

1 INTERNET-DRAFT  
2 draft-ietf-ipp-model-v11-07.txt

R. deBry  
Utah Valley State College  
T. Hastings (editor)  
Xerox Corporation  
R. Herriot  
Xerox Corporation  
S. Isaacson  
Novell, Inc.  
P. Powell  
Astart Technologies  
May 22, 2000

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

## Table of Contents

66

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.....	15
72	2.3	Object Relationships.....	16
73	2.4	Object Identity .....	17
74	3.	IPP Operations .....	19
75	3.1	Common Semantics.....	20
76	3.1.1	Required Parameters.....	20
77	3.1.2	Operation IDs and Request IDs .....	21
78	3.1.3	Attributes .....	21
79	3.1.4	Character Set and Natural Language Operation Attributes .....	22
80	3.1.4.1	Request Operation Attributes .....	23
81	3.1.4.2	Response Operation Attributes .....	26
82	3.1.5	Operation Targets .....	27
83	3.1.6	Operation Response Status Codes and Status Messages .....	28
84	3.1.6.1	"status-code" (type2 enum) .....	28
85	3.1.6.2	"status-message" (text(255)) .....	29
86	3.1.6.3	"detailed-status-message" (text(MAX)).....	29
87	3.1.6.4	"document-access-error" (text(MAX)).....	29
88	3.1.7	Unsupported Attributes .....	30
89	3.1.8	Versions .....	31
90	3.1.9	Job Creation Operations .....	33
91	3.2	Printer Operations.....	35
92	3.2.1	Print-Job Operation .....	35
93	3.2.1.1	Print-Job Request.....	35
94	3.2.1.2	Print-Job Response.....	39
95	3.2.2	Print-URI Operation .....	41
96	3.2.3	Validate-Job Operation.....	41
97	3.2.4	Create-Job Operation.....	42
98	3.2.5	Get-Printer-Attributes Operation.....	42
99	3.2.5.1	Get-Printer-Attributes Request .....	43
100	3.2.5.2	Get-Printer-Attributes Response.....	44
101	3.2.6	Get-Jobs Operation .....	45
102	3.2.6.1	Get-Jobs Request.....	45
103	3.2.6.2	Get-Jobs Response .....	47
104	3.2.7	Pause-Printer Operation.....	48
105	3.2.7.1	Pause-Printer Request .....	49
106	3.2.7.2	Pause-Printer Response.....	49
107	3.2.8	Resume-Printer Operation .....	50

108	3.2.9	Purge-Jobs Operation .....	51
109	3.3	Job Operations .....	51
110	3.3.1	Send-Document Operation .....	51
111	3.3.1.1	Send-Document Request .....	52
112	3.3.1.2	Send-Document Response .....	54
113	3.3.2	Send-URI Operation .....	54
114	3.3.3	Cancel-Job Operation .....	54
115	3.3.3.1	Cancel-Job Request .....	55
116	3.3.3.2	Cancel-Job Response .....	56
117	3.3.4	Get-Job-Attributes Operation .....	56
118	3.3.4.1	Get-Job-Attributes Request .....	57
119	3.3.4.2	Get-Job-Attributes Response .....	57
120	3.3.5	Hold-Job Operation .....	58
121	3.3.5.1	Hold-Job Request .....	59
122	3.3.5.2	Hold-Job Response .....	60
123	3.3.6	Release-Job Operation .....	60
124	3.3.7	Restart-Job Operation .....	61
125	3.3.7.1	Restart-Job Request .....	62
126	3.3.7.2	Restart-Job Response .....	63
127	4.	Object Attributes .....	63
128	4.1	Attribute Syntaxes .....	64
129	4.1.1	'text' .....	64
130	4.1.1.1	'textWithoutLanguage' .....	65
131	4.1.1.2	'textWithLanguage' .....	65
132	4.1.2	'name' .....	66
133	4.1.2.1	'nameWithoutLanguage' .....	66
134	4.1.2.2	'nameWithLanguage' .....	67
135	4.1.2.3	Matching 'name' attribute values .....	67
136	4.1.3	'keyword' .....	68
137	4.1.4	'enum' .....	69
138	4.1.5	'uri' .....	69
139	4.1.6	'uriScheme' .....	69
140	4.1.7	'charset' .....	70
141	4.1.8	'naturalLanguage' .....	70
142	4.1.9	'mimeType' .....	71
143	4.1.9.1	Application/octet-stream -- Auto-Sensing the document format .....	71
144	4.1.10	'octetString' .....	72
145	4.1.11	'boolean' .....	72
146	4.1.12	'integer' .....	72
147	4.1.13	'rangeOfInteger' .....	72
148	4.1.14	'dateTime' .....	73
149	4.1.15	'resolution' .....	73
150	4.1.16	'1setOf X' .....	73
151	4.2	Job Template Attributes .....	73

