The device is out of toner.

3949 'spool-area-full': The limit of persistent storage allocated for spooling has been reached. The Printer is  
3950 temporarily unable to accept more jobs. The Printer will remove this value when it is able to accept

3951 more jobs. This value SHOULD be used by a non-spooling Printer that only accepts one or a small  
3952 number jobs at a time or a spooling Printer that has filled the spool space.

3953 'cover-open': One or more covers on the device are open.

3954 'interlock-open': One or more interlock devices on the printer are unlocked.

3955 'door-open': One or more doors on the device are open.

3956 'input-tray-missing': One or more input trays are not in the device.

3957 'media-low': At least one input tray is low on media.

3958 'media-empty': At least one input tray is empty.

3959 'output-tray-missing': One or more output trays are not in the device

3960 'output-area-almost-full': One or more output area is almost full (e.g. tray, stacker, collator).

3961 'output-area-full': One or more output area is full. (e.g. tray, stacker, collator)

3962 'marker-supply-low': The device is low on at least one marker supply. (e.g. toner, ink, ribbon)

3963 'marker-supply-empty': The device is out of at least one marker supply. (e.g. toner, ink, ribbon)

3964 'marker-waste-almost-full': The device marker supply waste receptacle is almost full.

3965 'marker-waste-full': The device marker supply waste receptacle is full.

3966 'fuser-over-temp': The fuser temperature is above normal.

3967 'fuser-under-temp': The fuser temperature is below normal.

3968 'opc-near-eol': The optical photo conductor is near end of life.

3969 'opc-life-over': The optical photo conductor is no longer functioning.

3970 'developer-low': The device is low on developer.

3971 'developer-empty': The device is out of developer.

3972 'interpreter-resource-unavailable': An interpreter resource is unavailable (i.e. font, form)

3973

#### 3974 **4.4.13 printer-state-message (text(MAX))**

3975 This Printer attribute specifies ~~the additional~~ information about the "printer-state" and "printer-state-  
3976 reasons" attributes in human readable text. If the Printer object supports this attribute, the Printer object  
3977 MUST be able to generate this message in any of the natural languages identified by the Printer's  
3978 "generated-natural-language-supported" attribute (see the "attributes-natural-language" operation attribute  
3979 specified in Section 3.1.4.1).

#### 3980 **4.4.14 ipp-versions-supported (1setOf type2 keyword)**

3981 This REQUIRED attribute identifies the IPP protocol version(s) that this Printer supports, including major  
3982 and minor versions, i.e., the version numbers for which this Printer implementation meets the conformance  
3983 requirements. For version number validation, the Printer matches the (two-octet binary) "version-number"  
3984 parameter supplied by the client in each request (see sections 3.1.1 and 3.1.8) with the (US-ASCII) keyword  
3985 values of this attribute.

3986 The following standard keyword values are defined:

3987 '1.0': Meets the conformance requirement of IPP version 1.0 as specified in RFC 2566 [RFC2566] and  
3988 RFC 2565 [RFC2565] including any extensions registered according to Section 6 and any extension  
3989 defined in this version or any future version of the IPP "Model and Semantics" document or the IPP

3990 "Encoding and Transport" document following the rules, if any, when the "version-number"  
 3991 parameter is '1.0'.  
 3992 '1.1': Meets the conformance requirement of IPP version 1.1 as specified in this document and [IPP-  
 3993 PRO] including any extensions registered according to Section 6 and any extension defined in any  
 3994 future versions of the IPP "Model and Semantics" document or the IPP Encoding and Transport  
 3995 document following the rules, if any, when the "version-number" parameter is '1.1'.

#### 3996 **4.4.15 operations-supported (1setOf type2 enum)**

3997 This REQUIRED Printer attribute specifies the set of supported operations for this Printer object and  
 3998 contained Job objects.

3999 This attribute is encoded as any other enum attribute syntax according to [IPP-PRO] as 32-bits. However,  
 4000 all 32-bit enum values for this attribute MUST NOT exceed 0x00008FFF, since these same values are also  
 4001 passed in two octets in the "operation-id" parameter (see section 3.1.1) in each Protocol request with the  
 4002 two high order octets omitted in order to indicate the operation being performed [IPP-PRO].

4003 The following standard enum and "operation-id" (see section 3.1.2) values are defined:

4004	Value	Operation Name
4005	-----	-----
4006		
4007	0x0000	reserved, not used
4008	0x0001	reserved, not used
4009	0x0002	Print-Job
4010	0x0003	Print-URI
4011	0x0004	Validate-Job
4012	0x0005	Create-Job
4013	0x0006	Send-Document
4014	0x0007	Send-URI
4015	0x0008	Cancel-Job
4016	0x0009	Get-Job-Attributes
4017	0x000A	Get-Jobs
4018	0x000B	Get-Printer-Attributes
4019	0x000C	Hold-Job
4020	0x000D	Release-Job
4021	0x000E	Restart-Job
4022	0x000F	reserved for a future operation
4023	0x0010	Pause-Printer
4024	0x0011	Resume-Printer
4025	0x0012	Purge-Jobs
4026	0x0013-0x3FFF	reserved for future <a href="#">IETF standards track</a> operations ( <a href="#">see section 6.4</a> )
4027	0x4000-0x8FFF	reserved for <a href="#">private-vendor</a> extensions ( <a href="#">see section 6.4</a> )
4028		

4029 ~~The reserved block for private extensions allows for vendors to implement private extensions that are~~  
4030 ~~guaranteed to not conflict with future registered extensions. However, there is no guarantee that two or~~  
4031 ~~more private extensions will not conflict.~~

#### 4032 **4.4.16 multiple-document-jobs-supported (boolean)**

4033 This Printer attribute indicates whether or not the Printer supports more than one document per job, i.e.,  
4034 more than one Send-Document or Send-Data operation with document data. If the Printer supports the  
4035 Create-Job and Send-Document operations (see section 3.2.4 and 3.3.1), it MUST support this attribute.

#### 4036 **4.4.17 charset-configured (charset)**

4037 This REQUIRED Printer attribute identifies the charset that the Printer object has been configured to  
4038 represent 'text' and 'name' Printer attributes that are set by the operator, system administrator, or  
4039 manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info" (text), and "printer-  
4040 make-and-model" (text). Therefore, the value of the Printer object's "charset-configured" attribute MUST  
4041 also be among the values of the Printer object's "charset-supported" attribute.

#### 4042 **4.4.18 charset-supported (1setOf charset)**

4043 This REQUIRED Printer attribute identifies the set of charsets that the Printer and contained Job objects  
4044 support in attributes with attribute syntax 'text' and 'name'. At least the value 'utf-8' MUST be present, since  
4045 IPP objects MUST support the UTF-8 [RFC2279] charset. If a Printer object supports a charset, it means  
4046 that for all attributes of syntax 'text' and 'name' the IPP object MUST (1) accept the charset in requests and  
4047 return the charset in responses as needed.

4048 If more charsets than UTF-8 are supported, the IPP object MUST perform charset conversion between the  
4049 charsets as described in Section 3.1.4.2.

#### 4050 **4.4.19 natural-language-configured (naturalLanguage)**

4051 This REQUIRED Printer attribute identifies the natural language that the Printer object has been configured  
4052 to represent 'text' and 'name' Printer attributes that are set by the operator, system administrator, or  
4053 manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info" (text), and "printer-  
4054 make-and-model" (text). When returning these Printer attributes, the Printer object MAY return them in the  
4055 configured natural language specified by this attribute, instead of the natural language requested by the  
4056 client in the "attributes-natural-language" operation attribute. See Section 3.1.4.1 for the specification of  
4057 the OPTIONAL multiple natural language support. Therefore, the value of the Printer object's "natural-  
4058 language-configured" attribute MUST also be among the values of the Printer object's "generated-natural-  
4059 language-supported" attribute.



**4.4.20 generated-natural-language-supported (1setOf naturalLanguage)**

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

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

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

**4.4.21 document-format-default (mimeMediaType)**

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

**4.4.22 document-format-supported (1setOf mimeMediaType)**

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

**4.4.23 printer-is-accepting-jobs (boolean)**

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

This value is independent of the "printer-state" and "printer-state-reasons" attributes because its value does not affect the current job; rather it affects future jobs. This attribute, when 'false', causes the Printer to reject jobs even when the "printer-state" is 'idle' or, when 'true', causes the Printer object to accept jobs even when the "printer-state" is 'stopped'.



4091 **4.4.24 queued-job-count (integer(0:MAX))**

4092 This REQUIRED Printer attribute contains a count of the number of jobs that are either 'pending',  
4093 'processing', 'pending-held', or 'processing-stopped' and is set by the Printer object.

4094 **4.4.25 printer-message-from-operator (text(127))**

4095 This Printer attribute provides a message from an operator, system administrator or "intelligent" process to  
4096 indicate to the end user information or status of the printer, such as why it is unavailable or when it is  
4097 expected to be available.

4098 **4.4.26 color-supported (boolean)**

4099 This Printer attribute identifies whether the device is capable of any type of color printing at all, including  
4100 highlight color. All document instructions having to do with color are embedded within the document PDL  
4101 (none are external IPP attributes in IPP/1.1).

4102 Note: end-users are able to determine the nature and details of the color support by querying the "printer-  
4103 more-info-manufacturer" Printer attribute.

4104 **4.4.27 reference-uri-schemes-supported (1setOf uriScheme)**

4105 This Printer attribute specifies which URI schemes are supported for use in the "document-uri" operation  
4106 attribute of the Print-URI or Send-URI operation. If a Printer object supports these optional operations, it  
4107 MUST support the "reference-uri-schemes-supported" Printer attribute with at least the following schemed  
4108 URI value:

4109 'ftp': The Printer object will use an FTP 'get' operation as defined in RFC 2228 [RFC2228] using FTP  
4110 URLs as defined by [RFC2396] and [RFC2316].  
4111

4112 The Printer object MAY OPTIONALLY support other URI schemes (see section 4.1.6).

4113 **4.4.28 pdl-override-supported (type2 keyword)**

4114 This REQUIRED Printer attribute expresses the ability for a particular Printer implementation to either  
4115 attempt to override document data instructions with IPP attributes or not.

4116 This attribute takes on the following values:

- 4117 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values take  
4118 precedence over embedded instructions in the document data, however there is no guarantee.
- 4119 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP attribute  
4120 values take precedence over embedded instructions in the document data.  
4121

4122 Section 15 contains a full description of how this attribute interacts with and affects other IPP attributes,  
4123 especially the "ipp-attribute-fidelity" attribute.

#### 4124 **4.4.29 printer-up-time (integer(1:MAX))**

4125 This REQUIRED Printer attribute indicates the amount of time (in seconds) that this Printer instance has  
4126 been up and running. The value is a monotonically increasing value starting from 1 when the Printer object  
4127 is started-up (initialized, booted, etc.). This value is used to populate the Event Time Job Description Job  
4128 attributes "time-at-creation", "time-at-processing", and "time-at-completed" (see section 4.3.14).

4129 If the Printer object goes down at some value 'n', and comes back up, the implementation MAY:

- 4130 1. Know how long it has been down, and resume at some value greater than 'n', or
- 4131 2. Restart from 1.

4132 In other words, if the device or devices that the Printer object is representing are restarted or power cycled,  
4133 the Printer object MAY continue counting this value or MAY reset this value to 1 depending on  
4134 implementation. However, if the Printer object software ceases running, and restarts without knowing the  
4135 last value for "printer-up-time", the implementation MUST reset this value to 1. If this value is reset and  
4136 the Printer has persistent jobs, the Printer MUST reset the "time-at-xxx(integer) Event Time Job  
4137 Description attributes according to Section 4.3.14. An implementation MAY use both implementation  
4138 alternatives, depending on warm versus cold start, respectively.

#### 4139 **4.4.30 printer-current-time (dateTime)**

4140 This Printer attribute indicates the current date and time. This value is used to populate the Event Time Job  
4141 Description attributes: "time-at-creation", "time-at-processing", and "time-at-completed" (see Section  
4142 4.3.14).

4143 The date and time is obtained on a "best efforts basis" and does not have to be that precise in order to work  
4144 in practice. A Printer implementation sets the value of this attribute by obtaining the date and time via  
4145 some implementation-dependent means, such as getting the value from a network time server, initialization  
4146 at time of manufacture, or setting by an administrator. See [IPP-IIG] for examples. If an implementation  
4147 supports this attribute and the implementation knows that it has not yet been set, then the implementation  
4148 MUST return the value of this attribute using the out-of-band 'no-value' meaning not configured. See the  
4149 beginning of section 4.1.

4150 The time zone of this attribute NEED NOT be the time zone used by people located near the Printer object  
4151 or device. The client MUST NOT expect that the time zone of any received 'dateTime' value to be in the  
4152 time zone of the client or in the time zone of the people located near the printer.

4153 The client SHOULD display any dateTime attributes to the user in client local time by converting the  
4154 'dateTime' value returned by the server to the time zone of the client, rather than using the time zone  
4155 returned by the Printer in attributes that use the 'dateTime' attribute syntax.

**4156 4.4.31 multiple-operation-time-out (integer(1:MAX))**

4157 This Printer attribute identifies the minimum time (in seconds) that the Printer object waits for additional  
4158 Send-Document or Send-URI operations to follow a still-open multi-document Job object before taking  
4159 any recovery actions, such as the ones indicated in section 3.3.1. If the Printer object supports the Create-  
4160 Job and Send-Document operations (see section 3.2.4 and 3.3.1), it MUST support this attribute.

4161 It is RECOMMENDED that vendors supply a value for this attribute that is between 60 and 240 seconds.  
4162 An implementation MAY allow a system administrator to set this attribute (by means outside this IPP/1.1  
4163 document). If so, the system administrator MAY be able to set values outside this range.

**4164 4.4.32 compression-supported (1setOf type3 keyword)**

4165 This REQUIRED Printer attribute identifies the set of supported compression algorithms for document  
4166 data. Compression only applies to the document data; compression does not apply to the encoding of the  
4167 IPP operation itself. The supported values are used to validate the client supplied "compression" operation  
4168 attributes in Print-Job, Send-Document, and Send-URI requests.

4169 Standard values are :

4170 'none': no compression is used.  
4171 'deflate': ZIP public domain inflate/deflate) compression technology  
4172 'gzip' GNU zip compression technology described in RFC 1952 [RFC1952].  
4173 'compress': UNIX compression technology  
4174

**4175 4.4.33 job-k-octets-supported (rangeOfInteger(0:MAX))**

4176 This Printer attribute specifies the upper and lower bounds of total sizes of jobs in K octets, i.e., in units of  
4177 1024 octets. The supported values are used to validate the client supplied "job-k-octets" operation attributes  
4178 in create requests. The corresponding job description attribute "job-k-octets" is defined in section 4.3.17.1.

**4179 4.4.34 job-impressions-supported (rangeOfInteger(0:MAX))**

4180 This Printer attribute specifies the upper and lower bounds for the number of impressions per job. The  
4181 supported values are used to validate the client supplied "job-impressions" operation attributes in create  
4182 requests. The corresponding job description attribute "job-impressions" is defined in section 4.3.17.2.

**4183 4.4.35 job-media-sheets-supported (rangeOfInteger(0:MAX))**

4184 This Printer attribute specifies the upper and lower bounds for the number of media sheets per job. The  
4185 supported values are used to validate the client supplied "job-media-sheets" operation attributes in create  
4186 requests. The corresponding Job attribute "job-media-sheets" is defined in section 4.3.17.3.

**4187 4.4.36 pages-per-minute (integer(0:MAX))**

4188 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number which  
4189 may be generated by this printer (e.g., simplex, black-and-white). This attribute is informative, not a  
4190 service guarantee. Generally, it is the value used in the marketing literature to describe the device.

4191 A value of 0 indicates a device that takes more than two minutes to process a page.

**4192 4.4.37 pages-per-minute-color (integer(0:MAX))**

4193 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number which  
4194 may be generated by this printer when printing color (e.g., simplex, color). For purposes of this attribute,  
4195 "color" means the same as for the "color-supported" attribute, namely, the device is capable of any type of  
4196 color printing at all, including highlight color. This attribute is informative, not a service guarantee.  
4197 Generally, it is the value used in the marketing literature to describe the color capabilities of this device.

4198 A value of 0 indicates a device that takes more than two minutes to process a page.

4199 If a color device has several color modes, it MAY use the pages-per-minute value for this attribute that  
4200 corresponds to the mode that produces the highest number.

4201 Black and white only printers MUST NOT support this attribute. If this attribute is present, then the "color-  
4202 supported" Printer description attribute MUST be present and have a 'true' value.

4203 The values of these two attributes returned by the Get-Printer-Attributes operation MAY be affected by the  
4204 "document-format" attribute supplied by the client in the Get-Printer-Attributes request. In other words, the  
4205 implementation MAY have different speeds depending on the document format being processed. See  
4206 section 3.2.5.1 Get-Printer-Attributes.

**4207 5. Conformance**

4208 This section describes conformance issues and requirements. This document introduces model entities such  
4209 as objects, operations, attributes, attribute syntaxes, and attribute values. These conformance sections  
4210 describe the conformance requirements which apply to these model entities.

**4211 5.1 Client Conformance Requirements**

4212 This section describes the conformance requirements for a client (see section 2.1), whether it be:

- 4213 1. contained within software controlled by an end user, e.g. activated by the "Print" menu item in an  
4214 application that sends IPP requests or

- 4215 2. the print server component that sends IPP requests to either an output device or another  
4216 "downstream" print server.

4217 A conforming client MUST support all REQUIRED operations as defined in this document. For each  
4218 attribute included in an operation request, a conforming client MUST supply a value whose type and value  
4219 syntax conforms to the requirements of the Model document as specified in Sections 3 and 4. A  
4220 conforming client MAY supply any ~~registered-IETF standards track~~ extensions and/or ~~private-vendor~~  
4221 extensions in an operation request, as long as they extensions meet the requirements in Section 6.

4222 Otherwise, there are no conformance requirements placed on the user interfaces provided by IPP clients or  
4223 their applications. For example, one application might not allow an end user to submit multiple documents  
4224 per job, while another does. One application might first query a Printer object in order to supply a graphical  
4225 user interface (GUI) dialogue box with supported and default values whereas a different implementation  
4226 might not.

4227 When sending a request, an IPP client NEED NOT supply any attributes that are indicated as  
4228 OPTIONALLY supplied by the client.

4229 A client MUST be able to accept any of the attribute syntaxes defined in Section 4.1, including their full  
4230 range, that may be returned to it in a response from a Printer object. In particular for each attribute that the  
4231 client supports whose attribute syntax is 'text', the client MUST accept and process both the  
4232 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that the client supports  
4233 whose attribute syntax is 'name', the client MUST accept and process both the 'nameWithoutLanguage' and  
4234 'nameWithLanguage' forms. For presentation purposes, truncation of long attribute values is not  
4235 recommended. A recommended approach would be for the client implementation to allow the user to scroll  
4236 through long attribute values.

4237 A response MAY contain attribute groups, attributes, attribute syntaxes, values, and status codes that the  
4238 client does not expect. Therefore, a client implementation MUST gracefully handle such responses and not  
4239 refuse to inter-operate with a conforming Printer that is returning ~~registered-IETF standards track extension~~  
4240 or ~~private-vendor~~ extensions, including attribute groups, attributes, attribute syntaxes, attribute values, and  
4241 status codes, and out-of-band attribute values that conform to Section 6. Clients may choose to ignore any  
4242 parameters, attributes, attribute syntaxes, or values that they do not understand.

4243 While a client is sending data to a printer, it SHOULD do its best to prevent a channel from being closed by  
4244 a lower layer when the channel is blocked (i.e. flow-controlled off) for whatever reason, e.g. 'out of paper'  
4245 or 'job ahead hasn't freed up enough memory'. However, the layer that launched the print submission (e.g.  
4246 an end user) MAY close the channel in order to cancel the job. When a client closes a channel, a Printer  
4247 MAY print all or part of the received portion of the document. See the "Encoding and Transport" document  
4248 [IPP-PRO] for more details.

4249 A client MUST support Client Authentication as defined in the IPP/1.1 Encoding and Transport document  
4250 [IPP-PRO]. A client SHOULD support Operation Privacy and Server Authentication as defined in the  
4251 IPP/1.1 Encoding and Transport document [IPP-PRO]. See also section 8 of this document.

## 4252 5.2 IPP Object Conformance Requirements

4253 This section specifies the conformance requirements for conforming implementations of IPP objects (see  
4254 section 2). These requirements apply to an IPP object whether it is:

4255 (1) an (embedded) device component that accepts IPP requests and controls the device or

4256 (2) a component of a print server that accepts IPP requests (where the print server control one or  
4257 more networked devices using IPP or other protocols).

### 4258 5.2.1 Objects

4259 Conforming implementations MUST implement all of the model objects as defined in this document in the  
4260 indicated sections:

4261 Section 2.1 - Printer Object

4262 Section 2.2 - Job Object

### 4263 5.2.2 Operations

4264 Conforming IPP object implementations MUST implement all of the REQUIRED model operations,  
4265 including REQUIRED responses, as defined in this document in the indicated sections:

4266 For a Printer object:

4267	Print-Job (section 3.2.1)	REQUIRED
4268	Print-URI (section 3.2.2)	OPTIONAL
4269	Validate-Job (section 3.2.3)	REQUIRED
4270	Create-Job (section 3.2.4)	OPTIONAL
4271	Get-Printer-Attributes (section 3.2.5)	REQUIRED
4272	Get-Jobs (section 3.2.6)	REQUIRED
4273	Pause-Printer (section 3.2.7)	OPTIONAL
4274	Resume-Printer (section 3.2.8)	OPTIONAL
4275	Purge-Jobs (section 3.2.9)	OPTIONAL

4276 For a Job object:

4278	Send-Document (section 3.3.1)	OPTIONAL
4279	Send-URI (section 3.3.2)	OPTIONAL
4280	Cancel-Job (section 3.3.3)	REQUIRED
4281	Get-Job-Attributes (section 3.3.4)	REQUIRED
4282	Hold-Job (section 3.3.5)	OPTIONAL
4283	Release-Job (section 3.3.6)	OPTIONAL
4284	Restart-Job (section 3.3.7)	OPTIONAL

4286 Conforming IPP objects MUST support all REQUIRED operation attributes and all values of such  
4287 attributes if so indicated in the description. Conforming IPP objects MUST ignore all unsupported or

4288 unknown operation attributes or operation attribute groups received in a request, but MUST reject a request  
4289 that contains a supported operation attribute that contains an unsupported value.

4290 Conforming IPP objects MAY return operation responses that contain attributes groups, attributes names,  
4291 attribute syntaxes, attribute values, and status codes that are extensions to this standard. The additional  
4292 attribute groups MAY occur in any order.

4293 The following section on object attributes specifies the support required for object attributes.

### 4294 **5.2.3 IPP Object Attributes**

4295 Conforming IPP objects MUST support all of the REQUIRED object attributes, as defined in this document  
4296 in the indicated sections.

4297 If an object supports an attribute, it MUST support only those values specified in this document or through  
4298 the extension mechanism described in section 5.2.4. It MAY support any non-empty subset of these values.  
4299 That is, it MUST support at least one of the specified values and at most all of them.

### 4300 **5.2.4 Versions**

4301 IPP/1.1 clients MUST meet the conformance requirements for clients specified in this document and [IPP-  
4302 PRO]. IPP/1.1 clients MUST send requests containing a "version-number" parameter with a '1.1' value.

4303 IPP/1.1 Printer and Job objects MUST meet the conformance requirements for IPP objects specified in this  
4304 document and [IPP-PRO]. IPP/1.1 objects MUST accept requests containing a "version-number"  
4305 parameter with a '1.1' value (or reject the request if the operation is not supported).

4306 It is beyond the scope of this specification to mandate conformance with previous versions. IPP/1.1 was  
4307 deliberately designed, however, to make supporting previous versions easy. It is worth noting that, at the  
4308 time of composing this specification (1999), we would expect IPP/1.1 Printer implementations to:

4309 understand any valid request in the format of IPP/1.0, or 1.1;

4310 respond appropriately with a response containing the same "version-number" parameter value used  
4311 by the client in the request.

4312 And we would expect IPP/1.1 clients to:

4313 understand any valid response in the format of IPP/1.0, or 1.1.

4314 It is recommended that IPP/1.1 clients try supplying alternate version numbers if they receive a 'server-  
4315 error-version-not-supported' error return in a response.



### 4316 5.2.5 Extensions

4317 A conforming IPP object MAY support ~~registered-~~[IETF standards track](#) extensions and ~~private-~~[vendor](#)  
4318 extensions, as long as they [extensions](#) meet the requirements specified in Section 6.

4319 For each attribute included in an operation response, a conforming IPP object MUST return a value whose  
4320 type and value syntax conforms to the requirement of the Model document as specified in Sections 3 and 4.

### 4321 5.2.6 Attribute Syntaxes

4322 An IPP object MUST be able to accept any of the attribute syntaxes defined in Section 4.1, including their  
4323 full range, in any operation in which a client may supply attributes or the system administrator may  
4324 configure attributes (by means outside the scope of this IPP/1.1 document). In particular for each attribute  
4325 that the IPP object supports whose attribute syntax is 'text', the IPP object MUST accept and process both  
4326 the 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that the IPP object  
4327 supports whose attribute syntax is 'name', the IPP object MUST accept and process both the  
4328 'nameWithoutLanguage' and 'nameWithLanguage' forms. Furthermore, an IPP object MUST return  
4329 attributes to the client in operation responses that conform to the syntax specified in Section 4.1, including  
4330 their full range if supplied previously by a client.

### 4331 5.2.7 Security

4332 An IPP Printer implementation SHOULD contain support for Client Authentication as defined in the  
4333 IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer implementation MAY allow an  
4334 administrator to configure the Printer so that all, some, or none of the users are authenticated. See also  
4335 section 8 of this document.

4336 An IPP Printer implementation SHOULD contain support for Operation Privacy and Server Authentication  
4337 as defined in the IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer implementation MAY  
4338 allow an administrator to configure the degree of support for Operation Privacy and Server Authentication.  
4339 See also section 8 of this document.

4340 Security MUST NOT be compromised when a client supplies a lower "version-number" parameter in a  
4341 request. For example, if an IPP/1.1 conforming Printer object accepts version '1.0' requests and is  
4342 configured to enforce Digest Authentication, it MUST do the same for a version '1.0' request.

## 4343 5.3 Charset and Natural Language Requirements

4344 All clients and IPP objects MUST support the 'utf-8' charset as defined in section 4.1.7.

4345 IPP objects MUST be able to accept any client request which correctly uses the "attributes-natural-  
4346 language" operation attribute or the Natural Language Override mechanism on any individual attribute  
4347 whether or not the natural language is supported by the IPP object. If an IPP object supports a natural  
4348 language, then it MUST be able to translate (perhaps by table lookup) all generated 'text' or 'name' attribute  
4349 values into one of the supported languages (see section 3.1.4). That is, the IPP object that supports a

4350 natural language NEED NOT be a general purpose translator of any arbitrary 'text' or 'name' value supplied  
4351 by the client into that natural language. However, the object MUST be able to translate (automatically  
4352 generate) any of its own attribute values and messages into that natural language.

## 4353 **6. IANA Considerations** ~~(registered and private extensions)~~

4354 This section describes [the procedures for defining semantics for how IPP can be extended to allow the](#)  
4355 following ~~registered~~ [IETF standards track extensions](#) and ~~private~~ [vendor](#) extensions to [the IPP/1.1 Model](#)  
4356 [and Semantics document](#):

- 4357 1. keyword attribute values
- 4358 2. enum attribute values
- 4359 3. attributes
- 4360 4. attribute syntaxes
- 4361 5. operations
- 4362 6. attribute groups
- 4363 7. status codes
- 4364 [8. out-of-band attribute values](#)

4366 Extensions registered for use with IPP/1.1 are OPTIONAL for client and IPP object conformance to the  
4367 IPP/1.1 "[Model and Semantics](#)" document ([this document](#)).

4368 These extension procedures are aligned with the guidelines as set forth by the IESG [IANA-CON]. Section  
4369 11 describes how to propose new registrations for consideration. IANA will reject registration proposals  
4370 that leave out required information or do not follow the appropriate format described in Section 11. [The](#)  
4371 [IPP/1.1 Model and Semantics document](#) may also be extended by an appropriate RFC that specifies any of  
4372 the above extensions.

### 4373 **6.1 Typed 'keyword' and 'enum' Extensions**

4374 IPP allows for 'keyword' and 'enum' extensions (see sections 4.1.2.3 and 4.1.4). This document uses  
4375 prefixes to the 'keyword' and 'enum' basic attribute syntax type in order to communicate extra information  
4376 to the reader through its name. This extra information is not represented in the protocol because it is  
4377 unimportant to a client or Printer object. The list below describes the prefixes and their meaning.

4378 "type1": This IPP specification document must be revised ([or another IETF standards track document](#)  
4379 [which augments this document](#)) to add a new keyword or a new enum. No ~~private~~ [vendor defined](#)  
4380 keywords or enums are allowed.

4381 "type2": Implementers can, at any time, add new keyword or enum values by proposing the complete  
4382 specification to IANA:  
4383  
4384

4385 iana@iana.org

4386  
4387 IANA will forward the registration proposal to the IPP Designated Expert who will review the  
4388 proposal with a mailing list that the Designated Expert keeps for this purpose. Initially, that list will  
4389 be the mailing list used by the IPP WG:

4390  
4391 ipp@pwg.org

4392  
4393 even after the IPP WG is disbanded as permitted by [IANA-CON]. The IPP Designated Expert is  
4394 appointed by the IESG Area Director responsible for IPP, according to [IANA-CON].

4395  
4396 When a type2 keyword or enum is approved, the IPP Designated Expert becomes the point of  
4397 contact for any future maintenance that might be required for that registration.

4398  
4399 "type3": Implementers can, at any time, add new keyword and enum values by submitting the complete  
4400 specification to IANA as for type2 who will forward the proposal to the IPP Designated Expert.  
4401 While no additional technical review is required, the IPP Designated Expert may, at his/her  
4402 discretion, forward the proposal to the same mailing list as for type2 registrations for advice and  
4403 comment.

4404  
4405 When a type3 keyword or enum is approved by the IPP Designated Expert, the original proposer  
4406 becomes the point of contact for any future maintenance that might be required for that registration.

4407  
4408 For type2 and type3 keywords, the proposer includes the name of the keyword in the registration proposal  
4409 and the name is part of the technical review.

4410 After type2 and type3 enums specifications are approved, the IPP Designated Expert in consultation with  
4411 IANA assigns the next available enum number for each enum value.

4412 IANA will publish approved type2 and type3 keyword and enum attributes value registration specifications  
4413 in:

4414 ftp.isi.edu/iana/assignments/ipp/attribute-values/xxx/yyy.txt

4415 where xxx is the attribute name that specifies the initial values and yyy.txt is a descriptive file name that  
4416 contains one or more enums or keywords approved at the same time. For example, if several additional  
4417 enums for stapling are approved for use with the "finishings" attribute (and "finishings-default" and  
4418 "finishings-supported" attributes), IANA will publish the additional values in the file:

4419 ftp.isi.edu/iana/assignments/ipp/attribute-values/finishings/stapling.txt

4420 Note: Some attributes are defined to be: 'type3 keywords' | 'name' which allows for attribute values to be  
4421 extended by a site administrator with administrator defined names. Such names are not registered with  
4422 IANA.

4423 By definition, each of the three types above assert some sort of registry or review process in order for  
4424 extensions to be considered valid. Each higher numbered level (1, 2, 3) tends to be decreasingly less  
4425 stringent than the previous level. Therefore, any typeN value MAY be registered using a process for some  
4426 typeM where M is less than N, however such registration is NOT REQUIRED. For example, a type3 value  
4427 MAY be registered in a type 1 manner (by being included in a future version of an IPP specification),  
4428 however, it is NOT REQUIRED.

4429 This document defines keyword and enum values for all of the above types, including type3 keywords.

4430 For ~~private-(unregistered)~~vendor keyword extensions, implementers SHOULD use keywords with a  
4431 suitable distinguishing prefix, such as "xxx-" where xxx [follows the syntax rules for keywords \(see section](#)  
4432 [4.1.3\)](#) and is the (lowercase) fully qualified company name registered with IANA for use in domain names  
4433 [RFC1035]. For example, if the company XYZ Corp. had obtained the domain name "XYZ.com", then a  
4434 ~~private-~~vendor keyword 'abc' would be: 'xyz.com-abc'.

4435 Note: RFC 1035 [RFC1035] indicates that while upper and lower case letters are allowed in domain names,  
4436 no significance is attached to the case. That is, two names with the same spelling but different case are to  
4437 be treated as if identical. Also, the labels in a domain name must follow the rules for ARPANET host  
4438 names: They must start with a letter, end with a letter or digit, and have as interior characters only letters,  
4439 digits, and hyphen. Labels must be 63 characters or less. Labels are separated by the "." character.

4440 For ~~private-(unregistered)~~enum extensions, implementers MUST use values in the reserved integer  
4441 range which is  $2^{*}30$  to  $2^{*}31-1$ .

## 4442 6.2 Attribute Extensibility

4443 Attribute names ([see section 4.1.3](#)) are type2 keywords. Therefore, new attributes may be registered and  
4444 have the same status as attributes in this document by following the type2 extension rules. For ~~private~~  
4445 ~~(unregistered)-~~vendor attribute extensions, implementers SHOULD use keywords with a suitable  
4446 distinguishing prefix as described in Section 6.1.

4447 IANA will publish approved attribute registration specifications as separate files:

4448 ftp.isi.edu/iana/assignments/ipp/attributes/xxx-yyy.txt

4449 where "xxx-yyy" is the new attribute name.

4450 If a new Printer object attribute is defined and its values can be affected by a specific document format, its  
4451 specification needs to contain the following sentence:

4452 "The value of this attribute returned in a Get-Printer-Attributes response MAY depend on the  
4453 "document-format" attribute supplied (see Section 3.2.5.1)."

4454 If the specification does not, then its value in the Get-Printer-Attributes response MUST NOT depend on  
4455 the "document-format" supplied in the request. When a new Job Template attribute is registered, the value

4456 of the Printer attributes MAY vary with "document-format" supplied in the request without the  
4457 specification having to indicate so.

### 4458 **6.3 Attribute Syntax Extensibility**

4459 Attribute syntaxes ([see section 4.1](#)) are like type2 enums. Therefore, new attribute syntaxes may be  
4460 registered and have the same status as attribute syntaxes in this document by following the type2 extension  
4461 rules described in Section 6.1. The [initial set of](#) value codes that identify each of the attribute syntaxes [are](#)  
4462 [have been](#) assigned in the "Encoding and Transport" document [IPP-PRO], including a designated range for  
4463 [private, experimental use](#) [vendor extension](#).

4464 For attribute syntaxes, the IPP Designated Expert in consultation with IANA assigns the next attribute  
4465 syntax code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute  
4466 syntax registration specifications as separate files:

4467 `ftp.isi.edu/iana/assignments/ipp/attribute-syntaxes/xxx-yyy.txt`

4468 where 'xxx-yyy' is the new attribute syntax name.

### 4469 **6.4 Operation Extensibility**

4470 Operations ([see section 3](#)) may also be registered following the type2 procedures described in Section 6.1,  
4471 though major new operations will usually be done by a new standards track RFC that augments this  
4472 document. For [private \(unregistered\) vendor](#) operation extensions, implementers MUST use the range for  
4473 the "operation-id" in requests specified in Section 4.4.15 "operations-supported" Printer attribute.

4474 For operations, the IPP Designated Expert in consultation with IANA assigns the next operation-id code as  
4475 specified in Section 4.4.15. IANA will publish approved operation registration specifications as separate  
4476 files:

4477 `ftp.isi.edu/iana/assignments/ipp/operations/Xxx-Yyy.txt`

4478 where "Xxx-Yyy" is the new operation name.

### 4479 **6.5 Attribute Group Extensibilitys**

4480 Attribute groups ([see section 3.1.3](#)) passed in requests and responses may be registered following the type2  
4481 procedures described in Section 6.1. [The initial set of attribute group tags have been assigned in the](#)  
4482 ["Encoding and Transport" document](#) [The tags that identify each of the attribute groups are assigned in](#) [IPP-  
4483 PRO], [including a designated range for vendor extension](#).

4484 For attribute groups, the IPP Designated Expert in consultation with IANA assigns the next attribute group  
4485 tag code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute group  
4486 registration specifications as separate files:

4487 ftp.isi.edu/iana/assignments/ipp/attribute-group-tags/xxx-yyy-tag.txt

4488 where 'xxx-yyy-tag' is the new attribute group tag name.

## 4489 **6.6 Status Code Extensibility**

4490 Operation status codes ([see section 3.1.6.1](#)) may also be registered following the type2 procedures described  
4491 in Section 6.1. The values for status codes are allocated in ranges as specified in Section 14 for each status  
4492 code class:

4493 "informational" - Request received, continuing process

4494 "successful" - The action was successfully received, understood, and accepted

4495 "redirection" - Further action must be taken in order to complete the request

4496 "client-error" - The request contains bad syntax or cannot be fulfilled

4497 "server-error" - The IPP object failed to fulfill an apparently valid request

4498

4499 For ~~private (unregistered)~~ vendor operation status code extensions, implementers MUST use the top of each  
4500 range as specified in Section 13.

4501 For operation status codes, the IPP Designated Expert in consultation with IANA assigns the next status  
4502 code in the appropriate class range as specified in Section 13. IANA will publish approved status code  
4503 registration specifications as separate files:

4504 ftp.isi.edu/iana/assignments/ipp/status-codes/xxx-yyy.txt

4505 where "xxx-yyy" is the new operation status code keyword.

## 4506 **6.7 Out-of-band Attribute Value Extensibility**

4507 Out-of-band attribute values (see the beginning of section 4.1) passed in requests and responses may be  
4508 registered following the type2 procedures described in Section 6.1. The initial set of out-of-band attribute  
4509 value tags have been assigned in the "Encoding and Transport" document [IPP-PRO].

4510 For out-of-band attribute value tags, the IPP Designated Expert in consultation with IANA assigns the next  
4511 out-of-band attribute value tag code in the appropriate range as specified in [IPP-PRO]. IANA will publish  
4512 approved out-of-band attribute value tags registration specifications as separate files:

4513 ftp.isi.edu/iana/assignments/ipp/out-of-band-attribute-value-tags/xxx-yyy-tag.txt

4514 where 'xxx-yyy-tag' is the new out-of-band attribute value tag name.



## 4515 **6.8 Registration of MIME types/sub-types for document-formats**

4516 The "document-format" attribute's syntax is 'mimeMediaType'. This means that valid values are Internet  
4517 Media Types (see Section 4.1.9). RFC 2045 [RFC2045] defines the syntax for valid Internet media types.  
4518 IANA is the registry for all Internet media types.

## 4519 **6.9 Registration of charsets for use in 'charset' attribute values**

4520 The "attributes-charset" attribute's syntax is 'charset'. This means that valid values are charsets names.  
4521 When a charset in the IANA registry has more than one name (alias), the name labeled as "(preferred  
4522 MIME name)", if present, MUST be used (see Section 4.1.7). IANA is the registry for charsets following  
4523 the procedures of [RFC2278].

## 4524 **7. Internationalization Considerations**

4525 Some of the attributes have values that are text strings and names which are intended for human  
4526 understanding rather than machine understanding (see the 'text' and 'name' attribute syntaxes in Sections  
4527 4.1.1 and 4.1.2).

4528 In each operation request, the client

- 4529 - identifies the charset and natural language of the request which affects each supplied 'text' and 'name'  
4530 attribute value, and
  - 4531 - requests the charset and natural language for attributes returned by the IPP object in operation  
4532 responses (as described in Section 3.1.4.1).
- 4533

4534 In addition, the client MAY separately and individually identify the Natural Language Override of a  
4535 supplied 'text' or 'name' attribute using the 'textWithLanguage' and 'nameWithLanguage' technique  
4536 described section 4.1.1.2 and 4.1.2.2 respectively.

4537 All IPP objects MUST support the UTF-8 [RFC2279] charset in all 'text' and 'name' attributes supported. If  
4538 an IPP object supports more than the UTF-8 charset, the object MUST convert between them in order to  
4539 return the requested charset to the client according to Section 3.1.4.2. If an IPP object supports more than  
4540 one natural language, the object SHOULD return 'text' and 'name' values in the natural language requested  
4541 where those values are generated by the Printer (see Section 3.1.4.1).

4542 For Printers that support multiple charsets and/or multiple natural languages in 'text' and 'name' attributes,  
4543 different jobs may have been submitted in differing charsets and/or natural languages. All responses MUST  
4544 be returned in the charset requested by the client. However, the Get-Jobs operation uses the  
4545 'textWithLanguage' and 'nameWithLanguage' mechanism to identify the differing natural languages with  
4546 each job attribute returned.

4547 The Printer object also has configured charset and natural language attributes. The client can query the  
4548 Printer object to determine the list of charsets and natural languages supported by the Printer object and



4549 what the Printer object's configured values are. See the "charset-configured", "charset-supported", "natural-  
4550 language-configured", and "generated-natural-language-supported" Printer description attributes for more  
4551 details.

4552 The "charset-supported" attributed identifies the supported charsets. If a charset is supported, the IPP  
4553 object MUST be capable of converting to and from that charset into any other supported charset. In many  
4554 cases, an IPP object will support only one charset and it MUST be the UTF-8 charset.

4555 The "charset-configured" attribute identifies the one supported charset which is the native charset given the  
4556 current configuration of the IPP object (administrator defined).

4557 The "generated-natural-language-supported" attribute identifies the set of supported natural languages for  
4558 generated messages; it is not related to the set of natural languages that must be accepted for client supplied  
4559 'text' and 'name' attributes. For client supplied 'text' and 'name' attributes, an IPP object MUST accept ALL  
4560 supplied natural languages. Just because a Printer object is currently configured to support 'en-us' natural  
4561 language does not mean that the Printer object should reject a job if the client supplies a job name that is in  
4562 'fr-ca'.

4563 The "natural-language-configured" attribute identifies the one supported natural language for generated  
4564 messages which is the native natural language given the current configuration of the IPP object  
4565 (administrator defined).

4566 Attributes of type 'text' and 'name' are populated from different sources. These attributes can be categorized  
4567 into following groups (depending on the source of the attribute):

- 4568 1. Some attributes are supplied by the client (e.g., the client supplied "job-name", "document-name",  
4569 and "requesting-user-name" operation attributes along with the corresponding Job object's "job-  
4570 name" and "job-originating-user-name" attributes). The IPP object MUST accept these attributes in  
4571 any natural language no matter what the set of supported languages for generated messages
- 4572 2. Some attributes are supplied by the system administrator (e.g., the Printer object's "printer-name" and  
4573 "printer-location" attributes). These too can be in any natural language. If the natural language for  
4574 these attributes is different than what a client requests, then they must be reported using the Natural  
4575 Language Override mechanism.
- 4576 3. Some attributes are supplied by the device manufacturer (e.g., the Printer object's "printer-make-and-  
4577 model" attribute). These too can be in any natural language. If the natural language for these  
4578 attributes is different than what a client requests, then they must be reported using the Natural  
4579 Language Override mechanism.
- 4580 4. Some attributes are supplied by the operator (e.g., the Job object's "job-message-from-operator"  
4581 attribute). These too can be in any natural language. If the natural language for these attributes is  
4582 different than what a client requests, then they must be reported using the Natural Language  
4583 Override mechanism.
- 4584 5. Some attributes are generated by the IPP object (e.g., the Job object's "job-state-message" attribute,  
4585 the Printer object's "printer-state-message" attribute, and the "status-message" operation attribute).  
4586 These attributes can only be in one of the "generated-natural-language-supported" natural  
4587 languages. If a client requests some natural language for these attributes other than one of the

4588 supported values, the IPP object SHOULD respond using the value of the "natural-language-  
4589 configured" attribute (using the Natural Language Override mechanism if needed).  
4590

4591 The 'text' and 'name' attributes specified in this version of this document (additional ones will be registered  
4592 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 (text)	Job or Printer object
detailed-status-message (text)	Job or Printer object - see rule 1
document-access-error (text)	Job or Printer object - see rule 1
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
job-detailed-status-messages (1 setOf text)	Job or Printer object - see rule 1
job-document-access-errors (1 setOf text)	Job or Printer object - see rule 1
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

4593 Rule 1 - Neither the Printer nor the client localizes these message attributes, since they are intended for use  
4594 by the system administrator or other experienced technical persons.

4595

4596

## 8. Security Considerations

4597 It is difficult to anticipate the security risks that might exist in any given IPP environment. For example, if  
4598 IPP is used within a given corporation over a private network, the risks of exposing document data may be  
4599 low enough that the corporation will choose not to use encryption on that data. However, if the connection  
4600 between the client and the IPP object is over a public network, the client may wish to protect the content of  
4601 the information during transmission through the network with encryption.

4602 Furthermore, the value of the information being printed may vary from one IPP environment to the next.  
4603 Printing payroll checks, for example, would have a different value than printing public information from a  
4604 file. There is also the possibility of denial-of-service attacks, but denial-of-service attacks against printing  
4605 resources are not well understood and there is no published precedents regarding this scenario.

4606 Once the authenticated identity of the requester has been supplied to the IPP object, the object uses that  
4607 identity to enforce any authorization policy that might be in place. For example, one site's policy might be  
4608 that only the job owner is allowed to cancel a job. The details and mechanisms to set up a particular access  
4609 control policy are not part of IPP/1.1, and must be established via some other type of administrative or  
4610 access control framework. However, there are operation status codes that allow an IPP server to return  
4611 information back to a client about any potential access control violations for an IPP object.

4612 During a create operation, the client's identity is recorded in the Job object in an implementation-defined  
4613 attribute. This information can be used to verify a client's identity for subsequent operations on that Job  
4614 object in order to enforce any access control policy that might be in effect. See section 8.3 below for more  
4615 details.

4616 Since the security levels or the specific threats that any given IPP system administrator may be concerned  
4617 with cannot be anticipated, IPP MUST be capable of operating with different security mechanisms and  
4618 security policies as required by the individual installation. Security policies might vary from very strong, to  
4619 very weak, to none at all, and corresponding security mechanisms will be required.

4620

### 8.1 Security Scenarios

4621 The following sections describe specific security attacks for IPP environments. Where examples are  
4622 provided they should be considered illustrative of the environment and not an exhaustive set. Not all of  
4623 these environments will necessarily be addressed in initial implementations of IPP.

4624

#### 8.1.1 Client and Server in the Same Security Domain

4625 This environment is typical of internal networks where traditional office workers print the output of  
4626 personal productivity applications on shared work-group printers, or where batch applications print their  
4627 output on large production printers. Although the identity of the user may be trusted in this environment, a

4628 user might want to protect the content of a document against such attacks as eavesdropping, replaying or  
4629 tampering.

### 4630 **8.1.2 Client and Server in Different Security Domains**

4631 Examples of this environment include printing a document created by the client on a publicly available  
4632 printer, such as at a commercial print shop; or printing a document remotely on a business associate's  
4633 printer. This latter operation is functionally equivalent to sending the document to the business associate as  
4634 a facsimile. Printing sensitive information on a Printer in a different security domain requires strong  
4635 security measures. In this environment authentication of the printer is required as well as protection against  
4636 unauthorized use of print resources. Since the document crosses security domains, protection against  
4637 eavesdropping and document tampering are also required. It will also be important in this environment to  
4638 protect Printers against "spamming" and malicious document content.

### 4639 **8.1.3 Print by Reference**

4640 When the document is not stored on the client, printing can be done by reference. That is, the print request  
4641 can contain a reference, or pointer, to the document instead of the actual document itself (see sections 3.2.2  
4642 and 3.3.2). Standard methods currently do not exist for remote entities to "assume" the credentials of a  
4643 client for forwarding requests to a 3rd party. It is anticipated that Print-By-Reference will be used to access  
4644 "public" documents and that sophisticated methods for authenticating "proxies" is not specified in this  
4645 document.

## 4646 **8.2 URIs in Operation, Job, and Printer attributes**

4647 The "printer-uri-supported" attribute contains the Printer object's URI(s). Its companion attribute, "uri-  
4648 security-supported", identifies the security mechanism used for each URI listed in the "printer-uri-  
4649 supported" attribute. For each Printer operation request, a client MUST supply only one URI in the  
4650 "printer-uri" operation attribute. In other words, even though the Printer supports more than one URI, the  
4651 client only interacts with the Printer object using one of its URIs. This duality is not needed for Job objects,  
4652 since the Printer object is the factory for Job objects, and the Printer object will generate the correct URI  
4653 for new Job objects depending on the Printer object's security configuration.

## 4654 **8.3 URIs for each authentication mechanisms**

4655 Each URI has an authentication mechanism associated with it. If the URI is the i'th element of "printer-uri-  
4656 supported", then authentication mechanism is the "i th" element of "uri-authentication-supported". For a list  
4657 of possible authentication mechanisms, see section 4.4.2.

4658 The Printer object uses an authentication mechanism to determine the name of the user performing an  
4659 operation. This user is called the "authenticated user". The credibility of authentication depends on the  
4660 mechanism that the Printer uses to obtain the user's name. When the authentication mechanism is 'none', all  
4661 authenticated users are "anonymous".

4662 During job creation operations, the Printer initializes the value of the "job-originating-user-name" attribute  
4663 (see section 4.3.6) to be the authenticated user. The authenticated user in this case is called the "job\_  
4664 owner".

4665 If an implementation can be configured to support more than one authentication mechanism ([see section](#)  
4666 [4.4.2](#)), then it MUST implement rules for determining equality of authenticated user names which have  
4667 been authenticated via different authentication mechanisms. One possible policy is that identical names  
4668 that are authenticated via different mechanisms are different. For example, a user can cancel his job only if  
4669 he uses the same authentication mechanism for both Cancel-Job and Print-Job. Another policy is that  
4670 identical names that are authenticated via different mechanism are the same if the authentication  
4671 mechanism for the later operation is not less strong than the authentication mechanism for the earlier job  
4672 creation operation. For example, a user can cancel his job only if he uses the same or stronger  
4673 authentication mechanism for Cancel-Job and Print-Job. With this second policy a job submitted via  
4674 'requesting-user-name' authentication could be cancelled via 'digest' authentication. With the first policy, the  
4675 job could not be cancelled in this way.

4676 A client is able to determine the authentication mechanism used to create a job. It is the i'th value of the  
4677 Printer's "uri-authentication-supported" attribute (see section 4.4.2), where i is the index of the element of  
4678 the Printer's "printer-uri-supported" attribute (see section 4.4.1) equal to the job's "job-printer-uri" attribute  
4679 (see section 4.3.3).

## 4680 **8.4 Restricted Queries**

4681 In many IPP operations, a client supplies a list of attributes to be returned in the response. For security  
4682 reasons, an IPP object may be configured not to return all attributes (or all values) that a client requests.  
4683 The job attributes returned MAY depend on whether the requesting user is the same as the user that  
4684 submitted the job. The IPP object MAY even return none of the requested attributes. In such cases, the  
4685 status returned is the same as if the object had returned all requested attributes. The client cannot tell by  
4686 such a response whether the requested attribute was present or absent on the object.

## 4687 **8.5 Operations performed by operators and system administrators**

4688 For the three printer operations Pause-Printer, Resume-Printer, and Purge-Jobs (see sections 3.2.7, 3.2.8 and  
4689 3.2.9), the requesting user is intended to be an operator or administrator of the Printer object (see section 1).  
4690 [Otherwise, the IPP Printer MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-](#)  
4691 [authenticated', or 'client-error-not-authorized' as appropriate.](#) For operations on jobs, the requesting user is  
4692 intended to be the job owner or may be an operator or administrator of the Printer object. The means for  
4693 authorizing an operator or administrator of the Printer object are not specified in this document.

## 4694 **8.6 Queries on jobs submitted using non-IPP protocols**

4695 If the device that an IPP Printer is representing is able to accept jobs using other job submission protocols  
4696 in addition to IPP, it is RECOMMENDED that such an implementation at least allow such "foreign" jobs to  
4697 be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an implementation NEED

4698 NOT support all of the same IPP job attributes as for IPP jobs. The IPP object returns the 'unknown' out-of-  
4699 band value for any requested attribute of a foreign job that is supported for IPP jobs, but not for foreign  
4700 jobs.

4701 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such "foreign  
4702 jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-Attributes and  
4703 Cancel-Job. Such an implementation also needs to deal with the problem of authentication of such foreign  
4704 jobs. One approach would be to treat all such foreign jobs as belonging to users other than the user of the  
4705 IPP client. Another approach would be for the foreign job to belong to 'anonymous'. Only if the IPP client  
4706 has been authenticated as an operator or administrator of the IPP Printer object, could the foreign jobs be  
4707 queried by an IPP request. Alternatively, if the security policy is to allow users to query other users' jobs,  
4708 then the foreign jobs would also be visible to an end-user IPP client using Get-Jobs and Get-Job-Attributes.

## 4709 9. References

4710 [\[ASME-Y14.1M\]](#)

4711 [Metric Drawing Sheet Size and Format, ASME Y14.1M-1995. This standard defines metric sheet](#)  
4712 [sizes and formats for engineering drawings.](#)

4713 [ASCII]

4714 Coded Character Set - 7-bit American Standard Code for Information Interchange (ASCII), ANSI  
4715 X3.4-1986. This standard is the specification of the US-ASCII charset.

4716 [BCP-11]

4717 [Bradner S., Hovey R.](#), "The Organizations Involved in the IETF Standards Process", 1996/10/29  
4718 (RFC 2028)

4719 [HTPP]

4720 J. Barnett, K. Carter, R. DeBry, "Initial Draft - Hypertext Printing Protocol - HTPP/1.0", October  
4721 1996, <ftp://ftp.pwg.org/pub/pwg/ipp/historic/http/overview.ps.gz>

4722 [IANA-CON]

4723 Narte, T. and Alvestrand, H.T.: Guidelines for Writing an IANA Considerations Section in RFCs,  
4724 Work in Progress, draft-iesg-iana-considerations-04.txt, May 21, 1998.

4725 [IANA-CS]

4726 IANA Registry of Coded Character Sets: <ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>

4727 [IANA-MT]

4728 IANA Registry of Media Types: <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/>

4729 [IPP-IIG]

4730 Hastings, T., Manros, C., "Internet Printing Protocol/1.1: draft-ietf-ipp-implementers-guide-v11-  
4731 [00??](#).txt, [?? 1999](#), work in progress, [September 27, 1999](#).



- 4732 ~~[IPP-HG1.0]~~  
4733 ~~Hastings, T., Manros, C., "Internet Printing Protocol/1.0: Implementer's Guide", draft-ietf-ipp-~~  
4734 ~~implementers-guide-01.txt, February 1999, work in progress.~~
- 4735 [IPP-PRO]  
4736 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and  
4737 Transport", draft-ietf-ipp-protocol-v11-041.txt, February 23, 2000~~May, 1999~~.
- 4738 [ISO10646-1]  
4739 ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-Octet Coded Character Set  
4740 (UCS) - Part 1: Architecture and Basic Multilingual Plane, JTC1/SC2."
- 4741 [ISO8859-1]  
4742 ISO/IEC 8859-1:1987, "Information technology -- 8-bit One-Byte Coded Character Set - Part 1:  
4743 Latin Alphabet Nr 1", 1987, JTC1/SC2.
- 4744 [ISO10175]  
4745 ISO/IEC 10175 Document Printing Application (DPA), June 1996.
- 4746 [LDPA]  
4747 T. Hastings, S. Isaacson, M. MacKay, C. Manros, D. Taylor, P. Zehler, "LDPA - Lightweight  
4748 Document Printing Application", October 1996,  
4749 <ftp://ftp.pwg.org/pub/pwg/ipp/historic/ldpa/ldpa8.pdf.gz>
- 4750 [P1387.4]  
4751 Kirk, M. (editor), POSIX System Administration - Part 4: Printing Interfaces, POSIX 1387.4 D8,  
4752 1994.
- 4753 [PSIS] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August 1995.
- 4754 [PWG]  
4755 Printer Working Group, <http://www.pwg.org>.
- 4756 [RFC1035]  
4757 P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", RFC 1035,  
4758 November 1987.
- 4759 [RFC1179]  
4760 McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.
- 4761 [RFC1759]  
4762 Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759, March  
4763 1995.
- 4764 [RFC1766]  
4765 H. Alvestrand, "Tags for the Identification of Languages", RFC 1766, March 1995.

- 4766 [RFC1952]  
4767 P. Deutsch, "GZIP file format specification version 4.3", RFC 1952, May 1996.
- 4768 [RFC2026]  
4769 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 4770 [RFC2045]  
4771 N. Fried, N. Borenstein, ", Multipurpose Internet Mail Extensions (MIME) Part One: Format of  
4772 Internet Message Bodies " RFC 2045, November 1996.
- 4773 [RFC2046]  
4774 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed & N. Borenstein.  
4775 November 1996. (Obsoletes RFC1521, RFC1522, RFC1590), RFC 2046.
- 4776 [RFC2048]  
4777 N. Freed, J. Klensin & J. Postel, "Multipurpose Internet Mail Extension (MIME) Part Four:  
4778 Registration Procedures". RFC 2048, November 1996.
- 4779 ~~[RFC2616]~~  
4780 ~~— R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext~~  
4781 ~~Transfer Protocol - HTTP/1.1", RFC 2616, June 1999~~
- 4782 ~~[RFC2617]~~  
4783 ~~— J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, "HTTP~~  
4784 ~~Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.~~
- 4785 [RFC2119]  
4786 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March 1997
- 4787 [RFC2228]  
4788 M. Horowitz, S. Lunt, "FTP Security Extensions", RFC 2228, October 1997.
- 4789 [RFC2246]  
4790 T. Dierks, C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.
- 4791 [RFC2277]  
4792 H. Alvestrand, "IETF Policy on Character Sets and Languages" RFC 2277, January 1998.
- 4793 [RFC2278]  
4794 N. Freed, J. Postel: "IANA CharSet Registration Procedures", RFC 2278, January 1998.
- 4795 [RFC2279]  
4796 F. Yergeau , "UTF-8, a transformation format of ISO 10646", RFC 2279. January 1998.
- 4797 [RFC2316]  
4798 S. Bellovin , "Report of the IAB Security Architecture Workshop", RFC 2316, April 1998.

- 4799 [RFC2396]  
4800       Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax",  
4801       RFC 2396, August 1998.
- 4802 [RFC2565]  
4803       Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and  
4804       Transport", RFC 2565, April 1999.
- 4805 [RFC2566]  
4806       R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and  
4807       Semantics", RFC 2566, April 1999.
- 4808 [RFC2567]  
4809       Wright, D., "Design Goals for an Internet Printing Protocol", ~~draft-ietf-ipp-req-03.txt~~[RFC 2567](#),  
4810       ~~November, 1998~~[April 1999](#).
- 4811 [RFC2568]  
4812       Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",  
4813       ~~draft-ietf-ipp-rat-04.txt~~[RFC 2568](#), ~~November, 1998~~[April 1999](#).
- 4814 [RFC2569]  
4815       Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", ~~draft-~~  
4816       ~~ietf-ipp-lpd-ipp-map-05.txt~~[RFC 2569](#), ~~November 1998~~[April 1999](#).
- 4817 [RFC2579]  
4818       K. McCloghrie, D. Perkins, J. Schoenwaelder, "Textual Conventions for SMIV2" RFC 2579 (Also  
4819       STD0058), April 1999, .
- 4820 [RFC2616]  
4821       R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext  
4822       Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
- 4823 [RFC2617]  
4824       J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, "HTTP  
4825       Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.
- 4826 [SSL]  
4827       Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.
- 4828 [SWP]  
4829       P. Moore, B. Jahromi, S. Butler, "Simple Web Printing SWP/1.0", May 7, 1997,  
4830       ftp://ftp.pwg.org/pub/pwg/ipp/new\_PRO/swp9705.pdf

4831 **10. Author's Address**

4832 Scott A. Isaacson (Editor)  
4833 Novell, Inc.  
4834 122 E 1700 S  
4835 Provo, UT 84606  
4836

4837 Phone: 801-861-7366  
4838 Fax: 801-861-2517  
4839 e-mail: [sisaacson@novell.com](mailto:sisaacson@novell.com)  
4840

4841 Tom Hastings  
4842 Xerox Corporation  
4843 737 Hawaii St. ESAE 231  
4844 El Segundo, CA 90245  
4845

4846 Phone: 310-333-6413  
4847 Fax: 310-333-5514  
4848 e-mail: [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)  
4849

4850 Robert Herriot  
4851 Xerox Corp.  
4852 3400 Hill View Ave, Building 1  
4853 Palo Alto, CA 94304  
4854

4855 Phone: 650-813-7696  
4856 Fax: 650-813-6860  
4857 e-mail: [robert.herriot@pahv.xerox.com](mailto:robert.herriot@pahv.xerox.com)  
4858

4859 Roger deBry  
4860 Utah Valley State College  
4861 Orem, UT 84058  
4862

4863 Phone: (801) 222-8000  
4864 EMail: [debryro@uvsc.edu](mailto:debryro@uvsc.edu)  
4865

4866 Patrick Powell  
4867 Astart Technologies  
4868 9475 Chesapeake Dr., Suite D  
4869 San Diego, CA 95123  
4870

4871 Phone: (619) 874-6543  
4872 Fax: (619) 279-8424  
4873 e-mail: [papowell@astart.com](mailto:papowell@astart.com)  
4874

4875 IPP Mailing List: [ipp@pwg.org](mailto:ipp@pwg.org)  
4876 IPP Mailing List Subscription: [ipp-request@pwg.org](mailto:ipp-request@pwg.org)  
4877 IPP Web Page: <http://www.pwg.org/ipp/>

4878

4879 Implementers of this specification document are encouraged to join IPP Mailing List in order to participate  
4880 in any discussions of clarification issues and review of registration proposals for additional attributes and  
4881 values.

4882

4883 Other Participants:

Chuck Adams - Tektronix  
[Stefan Andersson - Axis](#)  
 Ron Bergman - ~~Data~~[Hitachi Koki Imaging Systems](#)  
 Keith Carter - IBM  
 Rajesh Chawla - TR Computing Solutions  
 Josh Cohen - Microsoft  
 Andy Davidson - Tektronix  
[Maulik Desai - Auco](#)  
 Lee Farrell - Canon [Information Systems](#)  
 Steve Gebert - IBM  
 Charles Gordon - Osicom  
 Jerry Hadsell - IBM  
 Tom Hastings - Xerox  
 Stephen Holmstead  
 Scott Isaacson - Novell  
 Swen Johnson - Xerox  
 Robert Kline - TrueSpectra  
 Carl Kugler - IBM  
 Takami Kurono - Brother  
 Scott Lawrence - Agranot Systems  
[Dwight Lewis - Lexmark](#)  
 Tony Liao - Vivid Image  
 Pete Loya - HP  
 Mike MacKay - Novell, Inc.  
 Carl-Uno Manros - Xerox  
 Stan McConnell - Xerox  
[Sandra Matts - Hewlett Packard](#)  
 Ira McDonald - High North Inc.  
 Tetsuya Morita - Ricoh  
 Pat Nogay - IBM  
[Hugo Parra, Novell](#)  
 Patrick Powell - Astart Technologies  
[Eric Random - Peerless](#)  
 Xavier Riley - Xerox  
 David Roach - Unisys  
[Yuji Sasaki - Japan Computer Industry](#)  
 Kris Schoff - HP  
 Bob Setterbo - Adobe  
[Hideki Tanaka - Cannon Information Systems](#)  
 Mike Timperman - Lexmark  
 Shigeru Ueda - Canon  
 William Wagner - ~~Osicom~~[NetSilicon/DPI](#)  
 Chris Wellens - Interworking Labs  
[Craig Whittle - Sharp Labs](#)  
 Jasper Wong - Xionics  
[Michael Wu - Heidelberg Digital](#)  
[Michael Yeung - Canon Information Systems](#)  
 Atsushi Yuki - Kyocera  
[William Zhang- Canon Information Systems](#)  
 Steve Zilles - Adobe

Shivaun Albright - HP  
 Jeff Barnett - IBM  
[Dennis Carney - IBM](#)  
 Angelo Caruso - Xerox  
[Nancy Chen - Okidata](#)  
 Jeff Copeland - QMS  
 Roger deBry - IBM  
 Mabry Dozier - QMS  
[Satoshi Fujitami - Ricoh](#)  
 Sue Gleeson - Digital  
 Brian Grimshaw - Apple  
 Richard Hart - Digital  
[Henrik Holst - I-data](#)  
 Zhi-Hong Huang - Zenographics  
 Babek Jahromi - Microsoft  
 David Kellerman - Northlake Software  
[Charles Kong - Panasonic](#)  
 Dave Kuntz - Hewlett-Packard  
 Rick Landau - Digital  
 Greg LeClair - Epson  
 Harry Lewis - IBM  
 Roy Lomicka - Digital  
 Ray Lutz - Cognisys  
 David Manchala - Xerox  
 Jay Martin - Underscore  
 Larry Masinter - Xerox  
 Peter Michalek - Shinesoft  
 Mike Moldovan - G3 Nova  
 Yuichi Niwa - Ricoh  
 Ron Norton - Printronics  
 Bob Pentecost - Hewlett-Packard  
 Jeff Rackowitz - Intermec  
 Rob Rhoads - Intel  
 Gary Roberts - Ricoh  
 Stuart Rowley - Kyocera  
 Richard Schneider - Epson  
[Katsuaki Sekiguchi - Canon Information Systems](#)  
[Gail Songer - Peerless](#)  
 Devon Taylor - Novell, Inc.  
[Atsushi Uchino - Epson](#)  
 Bob Von Anandel - Allegro Software  
 Jim Walker - DAZEL  
[Trevor Wells - Hewlett Packard](#)  
 Rob Whittle - Novell, Inc.  
 Don Wright - Lexmark  
 Rick Yardumian - Xerox  
 Lloyd Young - Lexmark  
 Peter Zehler - Xerox  
 Frank Zhao - Panasonic  
[Rob Zirnstein - Canon Information Systems](#)

## 4885 **11. Formats for IPP Registration Proposals**

4886 In order to propose an IPP extension for registration, the proposer must submit an application to IANA by  
4887 email to "iana@iana.org" or by filling out the appropriate form on the IANA web pages  
4888 (<http://www.iana.org>). This section specifies the required information and the formats for proposing  
4889 registrations of extensions to IPP as provided in Section 6 for:

4890

- 4891 1. type2 'keyword' attribute values
- 4892 2. type3 'keyword' attribute values
- 4893 3. type2 'enum' attribute values
- 4894 4. type3 'enum' attribute values
- 4895 5. attributes
- 4896 6. attribute syntaxes
- 4897 7. operations
- 4898 8. status codes
- 4899 9. out-of-band attribute values

### 4900 **11.1 Type2 keyword attribute values registration**

4901 Type of registration: type2 keyword attribute value

4902 Name of attribute to which this keyword specification is to be added:

4903 Proposed keyword name of this keyword value:

4904 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):

4905 Name of proposer:

4906 Address of proposer:

4907 Email address of proposer:

4908

4909 Note: For type2 keywords, the Designated Expert will be the point of contact for the approved registration  
4910 specification, if any maintenance of the registration specification is needed.

### 4911 **11.2 Type3 keyword attribute values registration**

4912 Type of registration: type3 keyword attribute value

4913 Name of attribute to which this keyword specification is to be added:

4914 Proposed keyword name of this keyword value:

4915 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):

4916 Name of proposer:

4917 Address of proposer:

4918 Email address of proposer:

4919

4920 Note: For type3 keywords, the proposer will be the point of contact for the approved registration  
4921 specification, if any maintenance of the registration specification is needed.



### 4922 **11.3 Type2 enum attribute values registration**

4923 Type of registration: type2 enum attribute value

4924 Name of attribute to which this enum specification is to be added:

4925 Keyword symbolic name of this enum value:

4926 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4927 Specification of this enum value (follow the style of IPP Model Section 4.1.4):

4928 Name of proposer:

4929 Address of proposer:

4930 Email address of proposer:

4931

4932 Note: For type2 enums, the Designated Expert will be the point of contact for the approved registration  
4933 specification, if any maintenance of the registration specification is needed.

### 4934 **11.4 Type3 enum attribute values registration**

4935 Type of registration: type3 enum attribute value

4936 Name of attribute to which this enum specification is to be added:

4937 Keyword symbolic name of this enum value:

4938 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4939 Specification of this enum value (follow the style of IPP Model Section 4.1.4):

4940 Name of proposer:

4941 Address of proposer:

4942 Email address of proposer:

4943

4944 Note: For type3 enums, the proposer will be the point of contact for the approved registration specification,  
4945 if any maintenance of the registration specification is needed.

### 4946 **11.5 Attribute registration**

4947 Type of registration: attribute

4948 Proposed keyword name of this attribute:

4949 Types of attribute (Operation, Job Template, Job Description, Printer Description):

4950 Operations to be used with if the attribute is an operation attribute:

4951 Object (Job, Printer, etc. if bound to an object):

4952 Attribute syntax(es) (include 1setOf and range as in Section 4.2):

4953 If attribute syntax is 'keyword' or 'enum', is it type2 or type3:

4954 If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2):

4955 If this is a Job Template attribute, how does its specification depend on the value of the "multiple-  
4956 document-handling" attribute:

4957 Specification of this attribute (follow the style of IPP Model Section 4.2):

4958 Name of proposer:

4959 Address of proposer:

4960 Email address of proposer:

4961

4962 Note: For attributes, the IPP Designated Expert will be the point of contact for the approved registration  
4963 specification, if any maintenance of the registration specification is needed.

## 4964 **11.6 Attribute Syntax registration**

4965 Type of registration: attribute syntax

4966 Proposed name of this attribute syntax:

4967 Type of attribute syntax (integer, octetString, character-string, see [IPP-PRO]):

4968 Numeric [tag according to \[IPP-PRO\]](#) ~~value~~ (to be assigned by the IPP Designated Expert in consultation  
4969 with IANA):

4970 Specification of this attribute (follow the style of IPP Model Section 4.1):

4971 Name of proposer:

4972 Address of proposer:

4973 Email address of proposer:

4974

4975 Note: For attribute syntaxes, the IPP Designated Expert will be the point of contact for the approved  
4976 registration specification, if any maintenance of the registration specification is needed.

## 4977 **11.7 Operation registration**

4978 Type of registration: operation

4979 Proposed name of this operation:

4980 Numeric operation-id value [according to section](#) 4.4.15 (to be assigned by the IPP Designated Expert in  
4981 consultation with IANA):

4982 Object Target (Job, Printer, etc. that operation is upon):

4983 Specification of this operation (follow the style of IPP Model Section 3):

4984 Name of proposer:

4985 Address of proposer:

4986 Email address of proposer:

4987

4988 Note: For operations, the IPP Designated Expert will be the point of contact for the approved registration  
4989 specification, if any maintenance of the registration specification is needed.

## 4990 **11.8 Attribute Group registration**

4991 Type of registration: attribute group

4992 Proposed name of this attribute group:

4993 Numeric tag according to [IPP-PRO] (to be assigned by the IPP Designated Expert in consultation with  
4994 IANA):

4995 Operation requests and group number for each operation in which the attribute group occurs:

4996 Operation responses and group number for each operation in which the attribute group occurs:

4997 Specification of this attribute group (follow the style of IPP Model Section 3):

4998 Name of proposer:

4999 Address of proposer:

5000 Email address of proposer:

5001  
5002 Note: For attribute groups, the IPP Designated Expert will be the point of contact for the approved  
5003 registration specification, if any maintenance of the registration specification is needed.

## 5004 **11.9 Status code registration**

5005 Type of registration: status code  
5006 Keyword symbolic name of this status code value:  
5007 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):  
5008 Operations that this status code may be used with:  
5009 Specification of this status code (follow the style of IPP Model Section 13 APPENDIX B: Status Codes  
5010 and Suggested Status Code Messages):  
5011 Name of proposer:  
5012 Address of proposer:  
5013 Email address of proposer:

5014  
5015 Note: For status codes, the Designated Expert will be the point of contact for the approved registration  
5016 specification, if any maintenance of the registration specification is needed.

## 5017 **11.10 Out-of-band Attribute Value registration**

5018 Type of registration: out-of-band attribute value  
5019 Proposed name of this out-of-band attribute value:  
5020 Numeric tag according to [IPP-PRO] (to be assigned by the IPP Designated Expert in consultation with  
5021 IANA):  
5022 Operations that this out-of-band attribute value may be used with:  
5023 Attributes that this out-of-band attribute value may be used with:  
5024 Specification of this out-of-band attribute value (follow the style of the beginning of IPP Model Section  
5025 4.1):  
5026 Name of proposer:  
5027 Address of proposer:  
5028 Email address of proposer:

5029  
5030 Note: For out-of-band attribute values, the IPP Designated Expert will be the point of contact for the  
5031 approved registration specification, if any maintenance of the registration specification is needed.

## 5032 **12. APPENDIX A: Terminology**

5033 This specification document uses the terminology defined in this section.

## 5034 **12.1 Conformance Terminology**

5035 The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT",  
5036 "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in  
5037 RFC 2119 [RFC2119].

### 5038 **12.1.1 NEED NOT**

5039 This term is not included in RFC 2119. The verb "NEED NOT" indicates an action that the subject of the  
5040 sentence does not have to implement in order to claim conformance to the standard. The verb "NEED  
5041 NOT" is used instead of "MAY NOT" since "MAY NOT" sounds like a prohibition.

## 5042 **12.2 Model Terminology**

### 5043 **12.2.1 Keyword**

5044 Keywords are used within this document as identifiers of semantic entities within the abstract model (see  
5045 section 4.1.2.3). Attribute names, some attribute values, attribute syntaxes, and attribute group names are  
5046 represented as keywords.

### 5047 **12.2.2 Attributes**

5048 An attribute is an item of information that is associated with an instance of an IPP object. An attribute  
5049 consists of an attribute name and one or more attribute values. Each attribute has a specific attribute syntax.  
5050 All object attributes are defined in section 4 and all operation attributes are defined in section 3.

5051 Job Template Attributes are described in section 4.2. The client optionally supplies Job Template attributes  
5052 in a create request (operation requests that create Job objects). The Printer object has associated attributes  
5053 which define supported and default values for the Printer.

#### 5054 **12.2.2.1 Attribute Name**

5055 Each attribute is uniquely identified in this document by its attribute name. An attribute name is a keyword.  
5056 The keyword attribute name is given in the section header describing that attribute. In running text in this  
5057 document, attribute names are indicated inside double quotation marks (") where the quotation marks are  
5058 not part of the keyword itself.

#### 5059 **12.2.2.2 Attribute Group Name**

5060 Related attributes are grouped into named groups. The name of the group is a keyword. The group name  
5061 may be used in place of naming all the attributes in the group explicitly. Attribute groups are defined in  
5062 section 3.

### 5063 12.2.2.3 Attribute Value

5064 Each attribute has one or more values. Attribute values are represented in the syntax type specified for that  
5065 attribute. In running text in this document, attribute values are indicated inside single quotation marks ('),  
5066 whether their attribute syntax is keyword, integer, text, etc. where the quotation marks are not part of the  
5067 value itself.

### 5068 12.2.2.4 Attribute Syntax

5069 Each attribute is defined using an explicit syntax type. In this document, each syntax type is defined as a  
5070 keyword with specific meaning. The "Encoding and Transport" document [IPP-PRO] indicates the actual  
5071 "on-the-wire" encoding rules for each syntax type. Attribute syntax types are defined in section 4.1.

### 5072 12.2.3 Supports

5073 By definition, a Printer object supports an attribute only if that Printer object responds with the  
5074 corresponding attribute populated with some value(s) in a response to a query for that attribute. A Printer  
5075 object supports an attribute value if the value is one of the Printer object's "supported values" attributes.  
5076 The device behind a Printer object may exhibit a behavior that corresponds to some IPP attribute, but if the  
5077 Printer object, when queried for that attribute, doesn't respond with the attribute, then as far as IPP is  
5078 concerned, that implementation does not support that feature. If the Printer object's "xxx-supported"  
5079 attribute is not populated with a particular value (even if that value is a legal value for that attribute), then  
5080 that Printer object does not support that particular value.

5081 A conforming implementation MUST support all REQUIRED attributes. However, even for REQUIRED  
5082 attributes, conformance to IPP does not mandate that all implementations support all possible values  
5083 representing all possible job processing behaviors and features. For example, if a given instance of a  
5084 Printer supports only certain document formats, then that Printer responds with the "document-format-  
5085 supported" attribute populated with a set of values, possibly only one, taken from the entire set of possible  
5086 values defined for that attribute. This limited set of values represents the Printer's set of supported  
5087 document formats. Supporting an attribute and some set of values for that attribute enables IPP end users to  
5088 be aware of and make use of those features associated with that attribute and those values. If an  
5089 implementation chooses to not support an attribute or some specific value, then IPP end users would have  
5090 no ability to make use of that feature within the context of IPP itself. However, due to existing practice and  
5091 legacy systems which are not IPP aware, there might be some other mechanism outside the scope of IPP to  
5092 control or request the "unsupported" feature (such as embedded instructions within the document data  
5093 itself).

5094 For example, consider the "finishings-supported" attribute.

- 5095 1) If a Printer object is not physically capable of stapling, the "finishings-supported" attribute MUST  
5096 NOT be populated with the value of 'staple'.
- 5097 2) A Printer object is physically capable of stapling, however an implementation chooses not to support  
5098 stapling in the IPP "finishings" attribute. In this case, 'staple' MUST NOT be a value in the  
5099 "finishings-supported" Printer object attribute. Without support for the value 'staple', an IPP end  
5100 user would have no means within the protocol itself to request that a Job be stapled. However, an

5101 existing document data formatter might be able to request that the document be stapled directly with  
5102 an embedded instruction within the document data. In this case, the IPP implementation does not  
5103 "support" stapling, however the end user is still able to have some control over the stapling of the  
5104 completed job.

- 5105 3) A Printer object is physically capable of stapling, and an implementation chooses to support stapling  
5106 in the IPP "finishings" attribute. In this case, 'staple' MUST be a value in the "finishings-supported"  
5107 Printer object attribute. Doing so, would enable end users to be aware of and make use of the  
5108 stapling feature using IPP attributes.  
5109

5110 Even though support for Job Template attributes by a Printer object is OPTIONAL, it is RECOMMENDED  
5111 that if the device behind a Printer object is capable of realizing any feature or function that corresponds to  
5112 an IPP attribute and some associated value, then that implementation SHOULD support that IPP attribute  
5113 and value.

5114 The set of values in any of the supported value attributes is set (populated) by some administrative process  
5115 or automatic sensing mechanism that is outside the scope of this IPP/1.1 document. For administrative  
5116 policy and control reasons, an administrator may choose to make only a subset of possible values visible to  
5117 the end user. In this case, the real output device behind the IPP Printer abstraction may be capable of a  
5118 certain feature, however an administrator is specifying that access to that feature not be exposed to the end  
5119 user through the IPP protocol. Also, since a Printer object may represent a logical print device (not just a  
5120 physical device) the actual process for supporting a value is undefined and left up to the implementation.  
5121 However, if a Printer object supports a value, some manual human action may be needed to realize the  
5122 semantic action associated with the value, but no end user action is required.

5123 For example, if one of the values in the "finishings-supported" attribute is 'staple', the actual process might  
5124 be an automatic staple action by a physical device controlled by some command sent to the device. Or, the  
5125 actual process of stapling might be a manual action by an operator at an operator attended Printer object.

5126 For another example of how supported attributes function, consider a system administrator who desires to  
5127 control all print jobs so that no job sheets are printed in order to conserve paper. To force no job sheets, the  
5128 system administrator sets the only supported value for the "job-sheets-supported" attribute to 'none'. In this  
5129 case, if a client requests anything except 'none', the create request is rejected or the "job-sheets" value is  
5130 ignored (depending on the value of "ipp-attribute-fidelity"). To force the use of job start/end sheets on all  
5131 jobs, the administrator does not include the value 'none' in the "job-sheets-supported" attribute. In this case,  
5132 if a client requests 'none', the create request is rejected or the "job-sheets" value is ignored (again depending  
5133 on the value of "ipp-attribute-fidelity").

#### 5134 **12.2.4 print-stream page**

5135 A "print-stream page" is a page according to the definition of pages in the language used to express the  
5136 document data.

5137 **12.2.5 impression**

5138 An "impression" is the image (possibly many print-stream pages in different configurations) imposed onto a  
5139 single media page.

5140 **13. APPENDIX B: Status Codes and Suggested Status Code Messages**

5141 This section defines status code enum keywords and values that are used to provide semantic information  
5142 on the results of an operation request. Each operation response **MUST** include a status code. The response  
5143 **MAY** also contain a status message that provides a short textual description of the status. The status code  
5144 is intended for use by automata, and the status message is intended for the human end user. Since the status  
5145 message is an **OPTIONAL** component of the operation response, an IPP application (i.e., a browser, GUI,  
5146 print driver or gateway) is **NOT REQUIRED** to examine or display the status message, since it **MAY** not be  
5147 returned to the application.

5148 The prefix of the status keyword defines the class of response as follows:

- 5149 "informational" - Request received, continuing process
  - 5150 "successful" - The action was successfully received, understood, and accepted
  - 5151 "redirection" - Further action must be taken in order to complete the request
  - 5152 "client-error" - The request contains bad syntax or cannot be fulfilled
  - 5153 "server-error" - The IPP object failed to fulfill an apparently valid request
- 5154

5155 As with type2 enums, IPP status codes are extensible. IPP clients are **NOT REQUIRED** to understand the  
5156 meaning of all registered status codes, though such understanding is obviously desirable. However, IPP  
5157 clients **MUST** understand the class of any status code, as indicated by the prefix, and treat any unrecognized  
5158 response as being equivalent to the first status code of that class, with the exception that an unrecognized  
5159 response **MUST NOT** be cached. For example, if an unrecognized status code of "client-error-xxx-yyy" is  
5160 received by the client, it can safely assume that there was something wrong with its request and treat the  
5161 response as if it had received a "client-error-bad-request" status code. In such cases, IPP applications  
5162 **SHOULD** present the **OPTIONAL** message (if present) to the end user since the message is likely to  
5163 contain human readable information which will help to explain the unusual status. The name of the enum  
5164 is the suggested status message for US English.

5165 The status code values range from 0x0000 to 0x7FFF. The value ranges for each status code class are as  
5166 follows:

- 5167 "successful" - 0x0000 to 0x00FF
  - 5168 "informational" - 0x0100 to 0x01FF
  - 5169 "redirection" - 0x0200 to 0x02FF
  - 5170 "client-error" - 0x0400 to 0x04FF
  - 5171 "server-error" - 0x0500 to 0x05FF
- 5172



5173 The top half (128 values) of each range (0x0n40 to 0x0nFF, for n = 0 to 5) is reserved for [private-vendor](#)  
5174 use within each status code class. Values 0x0600 to 0x7FFF are reserved for future assignment [by IETF](#)  
5175 [standards track documents](#) and MUST NOT be used.

## 5176 **13.1 Status Codes**

5177 Each status code is described below. Section 13.1.5.9 contains a table that indicates which status codes  
5178 apply to which operations. The Implementer's Guide [IPP-IIG] describe the suggested steps for processing  
5179 IPP attributes for all operations, including returning status codes.

### 5180 **13.1.1 Informational**

5181 This class of status code indicates a provisional response and is to be used for informational purposes only.

5182 There are no status codes defined in IPP/1.1 for this class of status code.

### 5183 **13.1.2 Successful Status Codes**

5184 This class of status code indicates that the client's request was successfully received, understood, and  
5185 accepted.

#### 5186 **13.1.2.1 successful-ok (0x0000)**

5187 The request has succeeded and no request attributes were substituted or ignored. In the case of a response  
5188 to a create request, the 'successful-ok' status code indicates that the request was successfully received and  
5189 validated, and that the Job object has been created; it does not indicate that the job has been processed. The  
5190 transition of the Job object into the 'completed' state is the only indicator that the job has been printed.

#### 5191 **13.1.2.2 successful-ok-ignored-or-substituted-attributes (0x0001)**

5192 The request has succeeded, but some supplied (1) attributes were ignored or (2) unsupported values were  
5193 substituted with supported values or were ignored in order to perform the operation without rejecting it.  
5194 Unsupported attributes, attribute syntaxes, or values MUST be returned in the Unsupported Attributes  
5195 group of the response for all operations. There is an exception to this rule for the query operations: Get-  
5196 Printer-Attributes, Get-Jobs, and Get-Job-Attributes for the "requested-attributes" operation attribute only.  
5197 When the supplied values of the "requested-attributes" operation attribute are requesting attributes that are  
5198 not supported, the IPP object MAY, but is NOT REQUIRED to, return the "requested-attributes" attribute  
5199 in the Unsupported Attribute response group (with the unsupported values only). See sections 3.1.7 and  
5200 3.2.1.2.

#### 5201 **13.1.2.3 successful-ok-conflicting-attributes (0x0002)**

5202 The request has succeeded, but some supplied attribute values conflicted with the values of other supplied  
5203 attributes. These conflicting values were either (1) substituted with (supported) values or (2) the attributes

5204 were removed in order to process the job without rejecting it. Attributes or values which conflict with other  
5205 attributes and have been substituted or ignored MUST be returned in the Unsupported Attributes group of  
5206 the response for all operations as supplied by the client. See sections 3.1.7 and 3.2.1.2.

### 5207 **13.1.3 Redirection Status Codes**

5208 This class of status code indicates that further action needs to be taken to fulfill the request.

5209 There are no status codes defined in IPP/1.1 for this class of status code.

### 5210 **13.1.4 Client Error Status Codes**

5211 This class of status code is intended for cases in which the client seems to have erred. The IPP object  
5212 SHOULD return a message containing an explanation of the error situation and whether it is a temporary or  
5213 permanent condition.

#### 5214 **13.1.4.1 client-error-bad-request (0x0400)**

5215 The request could not be understood by the IPP object due to malformed syntax (such as the value of a  
5216 fixed length attribute whose length does not match the prescribed length for that attribute - see the  
5217 Implementer's Guide [IPP-IIG] ). The IPP application SHOULD NOT repeat the request without  
5218 modifications.

#### 5219 **13.1.4.2 client-error-forbidden (0x0401)**

5220 The IPP object understood the request, but is refusing to fulfill it. Additional authentication information or  
5221 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is  
5222 commonly used when the IPP object does not wish to reveal exactly why the request has been refused or  
5223 when no other response is applicable.

#### 5224 **13.1.4.3 client-error-not-authenticated (0x0402)**

5225 The request requires user authentication. The IPP client may repeat the request with suitable authentication  
5226 information. If the request already included authentication information, then this status code indicates that  
5227 authorization has been refused for those credentials. If this response contains the same challenge as the  
5228 prior response, and the user agent has already attempted authentication at least once, then the response  
5229 message may contain relevant diagnostic information. This status codes reveals more information than  
5230 "client-error-forbidden".

#### 5231 **13.1.4.4 client-error-not-authorized (0x0403)**

5232 The requester is not authorized to perform the request. Additional authentication information or  
5233 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is used  
5234 when the IPP object wishes to reveal that the authentication information is understandable, however, the

5235 requester is explicitly not authorized to perform the request. This status codes reveals more information  
5236 than "client-error-forbidden" and "client-error-not-authenticated".

#### 5237 **13.1.4.5 client-error-not-possible (0x0404)**

5238 This status code is used when the request is for something that can not happen. For example, there might  
5239 be a request to cancel a job that has already been canceled or aborted by the system. The IPP client  
5240 SHOULD NOT repeat the request.

#### 5241 **13.1.4.6 client-error-timeout (0x0405)**

5242 The client did not produce a request within the time that the IPP object was prepared to wait. For example,  
5243 a client issued a Create-Job operation and then, after a long period of time, issued a Send-Document  
5244 operation and this error status code was returned in response to the Send-Document request (see section  
5245 3.3.1). The IPP object might have been forced to clean up resources that had been held for the waiting  
5246 additional Documents. The IPP object was forced to close the Job since the client took too long. The client  
5247 SHOULD NOT repeat the request without modifications.

#### 5248 **13.1.4.7 client-error-not-found (0x0406)**

5249 The IPP object has not found anything matching the request URI. No indication is given of whether the  
5250 condition is temporary or permanent. For example, a client with an old reference to a Job (a URI) tries to  
5251 cancel the Job, however in the mean time the Job might have been completed and all record of it at the  
5252 Printer has been deleted. This status code, 'client-error-not-found' is returned indicating that the referenced  
5253 Job can not be found. This error status code is also used when a client supplies a URI as a reference to the  
5254 document data in either a Print-URI or Send-URI operation, but the document can not be found.

5255 In practice, an IPP application should avoid a not found situation by first querying and presenting a list of  
5256 valid Printer URIs and Job URIs to the end-user.

#### 5257 **13.1.4.8 client-error-gone (0x0407)**

5258 The requested object is no longer available and no forwarding address is known. This condition should be  
5259 considered permanent. Clients with link editing capabilities should delete references to the request URI  
5260 after user approval. If the IPP object does not know or has no facility to determine, whether or not the  
5261 condition is permanent, the status code "client-error-not-found" should be used instead.

5262 This response is primarily intended to assist the task of maintenance by notifying the recipient that the  
5263 resource is intentionally unavailable and that the IPP object administrator desires that remote links to that  
5264 resource be removed. It is not necessary to mark all permanently unavailable resources as "gone" or to keep  
5265 the mark for any length of time -- that is left to the discretion of the IPP object administrator.

**5266 13.1.4.9 client-error-request-entity-too-large (0x0408)**

5267 The IPP object is refusing to process a request because the request entity is larger than the IPP object is  
5268 willing or able to process. An IPP Printer returns this status code when it limits the size of print jobs and it  
5269 receives a print job that exceeds that limit or when the attributes are so many that their encoding causes the  
5270 request entity to exceed IPP object capacity.

**5271 13.1.4.10 client-error-request-value-too-long (0x0409)**

5272 The IPP object is refusing to service the request because one or more of the client-supplied attributes has a  
5273 variable length value that is longer than the maximum length specified for that attribute. The IPP object  
5274 might not have sufficient resources (memory, buffers, etc.) to process (even temporarily), interpret, and/or  
5275 ignore a value larger than the maximum length. Another use of this error code is when the IPP object  
5276 supports the processing of a large value that is less than the maximum length, but during the processing of  
5277 the request as a whole, the object may pass the value onto some other system component which is not able  
5278 to accept the large value. For more details, see the Implementer's Guide [IPP-IIG] .

5279 Note: For attribute values that are URIs, this rare condition is only likely to occur when a client has  
5280 improperly submitted a request with long query information (e.g. an IPP application allows an end-user to  
5281 enter an invalid URI), when the client has descended into a URI "black hole" of redirection (e.g., a  
5282 redirected URI prefix that points to a suffix of itself), or when the IPP object is under attack by a client  
5283 attempting to exploit security holes present in some IPP objects using fixed-length buffers for reading or  
5284 manipulating the Request-URI.

**5285 13.1.4.11 client-error-document-format-not-supported (0x040A)**

5286 The IPP object is refusing to service the request because the document data is in a format, as specified in  
5287 the "document-format" operation attribute, that is not supported by the Printer object. This error is returned  
5288 independent of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return this status code,  
5289 even if there are other Job Template attributes that are not supported as well, since this error is a bigger  
5290 problem than with Job Template attributes. See sections 3.1.6.1, 3.1.7, and 3.2.1.1.

**5291 13.1.4.12 client-error-attributes-or-values-not-supported (0x040B)**

5292 In a create request, if the Printer object does not support one or more attributes, attribute syntaxes, or  
5293 attribute values supplied in the request and the client supplied the "ipp-attribute-fidelity" operation  
5294 attribute with the 'true' value, the Printer object MUST return this status code. The Printer object MUST  
5295 also return in the Unsupported Attributes Group all the attributes and/or values supplied by the client that  
5296 are not supported. See section 3.1.7. For example, if the request indicates 'iso-a4' media, but that media  
5297 type is not supported by the Printer object. Or, if the client supplies a Job Template attribute and the  
5298 attribute itself is not even supported by the Printer. If the "ipp-attribute-fidelity" attribute is 'false', the  
5299 Printer MUST ignore or substitute values for unsupported Job Template attributes and values rather than  
5300 reject the request and return this status code.

5301 For any operation where a client requests attributes (such as a Get-Jobs, Get-Printer-Attributes, or Get-Job-  
5302 Attributes operation), if the IPP object does not support one or more of the requested attributes, the IPP  
5303 object simply ignores the unsupported requested attributes and processes the request as if they had not been  
5304 supplied, rather than returning this status code. In this case, the IPP object MUST return the 'successful-ok-  
5305 ignored-or-substituted-attributes' status code and MAY return the unsupported attributes as values of the  
5306 "requested-attributes" in the Unsupported Attributes Group (see section 13.1.2.2).

#### 5307 **13.1.4.13 client-error-uri-scheme-not-supported (0x040C)**

5308 The scheme of the client-supplied URI in a Print-URI or a Send-URI operation is not supported. See  
5309 sections [3.1.6.1](#) and [3.1.7](#).

#### 5310 **13.1.4.14 client-error-charset-not-supported (0x040D)**

5311 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-  
5312 charset" operation attribute, the Printer MUST reject the operation and return this status and any 'text' or  
5313 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1). See sections [3.1.6.1](#) and [3.1.7](#).

#### 5314 **13.1.4.15 client-error-conflicting-attributes (0x040E)**

5315 The request is rejected because some attribute values conflicted with the values of other attributes which  
5316 this document does not permit to be substituted or ignored. The Printer object MUST also return in the  
5317 Unsupported Attributes Group the conflicting attributes supplied by the client. See sections 3.1.7 and  
5318 3.2.1.2.

#### 5319 **13.1.4.16 client-error-compression-not-supported (0x040F)**

5320 The IPP object is refusing to service the request because the document data, as specified in the  
5321 "compression" operation attribute, is compressed in a way that is not supported by the Printer object. This  
5322 error is returned independent of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return  
5323 this status code, even if there are other Job Template attributes that are not supported as well, since this  
5324 error is a bigger problem than with Job Template attributes. See sections [3.1.6.1](#), [3.1.7](#), and [3.2.1.1](#).

#### 5325 **13.1.4.17 client-error-compression-error (0x0410)**

5326 The IPP object is refusing to service the request because the document data cannot be decompressed when  
5327 using the algorithm specified by the "compression" operation attribute. This error is returned independent  
5328 of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return this status code, even if there  
5329 are Job Template attributes that are not supported as well, since this error is a bigger problem than with Job  
5330 Template attributes. See sections 3.1.7 and 3.2.1.1.

**5331 13.1.4.18 client-error-document-format-error (0x0411)**

5332 The IPP object is refusing to service the request because Printer encountered an error in the document data  
5333 while interpreting it. This error is returned independent of the client-supplied "ipp-attribute-fidelity". The  
5334 Printer object **MUST** return this status code, even if there are Job Template attributes that are not supported  
5335 as well, since this error is a bigger problem than with Job Template attributes. See sections 3.1.7 and  
5336 3.2.1.1.

**5337 13.1.4.19 client-error-document-access-error (0x0412)**

5338 The IPP object is refusing to service the Print-URI or Send-URI request because Printer encountered an  
5339 access error while attempting to validate the accessibility or access the document data specified in the  
5340 "document-uri" operation attribute. The Printer **MAY** also return a specific document access error code  
5341 using the "document-access-error" operation attribute (see section 3.1.6.4). This error is returned  
5342 independent of the client-supplied "ipp-attribute-fidelity". The Printer object **MUST** return this status code,  
5343 even if there are Job Template attributes that are not supported as well, since this error is a bigger problem  
5344 than with Job Template attributes. See sections [3.1.6.1](#) **and** [3.1.7](#).

**5345 13.1.5 Server Error Status Codes**

5346 This class of status codes indicates cases in which the IPP object is aware that it has erred or is incapable of  
5347 performing the request. The IPP object **SHOULD** include a message containing an explanation of the error  
5348 situation, and whether it is a temporary or permanent condition.

**5349 13.1.5.1 server-error-internal-error (0x0500)**

5350 The IPP object encountered an unexpected condition that prevented it from fulfilling the request. This error  
5351 status code differs from "server-error-temporary-error" in that it implies a more permanent type of internal  
5352 error. It also differs from "server-error-device-error" in that it implies an unexpected condition (unlike a  
5353 paper-jam or out-of-toner problem which is undesirable but expected). This error status code indicates that  
5354 probably some knowledgeable human intervention is required.

**5355 13.1.5.2 server-error-operation-not-supported (0x0501)**

5356 The IPP object does not support the functionality required to fulfill the request. This is the appropriate  
5357 response when the IPP object does not recognize an operation or is not capable of supporting it. See  
5358 sections [3.1.6.1](#) **and** [3.1.7](#).

**5359 13.1.5.3 server-error-service-unavailable (0x0502)**

5360 The IPP object is currently unable to handle the request due to a temporary overloading or maintenance of  
5361 the IPP object. The implication is that this is a temporary condition which will be alleviated after some  
5362 delay. If known, the length of the delay may be indicated in the message. If no delay is given, the IPP  
5363 application should handle the response as it would for a "server-error-temporary-error" response. If the



5364 condition is more permanent, the error status codes "client-error-gone" or "client-error-not-found" could be  
5365 used.

#### 5366 **13.1.5.4 server-error-version-not-supported (0x0503)**

5367 The IPP object does not support, or refuses to support, the IPP protocol version that was supplied as the  
5368 value of the "version-number" operation parameter in the request. The IPP object is indicating that it is  
5369 unable or unwilling to complete the request using the same major and minor version number as supplied in  
5370 the request other than with this error message. The error response SHOULD contain a "status-message"  
5371 attribute (see section 3.1.6.2) describing why that version is not supported and what other versions are  
5372 supported by that IPP object. See sections [3.1.6.1](#), [3.1.7](#), and [3.1.8](#).

5373 The error response MUST identify in the "version-number" operation parameter the closest version number  
5374 that the IPP object does support. For example, if a client supplies version '1.0' and an IPP/1.1 object  
5375 supports version '1.0', then it responds with version '1.0' in all responses to such a request. If the IPP/1.1  
5376 object does not support version '1.0', then it should accept the request and respond with version '1.1' or may  
5377 reject the request and respond with this error code and version '1.1'. If a client supplies a version '1.2', the  
5378 IPP/1.1 object should accept the request and return version '1.1' or may reject the request and respond with  
5379 this error code and version '1.1'. See sections 3.1.8 and 4.4.14.

#### 5380 **13.1.5.5 server-error-device-error (0x0504)**

5381 A printer error, such as a paper jam, occurs while the IPP object processes a Print or Send operation. The  
5382 response contains the true Job Status (the values of the "job-state" and "job-state-reasons" attributes).  
5383 Additional information can be returned in the OPTIONAL "job-state-message" attribute value or in the  
5384 OPTIONAL status message that describes the error in more detail. This error status code is only returned in  
5385 situations where the Printer is unable to accept the create request because of such a device error. For  
5386 example, if the Printer is unable to spool, and can only accept one job at a time, the reason it might reject a  
5387 create request is that the printer currently has a paper jam. In many cases however, where the Printer object  
5388 can accept the request even though the Printer has some error condition, the 'successful-ok' status code will  
5389 be returned. In such a case, the client would look at the returned Job Object Attributes or later query the  
5390 Printer to determine its state and state reasons.

#### 5391 **13.1.5.6 server-error-temporary-error (0x0505)**

5392 A temporary error such as a buffer full write error, a memory overflow (i.e. the document data exceeds the  
5393 memory of the Printer), or a disk full condition, occurs while the IPP Printer processes an operation. The  
5394 client MAY try the unmodified request again at some later point in time with an expectation that the  
5395 temporary internal error condition may have been cleared. Alternatively, as an implementation option, a  
5396 Printer object MAY delay the response until the temporary condition is cleared so that no error is returned.



**5397 13.1.5.7 server-error-not-accepting-jobs (0x0506)**

5398 A temporary error indicating that the Printer is not currently accepting jobs, because the administrator has  
5399 set the value of the Printer's "printer-is-~~not~~-accepting-jobs" attribute to 'false' (by means outside the scope of  
5400 this IPP/1.1 document).

**5401 13.1.5.8 server-error-busy (0x0507)**

5402 A temporary error indicating that the Printer is too busy processing jobs and/or other requests. The client  
5403 SHOULD try the unmodified request again at some later point in time with an expectation that the  
5404 temporary busy condition will have been cleared.

**5405 13.1.5.9 server-error-job-canceled (0x0508)**

5406 An error indicating that the job has been canceled by an operator or the system while the client was  
5407 transmitting the data to the IPP Printer. If a job-id and job-uri had been created, then they are returned in  
5408 the Print-Job, Send-Document, or Send-URI response as usual; otherwise, no job-id and job-uri are returned  
5409 in the response.

**5410 13.1.5.10 server-error-multiple-document-jobs-not-supported (0x0509)**

5411 The IPP object does not support multiple documents per job and a client attempted to supply document data  
5412 with a second Send-Document or Send-URI operation.

5413 **13.2 Status Codes for IPP Operations**

5414 PJ = Print-Job, PU = Print-URI, CJ = Create-Job, SD = Send-Document  
 5415 SU = Send-URI, V = Validate-Job, GA = Get-Job-Attributes and  
 5416 Get-Printer-Attributes, GJ = Get-Jobs, C = Cancel-Job

5417

5418

5419 IPP Status Keyword

IPP Operations

	PJ	PU	CJ	SD	SU	V	GA	GJ	C
--	----	----	----	----	----	---	----	----	---

5420 -----

	--	--	--	--	--	--	--	--	--
--	----	----	----	----	----	----	----	----	----

5421 successful-ok

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5422 successful-ok-ignored-or-substituted-  
5423 attributes

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5424 successful-ok-conflicting-attributes

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5425 client-error-bad-request

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5426 client-error-forbidden

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5427 client-error-not-authenticated

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5428 client-error-not-authorized

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5429 client-error-not-possible

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5430 client-error-timeout

				x	x				
--	--	--	--	---	---	--	--	--	--

5431 client-error-not-found

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5432 client-error-gone

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5433 client-error-request-entity-too-large

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5434 client-error-request-value-too-long

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5435 client-error-document-format-not-

	x	x		x	x	x	x		
--	---	---	--	---	---	---	---	--	--

5436 supported

5437 client-error-attributes-or-values-not-

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5438 supported

5439 client-error-uri-scheme-not-supported

		x			x				
--	--	---	--	--	---	--	--	--	--

5440 client-error-charset-not-supported

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5441 client-error-conflicting-attributes

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5442 client-error-compression-not-supported

	x	x		x	x	x			
--	---	---	--	---	---	---	--	--	--

5443 client-error-compression-error

	x	x		x	x				
--	---	---	--	---	---	--	--	--	--

5444 client-error-document-format-error

	x	x		x	x				
--	---	---	--	---	---	--	--	--	--

5445 client-error-document-access-error

		x			x				
--	--	---	--	--	---	--	--	--	--

5446 server-error-internal-error

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5447 server-error-operation-not-supported

		x	x	x	x				
--	--	---	---	---	---	--	--	--	--

5448 server-error-service-unavailable

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5449 server-error-version-not-supported

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5450 server-error-device-error

	x	x	x	x	x				
--	---	---	---	---	---	--	--	--	--

5451 server-error-temporary-error

	x	x	x	x	x				
--	---	---	---	---	---	--	--	--	--

5452 server-error-not-accepting-jobs

	x	x	x			x			
--	---	---	---	--	--	---	--	--	--

5453 server-error-busy

	x	x	x	x	x	x	x	x	x
--	---	---	---	---	---	---	---	---	---

5454 server-error-job-canceled

	x			x	x				
--	---	--	--	---	---	--	--	--	--

5455 server-error-multiple-document-jobs-

				x	x				
--	--	--	--	---	---	--	--	--	--

5456 not-supported

5457 HJ = Hold-Job, RJ = Release-Job, RS = Restart-Job  
 5458 PP = Pause-Printer, RP = Resume-Printer, PJ = Purge-Jobs

5459		IPP Operations (cont.)					
5460	IPP Status Keyword	HJ	RJ	RS	PP	RP	PJ
5461	-----	--	--	--	--	--	--
5462	successful-ok	x	x	x	x	x	x
5463	successful-ok-ignored-or-substituted-	x	x	x	x	x	x
5464	attributes						
5465	successful-ok-conflicting-attributes	x	x	x	x	x	x
5466	client-error-bad-request	x	x	x	x	x	x
5467	client-error-forbidden	x	x	x	x	x	x
5468	client-error-not-authenticated	x	x	x	x	x	x
5469	client-error-not-authorized	x	x	x	x	x	x
5470	client-error-not-possible	x	x	x	x	x	x
5471	client-error-timeout						
5472	client-error-not-found	x	x	x	x	x	x
5473	client-error-gone	x	x	x	x	x	x
5474	client-error-request-entity-too-large	x	x	x	x	x	x
5475	client-error-request-value-too-long	x	x	x	x	x	x
5476	client-error-document-format-not-						
5477	supported						
5478	client-error-attributes-or-values-not-	x	x	x	x	x	x
5479	supported						
5480	client-error-uri-scheme-not-supported						
5481	client-error-charset-not-supported	x	x	x	x	x	x
5482	client-error-conflicting-attributes	x	x	x	x	x	x
5483	client-error-compression-not-supported						
5484	client-error-compression-error						
5485	client-error-document-format-error						
5486	client-error-document-access-error						
5487	server-error-internal-error	x	x	x	x	x	x
5488	server-error-operation-not-supported	x	x	x	x	x	x
5489	server-error-service-unavailable	x	x	x	x	x	x
5490	server-error-version-not-supported	x	x	x	x	x	x
5491	server-error-device-error						
5492	server-error-temporary-error	x	x	x	x	x	x
5493	server-error-not-accepting-jobs						
5494	server-error-busy	x	x	x	x	x	x
5495	server-error-job-canceled						
5496	server-error-multiple-document-jobs-						
5497	not-supported						
5498							
5499							

5500

5501 **14. APPENDIX C: "media" keyword values**5502 **14. APPENDIX C: "media" keyword values**

5503 Standard keyword values are taken from several sources.

5504 Standard values are defined (taken from DPA[ISO10175] and the Printer MIB[RFC1759]):

5505 'default': The default medium for the output device

5506 'iso-a4-white': Specifies the ISO A4 white medium: [210 mm x 297 mm](#)5507 'iso-a4-colored': Specifies the ISO A4 colored medium: [210 mm x 297 mm](#)5508 'iso-a4-transparent': Specifies the ISO A4 transparent medium: [210 mm x 297 mm](#)5509 'iso-a3-white': Specifies the ISO A3 white medium: [297 mm x 420 mm](#)5510 'iso-a3-colored': Specifies the ISO A3 colored medium: [297 mm x 420 mm](#)5511 'iso-a5-white': Specifies the ISO A5 white medium: [148 mm x 210 mm](#)5512 'iso-a5-colored': Specifies the ISO A5 colored medium: [148 mm x 210 mm](#)5513 'iso-b4-white': Specifies the ISO B4 white medium: [250 mm x 353 mm](#)5514 'iso-b4-colored': Specifies the ISO B4 colored medium: [250 mm x 353 mm](#)5515 'iso-b5-white': Specifies the ISO B5 white medium: [176 mm x 250 mm](#)5516 'iso-b5-colored': Specifies the ISO B5 colored medium: [176 mm x 250 mm](#)5517 'jis-b4-white': Specifies the JIS B4 white medium: [257 mm x 364 mm](#)5518 'jis-b4-colored': Specifies the JIS B4 colored medium: [257 mm x 364 mm](#)5519 'jis-b5-white': Specifies the JIS B5 white medium: [182 mm x 257 mm](#)5520 'jis-b5-colored': Specifies the JIS B5 colored medium: [182 mm x 257 mm](#)

5521

5522 The following standard values are defined for North American media:

5523 'na-letter-white': Specifies the North American letter white medium

5524 'na-letter-colored': Specifies the North American letter colored medium

5525 'na-letter-transparent': Specifies the North American letter transparent medium

5526 'na-legal-white': Specifies the North American legal white medium

5527 'na-legal-colored': Specifies the North American legal colored medium

5528

5529 The following standard values are defined for envelopes:

5530 'iso-b4-envelope': Specifies the ISO B4 envelope medium

5531 'iso-b5-envelope': Specifies the ISO B5 envelope medium

5532 'iso-c3-envelope': Specifies the ISO C3 envelope medium

5533 'iso-c4-envelope': Specifies the ISO C4 envelope medium

5534 'iso-c5-envelope': Specifies the ISO C5 envelope medium

5535 'iso-c6-envelope': Specifies the ISO C6 envelope medium

5536 'iso-designated-long-envelope': Specifies the ISO Designated Long envelope medium

5537 'na-10x13-envelope': Specifies the North American 10x13 envelope medium

5538 'na-9x12-envelope': Specifies the North American 9x12 envelope medium  
5539 'monarch-envelope': Specifies the Monarch envelope  
5540 'na-number-10-envelope': Specifies the North American number 10 business envelope medium  
5541 'na-7x9-envelope': Specifies the North American 7x9 inch envelope  
5542 'na-9x11-envelope': Specifies the North American 9x11 inch envelope  
5543 'na-10x14-envelope': Specifies the North American 10x14 inch envelope  
5544 'na-number-9-envelope': Specifies the North American number 9 business envelope  
5545 'na-6x9-envelope': Specifies the North American 6x9 inch envelope  
5546 'na-10x15-envelope': Specifies the North American 10x15 inch envelope  
5547

5548 The following standard values are defined for the less commonly used media (~~white-only~~):

5549 'executive-white': Specifies the white executive medium  
5550 'folio-white': Specifies the folio white medium  
5551 'invoice-white': Specifies the white invoice medium  
5552 'ledger-white': Specifies the white ledger medium  
5553 'quarto-white': Specified the white quarto medium  
5554 'iso-a0-white': Specifies the ISO A0 white medium: 841 mm x 1189 mm  
5555 'iso-a0-transparent': Specifies the ISO A0 transparent medium: 841 mm x 1189 mm  
5556 'iso-a0-translucent': Specifies the ISO A0 translucent medium: 841 mm x 1189 mm  
5557 'iso-a1-white': Specifies the ISO A1 white medium: 594 mm x 841 mm  
5558 'iso-a1-transparent': Specifies the ISO A1 transparent medium: 594 mm x 841 mm  
5559 'iso-a1-translucent': Specifies the ISO A1 translucent medium: 594 mm x 841 mm  
5560 'iso-a2-white': Specifies the ISO A2 white medium: 420 mm x 594 mm  
5561 'iso-a2-transparent': Specifies the ISO A2 transparent medium: 420 mm x 594 mm  
5562 'iso-a2-translucent': Specifies the ISO A2 translucent medium: 420 mm x 594 mm  
5563 'iso-a3-transparent': Specifies the ISO A3 transparent medium: 297 mm x 420 mm  
5564 'iso-a3-translucent': Specifies the ISO A3 translucent medium: 297 mm x 420 mm  
5565 'iso-a4-translucent': Specifies the ISO A4 translucent medium: 210 mm x 297 mm  
5566 'iso-a5-transparent': Specifies the ISO A5 transparent medium: 148 mm x 210 mm  
5567 'iso-a5-translucent': Specifies the ISO A5 translucent medium: 148 mm x 210 mm  
5568 'iso-a6-white': Specifies the ISO A6 white medium: 105 mm x 148 mm  
5569 'iso-a7-white': Specifies the ISO A7 white medium: 74 mm x 105 mm  
5570 'iso-a8-white': Specifies the ISO A8 white medium: 52 mm x 74 mm  
5571 'iso-a9-white': Specifies the ISO A9 white medium: 37 mm x 52 mm  
5572 'iso-10-white': Specifies the ISO A10 white medium: 26 mm x 37 mm  
5573 'iso-b0-white': Specifies the ISO B0 white medium: 1000 mm x 1414 mm  
5574 'iso-b1-white': Specifies the ISO B1 white medium: 707 mm x 1000 mm  
5575 'iso-b2-white': Specifies the ISO B2 white medium: 500 mm x 707 mm  
5576 'iso-b3-white': Specifies the ISO B3 white medium: 353 mm x 500 mm  
5577 'iso-b6-white': Specifies the ISO B6 white medium: 125 mm x 176 mm  
5578 'iso-b7-white': Specifies the ISO B7 white medium: 88 mm x 125 mm  
5579 'iso-b8-white': Specifies the ISO B8 white medium: 62 mm x 88 mm  
5580 'iso-b9-white': Specifies the ISO B9 white medium: 44 mm x 62 mm  
5581 'iso-b10-white': Specifies the ISO B10 white medium: 31 mm x 44 mm

5582 'jis-b0-white': Specifies the JIS B0 white medium: 1030 mm x 1456 mm  
5583 'jis-b0-transparent': Specifies the JIS B0 transparent medium: 1030 mm x 1456 mm  
5584 'jis-b0-translucent': Specifies the JIS B0 translucent medium: 1030 mm x 1456 mm  
5585 'jis-b1-white': Specifies the JIS B1 white medium: 728 mm x 1030 mm  
5586 'jis-b1-transparent': Specifies the JIS B1 transparent medium: 728 mm x 1030 mm  
5587 'jis-b1-translucent': Specifies the JIS B1 translucent medium: 728 mm x 1030 mm  
5588 'jis-b2-white': Specifies the JIS B2 white medium: 515 mm x 728 mm  
5589 'jis-b2-transparent': Specifies the JIS B2 transparent medium: 515 mm x 728 mm  
5590 'jis-b2-translucent': Specifies the JIS B2 translucent medium: 515 mm x 728 mm  
5591 'jis-b3-white': Specifies the JIS B3 white medium: 364 mm x 515 mm  
5592 'jis-b3-transparent': Specifies the JIS B3 transparent medium: 364 mm x 515 mm  
5593 'jis-b3-translucent': Specifies the JIS B3 translucent medium: 364 mm x 515 mm  
5594 'jis-b4-transparent': Specifies the JIS B4 transparent medium: 257 mm x 364 mm  
5595 'jis-b4-translucent': Specifies the JIS B4 translucent medium: 257 mm x 364 mm  
5596 'jis-b5-transparent': Specifies the JIS B5 transparent medium: 182 mm x 257 mm  
5597 'jis-b5-translucent': Specifies the JIS B5 translucent medium: 182 mm x 257 mm  
5598 'jis-b6-white': Specifies the JIS B6 white medium: 128 mm x 182 mm  
5599 'jis-b7-white': Specifies the JIS B7 white medium: 91 mm x 128 mm  
5600 'jis-b8-white': Specifies the JIS B8 white medium: 64 mm x 91 mm  
5601 'jis-b9-white': Specifies the JIS B9 white medium: 45 mm x 64 mm  
5602 'jis-b10-white': Specifies the JIS B10 white medium: 32 mm x 45 mm  
5603

5604 The following standard values are defined for American Standard (i.e. ANSI) engineering media (~~white~~  
5605 ~~only~~):

5606 'a-white': Specifies the engineering ANSI A size white medium: 8.5 inches x 11 inches  
5607 'a-transparent': Specifies the engineering ANSI A size transparent medium: 8.5 inches x 11 inches  
5608 'a-translucent': Specifies the engineering ANSI A size translucent medium: 8.5 inches x 11 inches  
5609 'b-white': Specifies the engineering ANSI B size white medium: 11 inches x 17 inches  
5610 'b-transparent': Specifies the engineering ANSI B size transparent medium: 11 inches x 17 inches  
5611 'b-translucent': Specifies the engineering ANSI B size translucent medium: 11 inches x 17 inches  
5612 'c-white': Specifies the engineering ANSI C size white medium: 17 inches x 22 inches  
5613 'c-transparent': Specifies the engineering ANSI C size transparent medium: 17 inches x 22 inches  
5614 'c-translucent': Specifies the engineering ANSI C size translucent medium: 17 inches x 22 inches  
5615 'd-white': Specifies the engineering ANSI D size white medium: 22 inches x 34 inches  
5616 'd-transparent': Specifies the engineering ANSI D size transparent medium: 22 inches x 34 inches  
5617 'd-translucent': Specifies the engineering ANSI D size translucent medium: 22 inches x 34 inches  
5618 'e-white': Specifies the engineering ANSI E size white medium: 34 inches x 44 inches  
5619 'e-transparent': Specifies the engineering ANSI E size transparent medium: 34 inches x 44 inches  
5620 'e-translucent': Specifies the engineering ANSI E size translucent medium: 34 inches x 44 inches  
5621

5622 The following standard values are defined for American Standard (i.e. ANSI) engineering media for devices  
5623 that provide the "synchro-cut" feature (see section 14.1):



5624 'axsynchro-white': Specifies the roll paper having the width of the longer edge (11 inches) of the  
5625 engineering ANSI A size white medium and cuts synchronizing with data.  
5626 'axsynchro-transparent': Specifies the roll paper having the width of the longer edge (11 inches) of the  
5627 engineering ANSI A size transparent medium and cuts synchronizing with data.  
5628 'axsynchro-translucent': Specifies the roll paper having the width of the longer edge (11 inches) of the  
5629 engineering ANSI A size translucent medium and cuts synchronizing with data.  
5630 'bxsynchro-white': Specifies the roll paper having the width of the longer edge (17 inches) of the  
5631 engineering ANSI B size white medium and cuts synchronizing with data.  
5632 'bxsynchro-transparent': Specifies the roll paper having the width of the longer edge (17 inches) of the  
5633 engineering ANSI B size transparent medium and cuts synchronizing with data.  
5634 'bxsynchro-translucent': Specifies the roll paper having the width of the longer edge (17 inches) of the  
5635 engineering ANSI B size translucent medium and cuts synchronizing with data.  
5636 'cxsynchro-white': Specifies the roll paper having the width of the longer edge (22 inches) of the  
5637 engineering ANSI C size white medium and cuts synchronizing with data.  
5638 'cxsynchro-transparent': Specifies the roll paper having the width of the longer edge (22 inches) of the  
5639 engineering ANSI C size transparent medium and cuts synchronizing with data.  
5640 'cxsynchro-translucent': Specifies the roll paper having the width of the longer edge (22 inches) of the  
5641 engineering ANSI C size translucent medium and cuts synchronizing with data.  
5642 'dxsynchro-white': Specifies the roll paper having the width of the longer edge (34 inches) of the  
5643 engineering ANSI D size white medium and cuts synchronizing with data.  
5644 'dxsynchro-transparent': Specifies the roll paper having the width of the longer edge (34 inches) of the  
5645 engineering ANSI D size transparent medium and cuts synchronizing with data.  
5646 'dxsynchro-translucent': Specifies the roll paper having the width of the longer edge (34 inches) of the  
5647 engineering ANSI D size translucent medium and cuts synchronizing with data.  
5648 'exsynchro-white': Specifies the roll paper having the width of the longer edge (44 inches) of the  
5649 engineering ANSI E size white medium and cuts synchronizing with data.  
5650 'exsynchro-transparent': Specifies the roll paper having the width of the longer edge (44 inches) of the  
5651 engineering ANSI E size transparent medium and cuts synchronizing with data.  
5652 'exsynchro-translucent': Specifies the roll paper having the width of the longer edge (44 inches) of the  
5653 engineering ANSI E size translucent medium and cuts synchronizing with data.  
5654

5655 The following standard values are defined for American Architectural engineering media:

5656 'arch-a-white': Specifies the Architectural A size white medium: 9 inches x 12 inches  
5657 'arch-a-transparent': Specifies the Architectural A size transparent medium: 9 inches x 12 inches  
5658 'arch-a-translucent': Specifies the Architectural A size translucent medium: 9 inches x 12 inches  
5659 'arch-b-white': Specifies the Architectural B size white medium: 12 inches x 18 inches  
5660 'arch-b-transparent': Specifies the Architectural B size transparent medium: 12 inches x 18 inches  
5661 'arch-b-translucent': Specifies the Architectural B size translucent medium: 12 inches x 18 inches  
5662 'arch-c-white': Specifies the Architectural C size white medium: 18 inches x 24 inches  
5663 'arch-c-transparent': Specifies the Architectural C size transparent medium: 18 inches x 24 inches  
5664 'arch-c-translucent': Specifies the Architectural C size translucent medium: 18 inches x 24 inches  
5665 'arch-d-white': Specifies the Architectural D size white medium: 24 inches x 36 inches  
5666 'arch-d-transparent': Specifies the Architectural D size transparent medium: 24 inches x 36 inches  
5667 'arch-d-translucent': Specifies the Architectural D size translucent medium: 24 inches x 36 inches



5668 'arch-e-white': Specifies the Architectural E size white medium: 36 inches x 48 inches  
5669 'arch-e-transparent': Specifies the Architectural E size transparent medium: 36 inches x 48 inches  
5670 'arch-e-translucent': Specifies the Architectural E size translucent medium: 36 inches x 48 inches  
5671

5672 The following standard values are defined for American Architectural engineering media for devices that  
5673 provide the "synchro-cut" feature (see section 14.1):

5674 'arch-axsynchro-white': Specifies the roll paper having the width of the longer edge (12 inches) of the  
5675 Architectural A size white medium and cuts synchronizing with data.  
5676 'arch-axsynchro-transparent': Specifies the roll paper having the width of the longer edge (12 inches) of  
5677 the Architectural A size transparent medium and cuts synchronizing with data.  
5678 'arch-axsynchro-translucent': Specifies the roll paper having the width of the longer edge (12 inches) of  
5679 the Architectural A size translucent medium and cuts synchronizing with data.  
5680 'arch-bxsynchro-white': Specifies the roll paper having the width of the longer edge (18 inches) of the  
5681 Architectural B size white medium and cuts synchronizing with data.  
5682 'arch-bxsynchro-transparent': Specifies the roll paper having the width of the longer edge (18 inches) of  
5683 the Architectural B size transparent medium and cuts synchronizing with data.  
5684 'arch-bxsynchro-translucent': Specifies the roll paper having the width of the longer edge (18 inches) of  
5685 the Architectural B size translucent medium and cuts synchronizing with data.  
5686 'arch-cxsynchro-white': Specifies the roll paper having the width of the longer edge (24 inches) of the  
5687 Architectural C size white medium and cuts synchronizing with data.  
5688 'arch-cxsynchro-transparent': Specifies the roll paper having the width of the longer edge (24 inches) of  
5689 the Architectural C size transparent medium and cuts synchronizing with data.  
5690 'arch-cxsynchro-translucent': Specifies the roll paper having the width of the longer edge (24 inches) of  
5691 the Architectural C size translucent medium and cuts synchronizing with data.  
5692 'arch-dxsynchro-white': Specifies the roll paper having the width of the longer edge (36 inches) of the  
5693 Architectural D size white medium and cuts synchronizing with data.  
5694 'arch-dxsynchro-transparent': Specifies the roll paper having the width of the longer edge (36 inches) of  
5695 the Architectural D size transparent medium and cuts synchronizing with data.  
5696 'arch-dxsynchro-translucent': Specifies the roll paper having the width of the longer edge (36 inches) of  
5697 the Architectural D size translucent medium and cuts synchronizing with data.  
5698 'arch-exsynchro-white': Specifies the roll paper having the width of the longer edge (48 inches) of the  
5699 Architectural E size white medium and cuts synchronizing with data.  
5700 'arch-exsynchro-transparent': Specifies the roll paper having the width of the longer edge (48 inches) of  
5701 the Architectural E size transparent medium and cuts synchronizing with data.  
5702 'arch-exsynchro-translucent': Specifies the roll paper having the width of the longer edge (48 inches) of  
5703 the Architectural E size translucent medium and cuts synchronizing with data.  
5704

5705 The following standard values are defined for Japanese and European Standard (i.e. ISO) engineering  
5706 media, which are of a long fixed size [ASME-Y14.1M]:

5707 'iso-a1x3-white': Specifies the ISO A1X3 white medium having the width of the longer edge (841 mm)  
5708 of the ISO A1 medium  
5709 'iso-a1x3-transparent': Specifies the ISO A1X3 transparent medium having the width of the longer edge  
5710 (841 mm) of the ISO A1 medium

5711 'iso-a1x3-translucent': Specifies the ISO A1X3 translucent medium having the width of the longer edge  
5712 (841 mm) of the ISO A1 medium

5713 'iso-a1x4-white': Specifies the ISO A1X4 white medium having the width of the longer edge (841 mm)  
5714 of the ISO A1 medium

5715 'iso-a1x4-transparent': Specifies the ISO A1X4 transparent medium having the width of the longer edge  
5716 (841 mm) of the ISO A1 medium

5717 'iso-a1x4-translucent': Specifies the ISO A1X4 translucent medium having the width of the longer  
5718 edge (841 mm) of the ISO A1 medium

5719 'iso-a2x3-white': Specifies the ISO A2X3 white medium having the width of the longer edge (594 mm)  
5720 of the ISO A2 medium

5721 'iso-a2x3-transparent': Specifies the ISO A2X3 transparent medium having the width of the longer edge  
5722 (594 mm) of the ISO A2 medium

5723 'iso-a2x3-translucent': Specifies the ISO A2X3 translucent medium having the width of the longer edge  
5724 (594 mm) of the ISO A2 medium

5725 'iso-a2x4-white': Specifies the ISO A2X4 white medium having the width of the longer edge (594 mm)  
5726 of the ISO A2 medium

5727 'iso-a2x4-transparent': Specifies the ISO A2X4 transparent medium having the width of the longer edge  
5728 (594 mm) of the ISO A2 medium

5729 'iso-a2x4-translucent': Specifies the ISO A2X4 translucent medium having the width of the longer edge  
5730 (594 mm) of the ISO A2 medium

5731 'iso-a2x5-white': Specifies the ISO A2X5 white medium having the width of the longer edge (594 mm)  
5732 of the ISO A2 medium

5733 'iso-a2x5-transparent': Specifies the ISO A2X5 transparent medium having the width of the longer edge  
5734 (594 mm) of the ISO A2 medium

5735 'iso-a2x5-translucent': Specifies the ISO A2X5 translucent medium having the width of the longer edge  
5736 (594 mm) of the ISO A2 medium

5737 'iso-a3x3-white': Specifies the ISO A3X3 white medium having the width of the longer edge (420 mm)  
5738 of the ISO A3 medium

5739 'iso-a3x3-transparent': Specifies the ISO A3X3 transparent medium having the width of the longer edge  
5740 (420 mm) of the ISO A3 medium

5741 'iso-a3x3-translucent': Specifies the ISO A3X3 translucent medium having the width of the longer edge  
5742 (420 mm) of the ISO A3 medium

5743 'iso-a3x4-white': Specifies the ISO A3X4 white medium having the width of the longer edge (420 mm)  
5744 of the ISO A3 medium

5745 'iso-a3x4-transparent': Specifies the ISO A3X4 transparent medium having the width of the longer edge  
5746 (420 mm) of the ISO A3 medium

5747 'iso-a3x4-translucent': Specifies the ISO A3X4 translucent medium having the width of the longer edge  
5748 (420 mm) of the ISO A3 medium

5749 'iso-a3x5-white': Specifies the ISO A3X5 white medium having the width of the longer edge (420 mm)  
5750 of the ISO A3 medium

5751 'iso-a3x5-transparent': Specifies the ISO A3X5 transparent medium having the width of the longer edge  
5752 (420 mm) of the ISO A3 medium

5753 'iso-a3x5-translucent': Specifies the ISO A3X5 translucent medium having the width of the longer edge  
5754 (420 mm) of the ISO A3 medium

5755 'iso-a3x6-white': Specifies the ISO A3X6 white medium having the width of the longer edge (420 mm)  
5756 of the ISO A3 medium

5757 'iso-a3x6-transparent': Specifies the ISO A3X6 transparent medium having the width of the longer edge  
5758 (420 mm) of the ISO A3 medium

5759 'iso-a3x6-translucent': Specifies the ISO A3X6 translucent medium having the width of the longer edge  
5760 (420 mm) of the ISO A3 medium

5761 'iso-a3x7-white': Specifies the ISO A3X7 white medium having the width of the longer edge (420 mm)  
5762 of the ISO A3 medium

5763 'iso-a3x7-transparent': Specifies the ISO A3X7 transparent medium having the width of the longer edge  
5764 (420 mm) of the ISO A3 medium

5765 'iso-a3x7-translucent': Specifies the ISO A3X7 translucent' medium having the width of the longer  
5766 edge (420 mm) of the ISO A3 medium

5767 'iso-a4x3-white': Specifies the ISO A4X3 white medium having the width of the longer edge (297 mm)  
5768 of the ISO A4 medium

5769 'iso-a4x3-transparent': Specifies the ISO A4X3 transparent medium having the width of the longer edge  
5770 (297 mm) of the ISO A4 medium

5771 'iso-a4x3-translucent': Specifies the ISO A4X3 translucent' medium having the width of the longer  
5772 edge (297 mm) of the ISO A4 medium

5773 'iso-a4x4-white': Specifies the ISO A4X4 white medium having the width of the longer edge (297 mm)  
5774 of the ISO A4 medium

5775 'iso-a4x4-transparent': Specifies the ISO A4X4 transparent medium having the width of the longer edge  
5776 (297 mm) of the ISO A4 medium

5777 'iso-a4x4-translucent': Specifies the ISO A4X4 translucent medium having the width of the longer edge  
5778 (297 mm) of the ISO A4 medium

5779 'iso-a4x5-white': Specifies the ISO A4X5 white medium having the width of the longer edge (297 mm)  
5780 of the ISO A4 medium

5781 'iso-a4x5-transparent': Specifies the ISO A4X5 transparent medium having the width of the longer edge  
5782 (297 mm) of the ISO A4 medium

5783 'iso-a4x5-translucent': Specifies the ISO A4X5 translucent medium having the width of the longer edge  
5784 (297 mm) of the ISO A4 medium

5785 'iso-a4x6-white': Specifies the ISO A4X6 white medium having the width of the longer edge (297 mm)  
5786 of the ISO A4 medium

5787 'iso-a4x6-transparent': Specifies the ISO A4X6 transparent medium having the width of the longer edge  
5788 (297 mm) of the ISO A4 medium

5789 'iso-a4x6-translucent': Specifies the ISO A4X6 translucent medium having the width of the longer edge  
5790 (297 mm) of the ISO A4 medium

5791 'iso-a4x7-white': Specifies the ISO A4X7 white medium having the width of the longer edge (297 mm)  
5792 of the ISO A4 medium

5793 'iso-a4x7-transparent': Specifies the ISO A4X7 transparent medium having the width of the longer edge  
5794 (297 mm) of the ISO A4 medium

5795 'iso-a4x7-translucent': Specifies the ISO A4X7 translucent medium having the width of the longer edge  
5796 (297 mm) of the ISO A4 medium

5797 'iso-a4x8-white': Specifies the ISO A4X8 white medium having the width of the longer edge (297 mm)  
5798 of the ISO A4 medium

- 5799 'iso-a4x8-transparent': Specifies the ISO A4X8 transparent medium having the width of the longer edge  
5800 (297 mm) of the ISO A4 medium
- 5801 'iso-a4x8-translucent': Specifies the ISO A4X8 translucent medium having the width of the longer edge  
5802 (297 mm) of the ISO A4 medium
- 5803 'iso-a4x9-white': Specifies the ISO A4X9 white medium having the width of the longer edge (297 mm)  
5804 of the ISO A4 medium
- 5805 'iso-a4x9-transparent': Specifies the ISO A4X9 transparent medium having the width of the longer edge  
5806 (297 mm) of the ISO A4 medium
- 5807 'iso-a4x9-translucent': Specifies the ISO A4X9 translucent medium having the width of the longer edge  
5808 (297 mm) of the ISO A4 medium
- 5809

5810 The following standard values are defined for Japanese and European Standard (i.e. ISO) engineering  
5811 media, which are either a long fixed size [ASME-Y14.1M] or roll feed, for devices that provide the  
5812 "synchro-cut" feature (see section 14.1):

- 5813 'iso-a0xsynchro-white': Specifies the paper having the width of the longer edge (1189 mm) of the ISO  
5814 A0 white medium and cuts synchronizing with data.
- 5815 'iso-a0xsynchro-transparent': Specifies the paper having the width of the longer edge (1189 mm) of the  
5816 ISO A0 transparent medium and cuts synchronizing with data.
- 5817 'iso-a0xsynchro-translucent': Specifies the paper having the width of the longer edge (1189 mm) of the  
5818 ISO A0 translucent medium and cuts synchronizing with data.
- 5819 'iso-a1xsynchro-white': Specifies the paper having the width of the longer edge (841 mm) of the ISO  
5820 A1 white medium and cuts synchronizing with data.
- 5821 'iso-a1xsynchro-transparent': Specifies the paper having the width of the longer edge (841 mm) of the  
5822 ISO A1 transparent medium and cuts synchronizing with data.
- 5823 'iso-a1xsynchro-translucent': Specifies the paper having the width of the longer edge (841 mm) of the  
5824 ISO A1 translucent medium and cuts synchronizing with data.
- 5825 'iso-a2xsynchro-white': Specifies the paper having the width of the longer edge (594 mm) of the ISO  
5826 A2 white medium and cuts synchronizing with data.
- 5827 'iso-a2xsynchro-transparent': Specifies the paper having the width of the longer edge (594 mm) of the  
5828 ISO A2 transparent medium and cuts synchronizing with data.
- 5829 'iso-a2xsynchro-translucent': Specifies the paper having the width of the longer edge (594 mm) of the  
5830 ISO A2 translucent medium and cuts synchronizing with data.
- 5831 'iso-a3xsynchro-white': Specifies the paper having the width of the longer edge (420 mm) of the ISO  
5832 A3 white medium and cuts synchronizing with data.
- 5833 'iso-a3xsynchro-transparent': Specifies the paper having the width of the longer edge (420 mm) of the  
5834 ISO A3 transparent medium and cuts synchronizing with data.
- 5835 'iso-a3xsynchro-translucent': Specifies the paper having the width of the longer edge (420 mm) of the  
5836 ISO A3 translucent medium and cuts synchronizing with data.
- 5837 'iso-a4xsynchro-white': Specifies the paper having the width of the longer edge (297 mm) of the ISO  
5838 A4 white medium and cuts synchronizing with data.
- 5839 'iso-a4xsynchro-transparent': Specifies the paper having the width of the longer edge (297 mm) of the  
5840 ISO A4 transparent medium and cuts synchronizing with data.
- 5841 'iso-a4xsynchro-translucent': Specifies the paper having the width of the longer edge (297 mm) of the  
5842 ISO A4 transparent medium and cuts synchronizing with data.

5843

5844 The following standard values are defined for American Standard (i.e. ANSI) engineering media, American  
5845 Architectural engineering media, and Japanese and European Standard (i.e. ISO) engineering media, which  
5846 are either a long fixed size [ASME-Y14.1M] or roll feed, for devices that provide the "synchro-cut" feature  
5847 and/or the "auto-select" feature (see section 14.1):

5848 'auto-white': Specifies that the printer selects the white medium with the appropriate fixed size (e.g. a1,  
5849 a2, etc.) or data-synchro size, and the selection is implementation-defined.

5850 'auto-transparent': Specifies that the printer selects the transparent medium with the appropriate fixed  
5851 size (e.g. a1, a2, etc.) or data-synchro size, and the selection is implementation-defined.

5852 'auto-translucent': Specifies that the printer selects the translucent medium with the appropriate fixed  
5853 size (e.g. a1, a2, etc.) or data-synchro size, and the selection is implementation-defined.

5854 'auto-fixed-size-white': Specifies that the printer selects the white medium with the appropriate fixed  
5855 size (e.g. a1, a2, etc.) or the appropriate long fixed size listed above.

5856 'auto-fixed-size-transparent': Specifies that the printer selects the transparent medium with the  
5857 appropriate fixed size (e.g. a1, a2, etc.) or the appropriate long fixed size listed above.

5858 'auto-fixed-size-translucent': Specifies that the printer selects the translucent medium with the  
5859 appropriate fixed size (e.g. a1, a2, etc.) or the appropriate long fixed size listed above.

5860 'auto-synchro-white': Specifies that the printer selects the white paper with the appropriate width and  
5861 cuts it synchronizing with data.

5862 'auto-synchro-transparent': Specifies that the printer selects the transparent paper with the appropriate  
5863 width and cuts it synchronizing with data.

5864 'auto-synchro-translucent': Specifies that the printer selects the translucent paper with the appropriate  
5865 width and cuts it synchronizing with data.

5866

5867 The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

5868 'top': The top input tray in the printer.

5869 'middle': The middle input tray in the printer.

5870 'bottom': The bottom input tray in the printer.

5871 'envelope': The envelope input tray in the printer.

5872 'manual': The manual feed input tray in the printer.

5873 'large-capacity': The large capacity input tray in the printer.

5874 'main': The main input tray

5875 'side': The side input tray

5876

5877 The following standard values are defined for media sizes (from ISO DPA):

5878 'iso-a0': Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216

5879 'iso-a1': Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216

5880 'iso-a2': Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216

5881 'iso-a3': Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216

5882 'iso-a4': Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216

5883 'iso-a5': Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216



5884 'iso-a6': Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216  
5885 'iso-a7': Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216  
5886 'iso-a8': Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216  
5887 'iso-a9': Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216  
5888 'iso-a10': Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216  
5889 'iso-b0': Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216  
5890 'iso-b1': Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216  
5891 'iso-b2': Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216  
5892 'iso-b3': Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216  
5893 'iso-b4': Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216  
5894 'iso-b5': Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216  
5895 'iso-b6': Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216  
5896 'iso-b7': Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216  
5897 'iso-b8': Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216  
5898 'iso-b9': Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216  
5899 'iso-b10': Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216  
5900 'na-letter': Specifies the North American letter size: 8.5 inches by 11 inches  
5901 'na-legal': Specifies the North American legal size: 8.5 inches by 14 inches  
5902 ['na-8x10': Specifies the North American 8 inches by 10 inches](#)  
5903 ['na-5x7': Specifies the North American 5 inches by 7 inches](#)  
5904 'executive': Specifies the executive size (7.25 X 10.5 in)  
5905 'folio': Specifies the folio size (8.5 X 13 in)  
5906 'invoice': Specifies the invoice size (5.5 X 8.5 in)  
5907 'ledger': Specifies the ledger size (11 X 17 in)  
5908 'quarto': Specifies the quarto size (8.5 X 10.83 in)  
5909 'iso-c3': Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269  
5910 'iso-c4': Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269  
5911 'iso-c5': Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269  
5912 'iso-c6': Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269  
5913 'iso-designated-long': Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO  
5914 269  
5915 'na-10x13-envelope': Specifies the North American 10x13 size: 10 inches by 13 inches  
5916 'na-9x12-envelope': Specifies the North American 9x12 size: 9 inches by 12 inches  
5917 'na-number-10-envelope': Specifies the North American number 10 business envelope size: 4.125  
5918 inches by 9.5 inches  
5919 'na-7x9-envelope': Specifies the North American 7x9 inch envelope size  
5920 'na-9x11-envelope': Specifies the North American 9x11 inch envelope size  
5921 'na-10x14-envelope': Specifies the North American 10x14 inch envelope size  
5922 'na-number-9-envelope': Specifies the North American number 9 business envelope size  
5923 'na-6x9-envelope': Specifies the North American 6x9 envelope size  
5924 'na-10x15-envelope': Specifies the North American 10x15 envelope size  
5925 'monarch-envelope': Specifies the Monarch envelope size (3.87 x 7.5 in)  
5926 'jis-b0': Specifies the JIS B0 size: 1030mm x 1456mm  
5927 'jis-b1': Specifies the JIS B1 size: 728mm x 1030mm  
5928 'jis-b2': Specifies the JIS B2 size: 515mm x 728mm

5929 'jis-b3': Specifies the JIS B3 size: 364mm x 515mm  
5930 'jis-b4': Specifies the JIS B4 size: 257mm x 364mm  
5931 'jis-b5': Specifies the JIS B5 size: 182mm x 257mm  
5932 'jis-b6': Specifies the JIS B6 size: 128mm x 182mm  
5933 'jis-b7': Specifies the JIS B7 size: 91mm x 128mm  
5934 'jis-b8': Specifies the JIS B8 size: 64mm x 91mm  
5935 'jis-b9': Specifies the JIS B9 size: 45mm x 64mm  
5936 'jis-b10': Specifies the JIS B10 size: 32mm x 45mm

5937 The following standard values are defined for American Standard (i.e. ANSI) engineering media sizes:

5938 'a': Specifies the engineering ANSI A size medium: 8.5 inches x 11 inches  
5939 'b': Specifies the engineering ANSI B size medium: 11 inches x 17 inches  
5940 'c': Specifies the engineering ANSI C size medium: 17 inches x 22 inches  
5941 'd': Specifies the engineering ANSI D size medium: 22 inches x 34 inches  
5942 'e': Specifies the engineering ANSI E size medium: 34 inches x 44 inches  
5943

5944 The following standard values are defined for American Architectural engineering media sizes:

5945 'arch-a': Specifies the Architectural A size medium: 9 inches x 12 inches  
5946 'arch-b': Specifies the Architectural B size medium: 12 inches x 18 inches  
5947 'arch-c': Specifies the Architectural C size medium: 18 inches x 24 inches  
5948 'arch-d': Specifies the Architectural D size medium: 24 inches x 36 inches  
5949 'arch-e': Specifies the Architectural E size medium: 36 inches x 48 inches  
5950





5983

5984

5985 Example 2: "Auto-Cut", a device cutting the roll paper at multiples of fixed-size media width

5986 data height: A1 height

5987 data width (shaded): A1 width < data width < (A1 width) x 2

5988 specified value: 'auto-fixed-size-white'

5989

5990

5991

5992

5993

5994

5995

5996

5997

5998

5999

6000

6001

6002

6003

6004

6005

6006

6007

6008

6009

6010

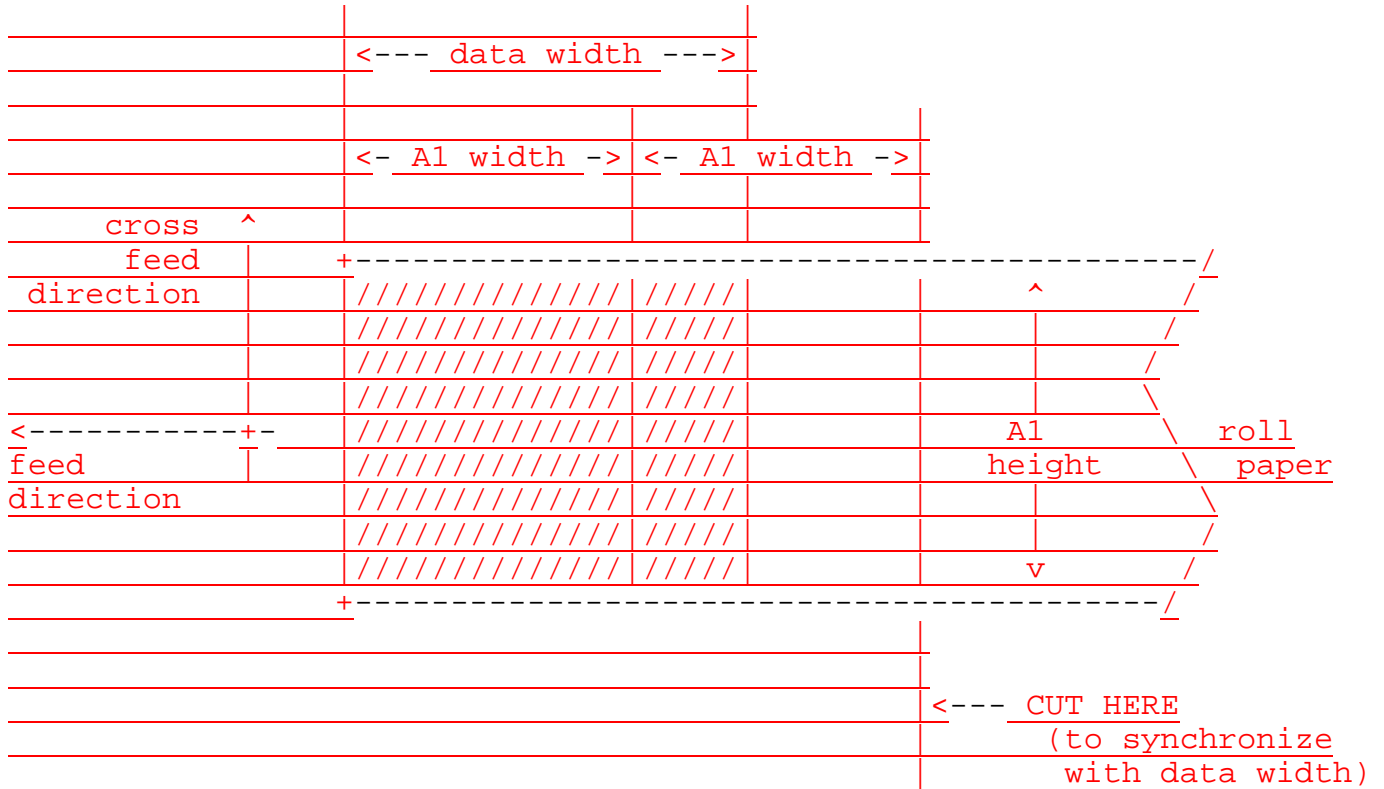
6011

6012

6013

6014

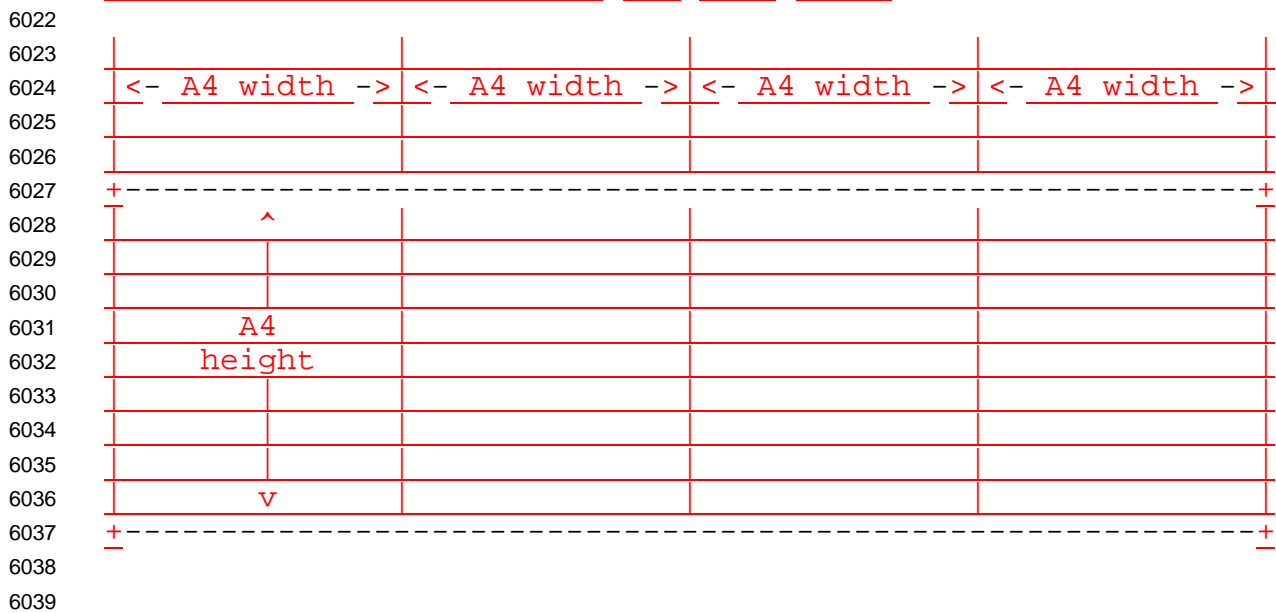
6015



6016  
6017

6018 Example 3: the 'iso-a4x4-white' fixed size paper

6019	<u>paper height:</u>	<u>A4 height</u>
6020	<u>paper width:</u>	<u>(A4 width) x 4</u>
6021	<u>specified value:</u>	<u>'iso-a4x4-white'</u>



6040

6041

6042 Example 4: "Synchro-Cut", a device cutting the fixed size paper in synchronization with the data

6043 data height: A4 height  
 6044 data width (shaded): (A4 width) x 2 < data width < (A4 width) x 3  
 6045 specified value: 'iso-a4xsynchro-white'

6046

6047

6048

6049

6050

6051

6052

6053

6054

6055

6056

6057

6058

6059

6060

6061

6062

6063

6064

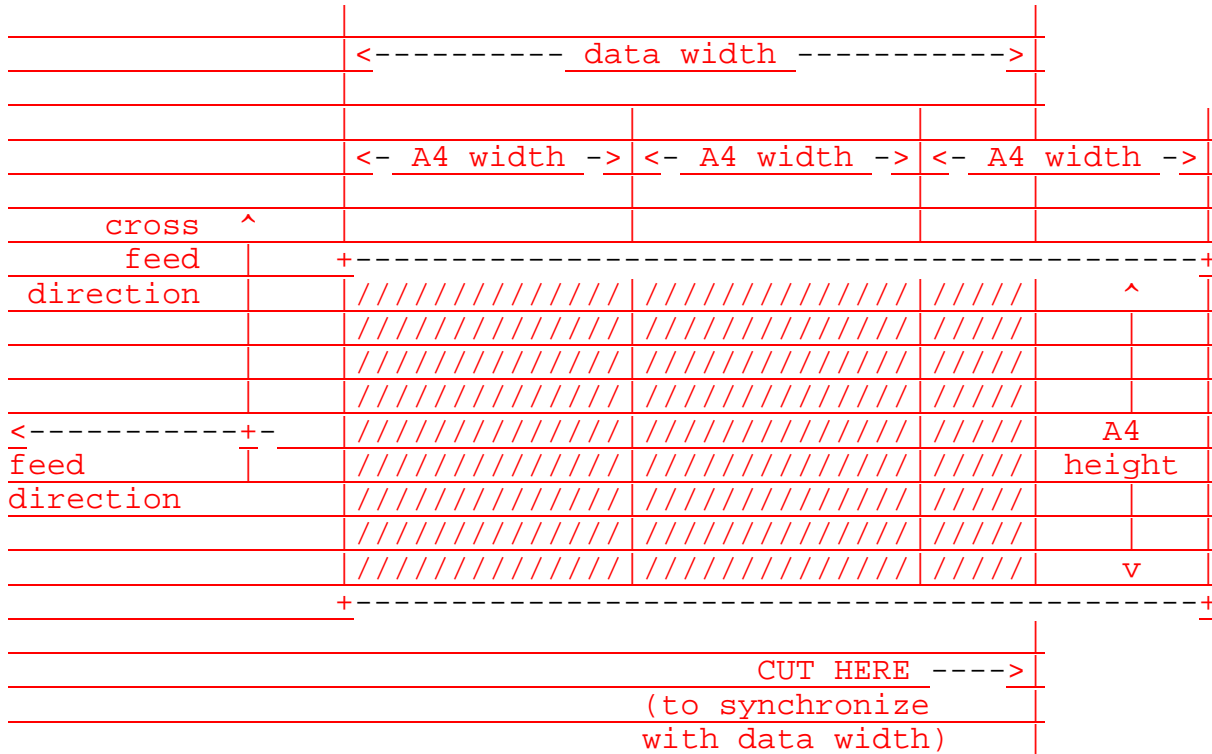
6065

6066

6067

6068

6069



6070

6071 Standard keyword values are taken from several sources.

6072 Standard values are defined (taken from DPA[ISO10175] and the Printer MIB[RFC1759]):

6073 'default': The default medium for the output device  
6074 'iso-a4-white': Specifies the ISO A4 white medium  
6075 'iso-a4-colored': Specifies the ISO A4 colored medium  
6076 'iso-a4-transparent' Specifies the ISO A4 transparent medium  
6077 'iso-a3-white': Specifies the ISO A3 white medium  
6078 'iso-a3-colored': Specifies the ISO A3 colored medium  
6079 'iso-a5-white': Specifies the ISO A5 white medium  
6080 'iso-a5-colored': Specifies the ISO A5 colored medium  
6081 'iso-b4-white': Specifies the ISO B4 white medium  
6082 'iso-b4-colored': Specifies the ISO B4 colored medium  
6083 'iso-b5-white': Specifies the ISO B5 white medium  
6084 'iso-b5-colored': Specifies the ISO B5 colored medium  
6085 'jis-b4-white': Specifies the JIS B4 white medium  
6086 'jis-b4-colored': Specifies the JIS B4 colored medium  
6087 'jis-b5-white': Specifies the JIS B5 white medium  
6088 'jis-b5-colored': Specifies the JIS B5 colored medium  
6089

6090 The following standard values are defined for North American media:

6091 'na-letter-white': Specifies the North American letter white medium  
6092 'na-letter-colored': Specifies the North American letter colored medium  
6093 'na-letter-transparent': Specifies the North American letter transparent medium  
6094 'na-legal-white': Specifies the North American legal white medium  
6095 'na-legal-colored': Specifies the North American legal colored medium  
6096

6097 The following standard values are defined for envelopes:

6098 'iso-b4-envelope': Specifies the ISO B4 envelope medium  
6099 'iso-b5-envelope': Specifies the ISO B5 envelope medium  
6100 'iso-c3-envelope': Specifies the ISO C3 envelope medium  
6101 'iso-c4-envelope': Specifies the ISO C4 envelope medium  
6102 'iso-c5-envelope': Specifies the ISO C5 envelope medium  
6103 'iso-c6-envelope': Specifies the ISO C6 envelope medium  
6104 'iso-designated-long-envelope': Specifies the ISO Designated Long envelope medium  
6105 'na-10x13-envelope': Specifies the North American 10x13 envelope medium  
6106 'na-9x12-envelope': Specifies the North American 9x12 envelope medium  
6107 'monarch-envelope': Specifies the Monarch envelope  
6108 'na-number-10-envelope': Specifies the North American number 10 business envelope medium  
6109 'na-7x9-envelope': Specifies the North American 7x9 inch envelope  
6110 'na-9x11-envelope': Specifies the North American 9x11 inch envelope

6111 'na-10x14-envelope': Specifies the North American 10x14 inch envelope  
6112 'na-number-9-envelope': Specifies the North American number 9 business envelope  
6113 'na-6x9-envelope': Specifies the North American 6x9 inch envelope  
6114 'na-10x15-envelope': Specifies the North American 10x15 inch envelope  
6115

6116 The following standard values are defined for the less commonly used media (white-only):

6117 'executive-white': Specifies the white executive medium  
6118 'folio-white': Specifies the folio white medium  
6119 'invoice-white': Specifies the white invoice medium  
6120 'ledger-white': Specifies the white ledger medium  
6121 'quarto-white': Specified the white quarto medium  
6122 'iso-a0-white': Specifies the ISO A0 white medium  
6123 'iso-a1-white': Specifies the ISO A1 white medium  
6124 'iso-a2-white': Specifies the ISO A2 white medium  
6125 'iso-a6-white': Specifies the ISO A6 white medium  
6126 'iso-a7-white': Specifies the ISO A7 white medium  
6127 'iso-a8-white': Specifies the ISO A8 white medium  
6128 'iso-a9-white': Specifies the ISO A9 white medium  
6129 'iso-10-white': Specifies the ISO A10 white medium  
6130 'iso-b0-white': Specifies the ISO B0 white medium  
6131 'iso-b1-white': Specifies the ISO B1 white medium  
6132 'iso-b2-white': Specifies the ISO B2 white medium  
6133 'iso-b3-white': Specifies the ISO B3 white medium  
6134 'iso-b6-white': Specifies the ISO B6 white medium  
6135 'iso-b7-white': Specifies the ISO B7 white medium  
6136 'iso-b8-white': Specifies the ISO B8 white medium  
6137 'iso-b9-white': Specifies the ISO B9 white medium  
6138 'iso-b10-white': Specifies the ISO B10 white medium  
6139 'jis-b0-white': Specifies the JIS B0 white medium  
6140 'jis-b1-white': Specifies the JIS B1 white medium  
6141 'jis-b2-white': Specifies the JIS B2 white medium  
6142 'jis-b3-white': Specifies the JIS B3 white medium  
6143 'jis-b6-white': Specifies the JIS B6 white medium  
6144 'jis-b7-white': Specifies the JIS B7 white medium  
6145 'jis-b8-white': Specifies the JIS B8 white medium  
6146 'jis-b9-white': Specifies the JIS B9 white medium  
6147 'jis-b10-white': Specifies the JIS B10 white medium  
6148

6149 The following standard values are defined for engineering media (white only):

6150 'a-white': Specifies the engineering A size medium  
6151 'b-white': Specifies the engineering B size medium  
6152 'c-white': Specifies the engineering C size medium

6153 'd-white': Specifies the engineering D size medium

6154 'e-white': Specifies the engineering E size medium

6155

6156 The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

6157 'top': The top input tray in the printer.

6158 'middle': The middle input tray in the printer.

6159 'bottom': The bottom input tray in the printer.

6160 'envelope': The envelope input tray in the printer.

6161 'manual': The manual feed input tray in the printer.

6162 'large-capacity': The large capacity input tray in the printer.

6163 'main': The main input tray

6164 'side': The side input tray

6165

6166 The following standard values are defined for media sizes (from ISO DPA):

6167 'iso-a0': Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216

6168 'iso-a1': Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216

6169 'iso-a2': Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216

6170 'iso-a3': Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216

6171 'iso-a4': Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216

6172 'iso-a5': Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216

6173 'iso-a6': Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216

6174 'iso-a7': Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216

6175 'iso-a8': Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216

6176 'iso-a9': Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216

6177 'iso-a10': Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216

6178 'iso-b0': Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216

6179 'iso-b1': Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216

6180 'iso-b2': Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216

6181 'iso-b3': Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216

6182 'iso-b4': Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216

6183 'iso-b5': Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216

6184 'iso-b6': Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216

6185 'iso-b7': Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216

6186 'iso-b8': Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216

6187 'iso-b9': Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216

6188 'iso-b10': Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216

6189 'na-letter': Specifies the North American letter size: 8.5 inches by 11 inches

6190 'na-legal': Specifies the North American legal size: 8.5 inches by 14 inches

6191 'executive': Specifies the executive size (7.25 X 10.5 in)

6192 'folio': Specifies the folio size (8.5 X 13 in)

6193 'invoice': Specifies the invoice size (5.5 X 8.5 in)

6194 'ledger': Specifies the ledger size (11 X 17 in)



6195 'quarto': Specifies the quarto size (8.5 X 10.83 in)  
6196 'iso-c3': Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269  
6197 'iso-c4': Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269  
6198 'iso-c5': Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269  
6199 'iso-c6': Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269  
6200 'iso-designated-long': Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO  
6201 269  
6202 'na-10x13-envelope': Specifies the North American 10x13 size: 10 inches by 13 inches  
6203 'na-9x12-envelope': Specifies the North American 9x12 size: 9 inches by 12 inches  
6204 'na-number-10-envelope': Specifies the North American number 10 business envelope size: 4.125  
6205 inches by 9.5 inches  
6206 'na-7x9-envelope': Specifies the North American 7x9 inch envelope size  
6207 'na-9x11-envelope': Specifies the North American 9x11 inch envelope size  
6208 'na-10x14-envelope': Specifies the North American 10x14 inch envelope size  
6209 'na-number-9-envelope': Specifies the North American number 9 business envelope size  
6210 'na-6x9-envelope': Specifies the North American 6x9 envelope size  
6211 'na-10x15-envelope': Specifies the North American 10x15 envelope size  
6212 'monarch-envelope': Specifies the Monarch envelope size (3.87 x 7.5 in)  
6213 'jis-b0': Specifies the JIS B0 size: 1030mm x 1456mm  
6214 'jis-b1': Specifies the JIS B1 size: 728mm x 1030mm  
6215 'jis-b2': Specifies the JIS B2 size: 515mm x 728mm  
6216 'jis-b3': Specifies the JIS B3 size: 364mm x 515mm  
6217 'jis-b4': Specifies the JIS B4 size: 257mm x 364mm  
6218 'jis-b5': Specifies the JIS B5 size: 182mm x 257mm  
6219 'jis-b6': Specifies the JIS B6 size: 128mm x 182mm  
6220 'jis-b7': Specifies the JIS B7 size: 91mm x 128mm  
6221 'jis-b8': Specifies the JIS B8 size: 64mm x 91mm  
6222 'jis-b9': Specifies the JIS B9 size: 45mm x 64mm  
6223 'jis-b10': Specifies the JIS B10 size: 32mm x 45mm  
6224 The following standard values are defined for engineering media sizes:  
6225 'a': Specifies the engineering A size: 8.5 inches x 11 inches  
6226 'b': Specifies the engineering B size: 11 inches x 17 inches  
6227 'c': Specifies the engineering C size: 17 inches x 22 inches  
6228 'd': Specifies the engineering D size: 22 inches x 34 inches  
6229 'e': Specifies the engineering E size: 34 inches x 44 inches  
6230

## 6231 **15. APPENDIX D: Processing IPP Attributes**

6232 When submitting a print job to a Printer object, the IPP model allows a client to supply operation and Job  
6233 Template attributes along with the document data. These Job Template attributes in the create request  
6234 affect the rendering, production and finishing of the documents in the job. Similar types of instructions  
6235 may also be contained in the document to be printed, that is, embedded within the print data itself. In

6236 addition, the Printer has a set of attributes that describe what rendering and finishing options which are  
6237 supported by that Printer. This model, which allows for flexibility and power, also introduces the potential  
6238 that at job submission time, these client-supplied attributes may conflict with either:

- 6239 - what the implementation is capable of realizing (i.e., what the Printer supports), as well as
- 6240 - the instructions embedded within the print data itself.

6241

6242 The following sections describe how these two types of conflicts are handled in the IPP model.

## 6243 **15.1 Fidelity**

6244 If there is a conflict between what the client requests and what a Printer object supports, the client may  
6245 request one of two possible conflict handling mechanisms:

- 6246 1) either reject the job since the job can not be processed exactly as specified, or
- 6247 2) allow the Printer to make any changes necessary to proceed with processing the Job the best it can.

6248

6249 In the first case the client is indicating to the Printer object: "Print the job exactly as specified with no  
6250 exceptions, and if that can't be done, don't even bother printing the job at all." In the second case, the client  
6251 is indicating to the Printer object: "It is more important to make sure the job is printed rather than be  
6252 processed exactly as specified; just make sure the job is printed even if client supplied attributes need to be  
6253 changed or ignored."

6254 The IPP model accounts for this situation by introducing an "ipp-attribute-fidelity" attribute.

6255 In a create request, "ipp-attribute-fidelity" is a boolean operation attribute that is **OPTIONALLY** supplied  
6256 by the client. The value 'true' indicates that total fidelity to client supplied Job Template attributes and  
6257 values is required. The client is requesting that the Job be printed exactly as specified, and if that is not  
6258 possible then the job **MUST** be rejected rather than processed incorrectly. The value 'false' indicates that a  
6259 reasonable attempt to print the Job is acceptable. If a Printer does not support some of the client supplied  
6260 Job Template attributes or values, the Printer **MUST** ignore them or substitute any supported value for  
6261 unsupported values, respectively. The Printer may choose to substitute the default value associated with  
6262 that attribute, or use some other supported value that is similar to the unsupported requested value. For  
6263 example, if a client supplies a "media" value of 'na-letter', the Printer may choose to substitute 'iso-a4' rather  
6264 than a default value of 'envelope'. If the client does not supply the "ipp-attribute-fidelity" attribute, the  
6265 Printer assumes a value of 'false'.

6266 Each Printer implementation **MUST** support both types of "fidelity" printing (that is whether the client  
6267 supplies a value of 'true' or 'false'):

- 6268 - If the client supplies 'false' or does not supply the attribute, the Printer object **MUST** always accept the  
6269 request by ignoring unsupported Job Template attributes and by substituting unsupported values of  
6270 supported Job Template attributes with supported values.
- 6271 - If the client supplies 'true', the Printer object **MUST** reject the request if the client supplies  
6272 unsupported Job Template attributes.

6273

6274 Since a client can always query a Printer to find out exactly what is and is not supported, "ipp-attribute-  
6275 fidelity" set to 'false' is useful when:

- 6276 1) The End-User uses a command line interface to request attributes that might not be supported.
- 6277 2) In a GUI context, if the End User expects the job might be moved to another printer and prefers a  
6278 sub-optimal result to nothing at all.
- 6279 3) The End User just wants something reasonable in lieu of nothing at all.

6280

## 6281 15.2 Page Description Language (PDL) Override

6282 If there is a conflict between the value of an IPP Job Template attribute and a corresponding instruction in  
6283 the document data, the value of the IPP attribute SHOULD take precedence over the document instruction.  
6284 Consider the case where a previously formatted file of document data is sent to an IPP Printer. In this case,  
6285 if the client supplies any attributes at job submission time, the client desires that those attributes override  
6286 the embedded instructions. Consider the case were a previously formatted document has embedded in it  
6287 commands to load 'iso-a4' media. However, the document is passed to an end user that only has access to a  
6288 printer with 'na-letter' media loaded. That end user most likely wants to submit that document to an IPP  
6289 Printer with the "media" Job Template attribute set to 'na-letter'. The job submission attribute should take  
6290 precedence over the embedded PDL instruction. However, until companies that supply document data  
6291 interpreters allow a way for external IPP attributes to take precedence over embedded job production  
6292 instructions, a Printer might not be able to support the semantics that IPP attributes override the embedded  
6293 instructions.

6294 The IPP model accounts for this situation by introducing a "pdl-override-supported" attribute that describes  
6295 the Printer objects capabilities to override instructions embedded in the PDL data stream. The value of the  
6296 "pdl-override-supported" attribute is configured by means outside the scope of this IPP/1.1 document.

6297 This REQUIRED Printer attribute takes on the following values:

- 6298 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values take  
6299 precedence over embedded instructions in the document data, however there is no guarantee.
- 6300 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP attribute  
6301 values take precedence over embedded instructions in the document data.

6302

6303 At job processing time, an implementation that supports the value of 'attempted' might do one of several  
6304 different actions:

- 6305 1) Generate an output device specific command sequence to realize the feature represented by the IPP  
6306 attribute value.
- 6307 2) Parse the document data itself and replace the conflicting embedded instruction with a new  
6308 embedded instruction that matches the intent of the IPP attribute value.
- 6309 3) Indicate to the Printer that external supplied attributes take precedence over embedded instructions  
6310 and then pass the external IPP attribute values to the document data interpreter.

- 6311 4) Anything else that allows for the semantics that IPP attributes override embedded document data  
6312 instructions.  
6313

6314 Since 'attempted' does not offer any type of guarantee, even though a given Printer object might not do a  
6315 very "good" job of attempting to ensure that IPP attributes take a higher precedence over instructions  
6316 embedded in the document data, it would still be a conforming implementation.

6317 At job processing time, an implementation that supports the value of 'not-attempted' might do one of the  
6318 following actions:

- 6319 1) Simply pre-pend the document data with the PDL instruction that corresponds to the client-supplied  
6320 PDL attribute, such that if the document data also has the same PDL instruction, it will override  
6321 what the Printer object pre-pended. In other words, this implementation is using the same  
6322 implementation semantics for the client-supplied IPP attributes as for the Printer object defaults.  
6323 2) Parse the document data and replace the conflicting embedded instruction with a new embedded  
6324 instruction that approximates, but does not match, the semantic intent of the IPP attribute value.  
6325

6326 Note: The "ipp-attribute-fidelity" attribute applies to the Printer's ability to either accept or reject other  
6327 unsupported Job Template attributes. In other words, if "ipp-attribute-fidelity" is set to 'true', a Job is  
6328 accepted if and only if the client supplied Job Template attributes and values are supported by the Printer.  
6329 Whether these attributes actually affect the processing of the Job when the document data contains  
6330 embedded instructions depends on the ability of the Printer to override the instructions embedded in the  
6331 document data with the semantics of the IPP attributes. If the document data attributes can be overridden  
6332 ("pdl-override-supported" set to 'attempted'), the Printer makes an attempt to use the IPP attributes when  
6333 processing the Job. If the document data attributes can not be overridden ("pdl-override-supported" set to  
6334 'not-attempted'), the Printer makes no attempt to override the embedded document data instructions with the  
6335 IPP attributes when processing the Job, and hence, the IPP attributes may fail to affect the Job processing  
6336 and output when the corresponding instruction is embedded in the document data.

### 6337 **15.3 Using Job Template Attributes During Document Processing.**

6338 The Printer object uses some of the Job object's Job Template attributes during the processing of the  
6339 document data associated with that job. These include, but are not limited to, "orientation-requested",  
6340 "number-up", "sides", "media", and "copies". The processing of each document in a Job Object MUST  
6341 follow the steps below. These steps are intended only to identify when and how attributes are to be used in  
6342 processing document data and any alternative steps that accomplishes the same effect can be used to  
6343 implement this specification document.

- 6344 1. Using the client supplied "document-format" attribute or some form of document format detection  
6345 algorithm (if the value of "document-format" is not specific enough), determine whether or not the  
6346 document data has already been formatted for printing. If the document data has been formatted,  
6347 then go to step 2. Otherwise, the document data MUST be formatted. The formatting detection  
6348 algorithm is implementation defined and is not specified by this document. The formatting of the  
6349 document data uses the "orientation-requested" attribute to determine how the formatted print data  
6350 should be placed on a print-stream page, see section 4.2.10 for the details.

- 6351
- 6352 2. The document data is in the form of a print-stream in a known media type. The "page-ranges"
- 6353 attribute is used to select, as specified in section 4.2.7, a sub-sequence of the pages in the print-
- 6354 stream that are to be processed and images.
- 6355
- 6356 3. The input to this step is a sequence of print-stream pages. This step is controlled by the "number-up"
- 6357 attribute. If the value of "number-up" is N, then during the processing of the print-stream pages,
- 6358 each N print-stream pages are positioned, as specified in section 4.2.9, to create a single impression.
- 6359 If a given document does not have N more print-stream pages, then the completion of the
- 6360 impression is controlled by the "multiple-document-handling" attribute as described in section 4.2.4;
- 6361 when the value of this attribute is 'single-document' or 'single-document-new-sheet', the print-stream
- 6362 pages of document data from subsequent documents is used to complete the impression.
- 6363

6364 The size(scaling), position(translation) and rotation of the print-stream pages on the impression is

6365 implementation defined. Note that during this process the print-stream pages may be rendered to a

6366 form suitable for placing on the impression; this rendering is controlled by the values of the "printer-

6367 resolution" and "print-quality" attributes as described in sections 4.2.12 and 4.2.13. In the case N=1,

6368 the impression is nearly the same as the print-stream page; the differences would only be in the size,

6369 position and rotation of the print-stream page and/or any decoration, such as a frame to the page,

6370 that is added by the implementation.

6371

- 6372 4. The collection of impressions is placed, in sequence, onto sides of the media sheets. This placement
- 6373 is controlled by the "sides" attribute and the orientation of the print-stream page, as described in
- 6374 section 4.2.8. The orientation of the print-stream pages affects the orientation of the impression; for
- 6375 example, if "number-up" equals 2, then, typically, two portrait print-stream pages become one
- 6376 landscape impression. Note that the placement of impressions onto media sheets is also controlled
- 6377 by the "multiple-document-handling" attribute as described in section 4.2.4.
- 6378
- 6379 5. The "copies" and "multiple-document-handling" attributes are used to determine how many copies of
- 6380 each media instance are created and in what order. See sections 4.2.5 and 4.2.4 for the details.
- 6381
- 6382 6. When the correct number of copies are created, the media instances are finished according to the
- 6383 values of the "finishings" attribute as described in 4.2.6. Note that sometimes finishing operations
- 6384 may require manual intervention to perform the finishing operations on the copies, especially
- 6385 uncollated copies. This document allows any or all of the processing steps to be performed
- 6386 automatically or manually at the discretion of the Printer object.

## 6387 16. APPENDIX E: Generic Directory Schema

6388 This section defines a generic schema for an entry in a directory service. A directory service is a means by

6389 which service users can locate service providers. In IPP environments, this means that IPP Printers can be

6390 registered (either automatically or with the help of an administrator) as entries of type printer in the

6391 directory using an implementation specific mechanism such as entry attributes, entry type fields, specific

6392 branches, etc. IPP clients can search or browse for entries of type printer. Clients use the directory service



6393 to find entries based on naming, organizational contexts, or filtered searches on attribute values of entries.  
 6394 For example, a client can find all printers in the "Local Department" context. Authentication and  
 6395 authorization are also often part of a directory service so that an administrator can place limits on end users  
 6396 so that they are only allowed to find entries to which they have certain access rights. IPP itself does not  
 6397 require any specific directory service protocol or provider.

6398 Note: Some directory implementations allow for the notion of "aliasing". That is, one directory entry object  
 6399 can appear as multiple directory entry object with different names for each object. In each case, each alias  
 6400 refers to the same directory entry object which refers to a single IPP Printer object.

6401 The generic schema is a subset of IPP Printer Job Template and Printer Description attributes (sections 4.2  
 6402 and 4.4). These attributes are identified as either RECOMMENDED or OPTIONAL for the directory entry  
 6403 itself. This conformance labeling is NOT the same conformance labeling applied to the attributes of IPP  
 6404 Printers objects. The conformance labeling in this Appendix is intended to apply to directory templates and  
 6405 to IPP Printer implementations that subscribe by adding one or more entries to a directory.  
 6406 RECOMMENDED attributes SHOULD be associated with each directory entry. OPTIONAL attributes  
 6407 MAY be associated with the directory entry (if known or supported). In addition, all directory entry  
 6408 attributes SHOULD reflect the current attribute values for the corresponding Printer object.

6409 The names of attributes in directory schema and entries SHOULD be the same as the IPP Printer attribute  
 6410 names as shown.

6411 In order to bridge between the directory service and the IPP Printer object, one of the RECOMMENDED  
 6412 directory entry attributes is the Printer object's "printer-uri-supported" attribute. The IPP client queries the  
 6413 "printer-uri-supported" attribute in the directory entry and then addresses the IPP Printer object using one of  
 6414 its URIs. The "uri-security-supported" attribute identifies the protocol (if any) used to secure a channel.

6415 The following attributes define the generic schema for directory entries of type PRINTER:

6416	printer-uri-supported	RECOMMENDED	Section 4.4.1
6417	uri-authentication-supported	RECOMMENDED	Section 4.4.2
6418	uri-security-supported	RECOMMENDED	Section 4.4.3
6419	printer-name	RECOMMENDED	Section 4.4.4
6420	printer-location	RECOMMENDED	Section 4.4.5
6421	printer-info	OPTIONAL	Section 4.4.6
6422	printer-more-info	OPTIONAL	Section 4.4.7
6423	printer-make-and-model	RECOMMENDED	Section 4.4.9
6424	ipp-versions-supported	RECOMMENDED	Section 4.4.14
6425	multiple-document-jobs-supported	OPTIONAL	Section 4.4.16
6426	charset-supported	OPTIONAL	Section 4.4.18
6427	generated-natural-language-		
6428	supported	OPTIONAL	Section 4.4.20
6429	document-format-supported	RECOMMENDED	Section 4.4.22
6430	color-supported	RECOMMENDED	Section 4.4.26
6431	compression-supported	RECOMMENDED	Section 4.4.32
6432	pages-per-minute	OPTIONAL	Section 4.4.36

6433	pages-per-minute-color	OPTIONAL	Section 4.4.37
6434			
6435	finishings-supported	OPTIONAL	Section 4.2.6
6436	number-up-supported	OPTIONAL	Section 4.2.7
6437	sides-supported	RECOMMENDED	Section 4.2.8
6438	media-supported	RECOMMENDED	Section 4.2.11
6439	printer-resolution-supported	OPTIONAL	Section 4.2.12
6440	print-quality-supported	OPTIONAL	Section 4.2.13



6441

## 6442 **17. APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model** 6443 **and Semantics" Documents**

6444 This Appendix is divided into two lists that summarize the differences between IPP/1.1 (this document) and  
6445 IPP/1.0 [RFC2566]. The section numbers refer to the numbers in this document which in some cases have  
6446 changed from RFC 2566. When a change affects multiple sections, the item is listed once in the order of  
6447 the first section affected and the remaining affected section numbers are indicated.

6448 The first list contains extensions and clarifications and the second list contains changes in semantics or  
6449 conformance. However, client and IPP object implementations of IPP/1.0 may implement any of the  
6450 extensions and clarifications in this document.

6451 The following extensions and clarifications have been incorporated into this document:

- 6452 1. Section 2.1 - clarified that the term "client" can be either contained in software controlled by an end  
6453 user or a part of a print server that controls devices.
- 6454 2. Section 2 - clarified that the term "IPP object" and "Printer object" can either be embedded in a  
6455 device object or part of a print server that accepts IPP requests.
- 6456 3. Section 2.4 - added the description of the new "uri-authentication-supported" Printer Description  
6457 attribute.
- 6458 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  
6459 have their own status code.
- 6460 5. Section 3.1.3 - clarified that multiple occurrences of the same attribute in an attribute group is mal-  
6461 formed. An IPP Printer MAY reject the request or choose one of the attributes.
- 6462 6. Section 3.1.6 - reorganized this section into sub-sections to separately describe "status-code",  
6463 "status-message", "detailed-status-message", and "document-access-error" attributes.
- 6464 7. Section 3.1.6.1 - clarified the error status codes and their relationship to operation attributes.
- 6465 8. Section 3.1.6.3 - Added the OPTIONAL "detailed-status-message (text(MAX))" operation attribute  
6466 to provide additional more detailed information about a response.
- 6467 9. Section 3.1.6.4 and 3.2.2 - Added the OPTIONAL "document-access-error (text(MAX))" operation  
6468 attribute for use with Print-URI and Send-URI responses.
- 6469 10. Sections 3.1.7 - Added this new section to clarify returning Unsupported Attributes for all  
6470 operations, including only returning attributes that were in the request. Moved the text from section  
6471 3.2.1.2 Unsupported Attributes to this section.
- 6472 11. Sections 3.1.7 and 4.1 - clarified the encoding of the "out-of-band" 'unsupported' and 'unknown'  
6473 values.
- 6474 12. Section 3.1.8 - clarified that only the version number parameter will be carried forward into future  
6475 major or minor versions of the protocol.
- 6476 13. Section 3.1.8 - relaxed the requirements to increment the major version number in future versions of  
6477 the Model and Semantics document.
- 6478 14. 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  
6479 a Create-Job operation.

- 6480 15. Section 3.1.9 - clarified that a non-spooling Printer MAY accept zero or more subsequent jobs while  
6481 processing a job and flow control them down. Subsequent create requests are rejected with the  
6482 'server-error-busy' error status.
- 6483 16. Section 3.2.1.1 - clarified the validation of the "compression" operation attribute and its relationship  
6484 to the validation of the "document-format" attribute and returning Unsupported Attributes.
- 6485 17. Sections 3.2.1.1, 4.3.8, 13.1.4.16, and 13.1.4.17 - added the 'client-error-compression-not-  
6486 supported', 'client-error-compression-error' status codes and the 'unsupported-compression' and  
6487 'compression-error' job-state-reasons.
- 6488 18. Sections 3.2.1.1 and 4.3.8 - added 'unsupported-document-format' and 'document-format-error' job-  
6489 state-reasons.
- 6490 19. Sections 3.2.2, 4.3.8 and 13.1.4.19 - added 'client-error-document-access-error' status code and  
6491 'document-access-error' job state reason.
- 6492 20. Section 3.2.5.2 and 3.2.6.2 - clarified that the Unsupported Attributes group MUST NOT include  
6493 attributes not requested in the Get-Printer-Attributes request.
- 6494 21. Section 3.2.6 - clarified that "limit" takes precedence over "which-jobs" and "my-jobs".
- 6495 22. Section 3.2.6.2 - clarified that Get-Jobs returns 'successful-ok' when no jobs to return.
- 6496 23. Sections 3.2.7, 3.2.8, and 3.2.9 - added the OPTIONAL Pause-Printer, Resume-Printer, and Purge-  
6497 Jobs operations
- 6498 24. Section 3.3.1 - clarified that the authorization required for a Send-Document request MUST be the  
6499 same user as the Create-Job or an operator.
- 6500 25. Section 3.3.1.1 - clarified that a Create-Job Send-Document with "last-document" = 'true' and no  
6501 data is not an error; its a job with no documents.
- 6502 26. Sections 3.3.5, 3.3.6, and 3.3.7 - added the OPTIONAL Hold-Job, Release-Job, and Restart-Job  
6503 operations. Clarified the Restart-Job operation so that the Printer MUST re-fetch any documents  
6504 passed by-reference (Print-URI or Send-URI).
- 6505 27. Section 4.1 - clarified that the encoding of the out-of-band values are specified in the Encoding and  
6506 Transport" document.
- 6507 28. Sections 4.1.1 and 4.1.2 - clarified that the maximum 'text' and 'name' values of 1023 and 255 are  
6508 for the 'textWithoutLanguage' portion of the 'textWithLanguage' form, so that the maximum number  
6509 of octets for the actual text and name data is the same for the without and with language forms; the  
6510 'naturalLanguage' part is in addition.
- 6511 29. Section 4.1.9.1 - clarified that 'application/octet-stream' auto-sensing can happen at create request  
6512 time and/or job/document processing time.
- 6513 30. Section 4.1.14 - clarified that the localization of dateTime by the client includes the time zone.
- 6514 31. Section 4.2 - clarified that xxx-supported have multiple keywords and/or names by adding  
6515 parentheses to the table to give: (1setOf (type3 keyword | name))
- 6516 32. Section 4.2.2 - added the 'indefinite' keyword value to the "job-hold-until" attribute for use with the  
6517 create operations and Hold-Job and Restart-Job operations.
- 6518 33. Section 4.2.6 - added more enum values to the "finishings" Job Template attribute.
- 6519 34. Section 4.3.7 - added that a forwarding server that cannot get any job state MAY return the job's  
6520 state as 'completed', provided that it also return the new 'queued-in-device' job state reason.
- 6521 35. Section 4.3.7.2 - added the Partitioning of Job States section to clarify the concepts of Job  
6522 Retention, Job History, and Job Removal.
- 6523 36. Section 4.3.8 - added 'job-data-insufficient' job state reason to indicate whether sufficient data has  
6524 arrived for the document to start to be processed.

- 6525 37. Section 4.3.8 - added 'document-access-error' job state reason to indicate an access error of any kind.
- 6526 38. Section 4.3.8 - added 'job-queued-for-marker' job state reason to indicate whether the job has
- 6527 completed some processing and is waiting for the marker.
- 6528 39. Section 4.3.8 - added 'unsupported-compression' and 'compression-error' job state reasons to
- 6529 indicate compression not supported or compression processing error after the create has been
- 6530 accepted.
- 6531 40. Section 4.3.8 - added 'unsupported-document-format' and 'document-format-error' job state reasons
- 6532 to indicate document not supported or document format processing error after the create has been
- 6533 accepted.
- 6534 41. Section 4.3.8 - added 'queued-in-device' job state reason to indicate that a job as been forwarded to a
- 6535 print system or device that does not provide any job status.
- 6536 42. Section 4.3.10 - added "job-detailed-status-messages (1setOf text(MAX)) for returning detailed
- 6537 error messages.
- 6538 43. Section 4.3.11 - added the "job-document-access-errors (1setOf text(MAX))
- 6539 44. Section 4.3.14.2 - clarified that the time recorded is the first time processing since the create
- 6540 operation or the Restart-Job operation.
- 6541 45. Section 4.3.14.2 and 4.3.14.3 - clarified that the out-of-band value 'no-value' is returned if the job
- 6542 has not started processing or has not completed, respectively.
- 6543 46. Section 4.3.14 - Added the OPTIONAL "date-time-at-creation", "date-time-at-processing", and
- 6544 "date-time-at-completed" Event Time Job Description attributes
- 6545 47. Section 4.4.3 - added the 'tls' value to "uri-security-supported" attribute.
- 6546 48. Section 4.4.3 - clarified "uri-security-supported" is orthogonal to Client Authentication so that 'none'
- 6547 does not exclude Client Authentication.
- 6548 49. Section 4.4.11 - simplified the "printer-state" descriptions while generalizing to allow high end
- 6549 devices that interpret one or more jobs while marking another. Indicated that 'spool-area-full' and
- 6550 'stopped-partly' "printer-state-reasons" may be used to provide further state information.
- 6551 50. Section 4.4.12 - added the 'moving-to-paused' keyword value to the "printer-state-reasons" attribute
- 6552 for use with the Pause-~~PrinterJob~~ operation.
- 6553 51. Section 4.4.12 - replaced the duplicate 'marker-supply-low' keyword with the missing 'toner-empty'
- 6554 keyword for the "printer-state-reasons" attribute. (This correction was also made before RFC 2566
- 6555 was published).
- 6556 52. Section 4.4.12 - clarified 'spool-area-full' "printer-state-reasons" to include non-spooling printers to
- 6557 indicate when it can and cannot accept another job.
- 6558 53. Section 4.4.15 - added the enum values to the "operations-supported" attribute for the new
- 6559 operations. Clarified that the values of this attribute are encoded as any enum, namely 32-bit values.
- 6560 54. Section 4.4.30 - clarified that the dateTime value of "printer-current-time" is on a "best efforts
- 6561 basis". If a proper date-time cannot be obtained, the implementation returns the 'no-value' out-of-
- 6562 band value. Also clarified that the time zone NEED NOT be the time zone that the people near the
- 6563 device use and that the client SHOULD display the dateTime attributes in the user's local time.
- 6564 55. Sections 4.4.36 and 4.4.37 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-
- 6565 color" Printer Description attributes.
- 6566 56. Section 5.1 - clarified that the client conformance requirements apply to clients controlled by an end
- 6567 user and clients in servers.
- 6568 57. Section 5.1 - clarified that any response MAY contain additional attribute groups, attributes,
- 6569 attribute syntaxes, or attribute values.

- 6570 58. Section 5.1 - clarified that a client SHOULD do its best to prevent a channel from being closed by a  
6571 lower layer when the channel is flow controlled off by the IPP Printer.
- 6572 59. Section 5.2 - clarified that the IPP object requirements apply to objects embedded in devices or that  
6573 are parts of servers.
- 6574 60. Section 5.2.2 - clarified that IPP objects MAY return operation responses that contain attribute  
6575 groups, attribute names, attribute syntaxes, attribute values, and status codes that are extensions to  
6576 this standard.
- 6577 61. Section 6 - changed the terminology of "private extensions" to "vendor extensions" and indicated  
6578 that they are registered with IANA along with IETF standards track extensions.
- 6579 62. Section 6.7 - inserted this section on registering out-of-band attribute values with IANA as  
6580 extensions.
- 6581 63. Section 8.3 - clarified the use of URIs for each Client Authentication mechanism.
- 6582 64. Section 8.5 - added the security discussion around the new operator/administrator operations.
- 6583 65. Section 13.1.4.16 - added client-error-compression-not-supported (0x040F)
- 6584 66. Section 13.1.4.17 - added client-error-compression-error (0x0410)
- 6585 67. Section 13.1.4.18 - added client-error-document-format-error (0x0411)
- 6586 68. Section 13.1.4.19 - added client-error-document-access-error (0x0412)
- 6587 69. Section 13.1.5.10 - added server-error-multiple-document-jobs-not-supported (0x0509)
- 6588 70. Section 14 - added 'a-white', 'b-white', 'c-white', 'd-white', and 'e-white' and clarified that the existing  
6589 'a', 'b', 'c', 'd', and 'e' values are size values. Added American, Japanese, and European Engineering  
6590 sizes, filled out -transparent and -translucent media names and drawings for the synchro cut sizes.
- 6591 71. Section 16 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-color" Printer  
6592 attributes to the Directory schema.
- 6593 72. Section 16 - added OPTIONAL "multiple-document-jobs-supported" to the Directory schema.
- 6594 73. Section 16 - added RECOMMENDED "uri-authentication-supported", "ipp-versions-supported",  
6595 and "compression-supported" to the Directory schema.

6596 The following changes in semantics and/or conformance have been incorporated into this document:

- 6597 1. Section 3.1.8, 5.2.4, and 13.1.5.4 - Clients and IPP objects MUST support version 1.1  
6598 conformance requirements. It is recommended that they interoperate with 1.0. Also clarified  
6599 that IPP Printers MUST accept '1.1' requests. It is recommended that they also accept '1.x'  
6600 requests.
- 6601 2. Section 3.2.1.1 and section 4.4.32 - changed the "compression" operation and the "compression-  
6602 supported" Printer Description attribute from OPTIONAL to REQUIRED.
- 6603 3. Sections 3.2.1.2 and 4.3.8 - changed "job-state-reasons" from RECOMMENDED to REQUIRED,  
6604 so that "job-state-reasons" MUST be returned in create operation responses.
- 6605 4. Sections 3.2.4, 3.3.1, 4.4.16, and 16 - changed Create-Job/Send-Document so that they MAY be  
6606 implemented while only supporting one document jobs. Added the "multiple-document-jobs-  
6607 supported" boolean Printer Description attribute to indicate whether Create-Job/Send-  
6608 Document support multiple document jobs or not. Added to the Directory schema.
- 6609 5. Section 4.1.9 - deleted 'text/plain; charset=iso-10646-ucs-2', since binary is not legal with the 'text'  
6610 type.
- 6611 6. Section 4.2.4 - indicated that the "multiple-document-handling" Job Template attribute MUST be  
6612 supported with at least one value if the Printer supports multiple documents per job

- 6613 7. Section 4.3.7.2 - indicated that the 'job-restartable' job state reason SHOULD be supported if the  
6614 Restart-Job operation is supported.
- 6615 8. Section 4.3.8 - changed "job-state-reasons" from RECOMMENDED to REQUIRED.
- 6616 9. Section 4.3.8 - clarified the conformance of the values of the "job-state-reasons" attribute by  
6617 copying conformance requirements from other sections of the document so that it is clear from  
6618 reading the definition of "job-state-reasons" which values MUST or SHOULD be supported.  
6619 The 'none', 'unsupported-compression', and 'unsupported-document-format' values MUST be  
6620 supported. The 'job-hold-until-specified' SHOULD be specified if the "job-hold-until" Job  
6621 Template is supported. The following values SHOULD be supported: 'job-canceled-by-user',  
6622 'aborted-by-system', and 'job-completed-successfully'. The 'job-canceled-by-operator' SHOULD  
6623 be supported if the implementation permits canceling by other than the job owner. The 'job-  
6624 canceled-at-device' SHOULD be supported if the device supports canceling jobs at the console.  
6625 The 'job-completed-with-warnings' SHOULD be supported, if the implementation detects  
6626 warnings. The 'job-completed-with-errors' SHOULD be supported if the implementation  
6627 detects errors. The 'job-restartable' SHOULD be supported if the Restart-Job operation is  
6628 supported.
- 6629 10. Section 4.3.14 - changed the "time-at-creation", "time-at-processing", and "time-at-completed"  
6630 Event Time Job Description attributes from OPTIONAL to REQUIRED.
- 6631 11. Section 4.3.14.4 - added the REQUIRED "job-printer-up-time (integer(1:MAX))" Job Description  
6632 attribute as an alias for "printer-up-time" to reduce number of operations to get job times.
- 6633 12. Section 4.4.2 - added the REQUIRED "uri-authentication-supported (1setOf type2 keyword)"  
6634 Printer Description attribute to describe the Client Authentication used by each Printer URI.
- 6635 13. Section 4.4.12 - changed "printer-state-reasons" Printer Description attribute from OPTIONAL to  
6636 REQUIRED.
- 6637 14. Section 4.4.12 - changed 'paused' value of "printer-state-reasons" to MUST if Pause-Printer  
6638 operation is supported.
- 6639 15. Section 4.4.14 - added the REQUIRED "ipp-versions-supported (1setOf keyword)" Printer  
6640 Description attribute, since IPP/1.1 Printers do not have to support version '1.0' conformance  
6641 requirements. Section 4.4.16 - added the "multiple-document-jobs-supported (boolean)" Printer  
6642 Description attribute so that a client can tell whether a Printer that supports Create-Job/Send-  
6643 Document supports multiple document jobs or not. This attribute is REQUIRED if the Create-  
6644 Job operation is supported.
- 6645 16. Section 4.4.24 - changed the "queued-job-count" Printer Description attribute from  
6646 RECOMMENDED to REQUIRED.
- 6647 17. Section 4.4.32 - changed "compression-supported (1setOf type3 keyword)" Printer Description  
6648 attribute from OPTIONAL to REQUIRED.
- 6649 18. Section 5.1 - changed the client security requirements from RECOMMENDED non-standards  
6650 track SSL3 to MUST support Client Authentication as defined in the IPP/1.1 Encoding and  
6651 Transport document [IPP-PRO]. A client SHOULD support Operation Privacy and Server  
6652 Authentication as defined in the IPP/1.1 Encoding and Transport document [IPP-PRO].
- 6653 19. Section 5.2.7 - changed the IPP object security requirements from OPTIONAL non-standards track  
6654 SSL3 to SHOULD contain support for Client Authentication as defined in the IPP/1.1 Encoding  
6655 and Transport document [IPP-PRO]. A Printer implementation MAY allow an administrator to  
6656 configure the Printer so that all, some, or none of the users are authenticated. An IPP Printer  
6657 implementation SHOULD contain support for Operation Privacy and Server Authentication as



6658 defined in the IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer implementation  
6659 MAY allow an administrator to configure the degree of support for Operation Privacy and  
6660 Server Authentication. Security MUST NOT be compromised when the client supplies a lower  
6661 version-number in a request.

6662 See also the "IPP/1.1 Encoding and Transport" [IPP-PRO] document for differences between IPP/1.0  
6663 [RFC2565] and IPP/1.1 [IPP-PRO].

## 6664 **18. Full Copyright Statement**

6665 Copyright (C) The Internet Society (1999). All Rights Reserved.

6666 This document and translations of it may be copied and furnished to others, and derivative works that  
6667 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and  
6668 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and  
6669 this paragraph are included on all such copies and derivative works. However, this document itself may not  
6670 be modified in any way, such as by removing the copyright notice or references to the Internet Society or  
6671 other Internet organizations, except as needed for the purpose of developing Internet standards in which  
6672 case the procedures for copyrights defined in the Internet Standards process must be followed, or as  
6673 required to translate it into languages other than English.

6674 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its  
6675 successors or assigns.

6676 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET  
6677 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,  
6678 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE  
6679 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED  
6680 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

6681