152	4.2.1	job-priority (integer(1:100)) .....	77
153	4.2.2	job-hold-until (type3 keyword   name (MAX)) .....	78
154	4.2.3	job-sheets (type3 keyword   name(MAX)) .....	78
155	4.2.4	multiple-document-handling (type2 keyword) .....	79
156	4.2.5	copies (integer(1:MAX)) .....	80
157	4.2.6	finishings (1setOf type2 enum) .....	80
158	4.2.7	page-ranges (1setOf rangeOfInteger (1:MAX)) .....	82
159	4.2.8	sides (type2 keyword).....	83
160	4.2.9	number-up (integer(1:MAX)).....	83
161	4.2.10	orientation-requested (type2 enum).....	84
162	4.2.11	media (type3 keyword   name(MAX)).....	85
163	4.2.12	printer-resolution (resolution).....	85
164	4.2.13	print-quality (type2 enum) .....	85
165	4.3	Job Description Attributes.....	86
166	4.3.1	job-uri (uri) .....	88
167	4.3.2	job-id (integer(1:MAX)).....	88
168	4.3.3	job-printer-uri (uri) .....	88
169	4.3.4	job-more-info (uri).....	89
170	4.3.5	job-name (name(MAX)).....	89
171	4.3.6	job-originating-user-name (name(MAX)) .....	89
172	4.3.7	job-state (type1 enum) .....	89
173	4.3.7.1	Forwarding Servers .....	92
174	4.3.7.2	Partitioning of Job States .....	92
175	4.3.8	job-state-reasons (1setOf type2 keyword) .....	93
176	4.3.9	job-state-message (text(MAX)).....	96
177	4.3.10	job-detailed-status-messages (1setOf text(MAX)) .....	96
178	4.3.11	job-document-access-errors (1setOf text(MAX)).....	96
179	4.3.12	number-of-documents (integer(0:MAX)).....	96
180	4.3.13	output-device-assigned (name(127)) .....	97
181	4.3.14	Event Time Job Description Attributes .....	97
182	4.3.14.1	time-at-creation (integer(MIN:MAX)).....	97
183	4.3.14.2	time-at-processing (integer(MIN:MAX)).....	98
184	4.3.14.3	time-at-completed (integer(MIN:MAX)).....	98
185	4.3.14.4	job-printer-up-time (integer(1:MAX)).....	98
186	4.3.14.5	date-time-at-creation (dateTime) .....	98
187	4.3.14.6	date-time-at-processing (dateTime) .....	98
188	4.3.14.7	date-time-at-completed (dateTime).....	98
189	4.3.15	number-of-intervening-jobs (integer(0:MAX)) .....	98
190	4.3.16	job-message-from-operator (text(127)) .....	99
191	4.3.17	Job Size Attributes.....	99
192	4.3.17.1	job-k-octets (integer(0:MAX)).....	99
193	4.3.17.2	job-impressions (integer(0:MAX)) .....	99
194	4.3.17.3	job-media-sheets (integer(0:MAX)).....	100
195	4.3.18	Job Progress Attributes.....	100
196	4.3.18.1	job-k-octets-processed (integer(0:MAX)).....	100

197	4.3.18.2	job-impressions-completed (integer(0:MAX)) .....	100
198	4.3.18.3	job-media-sheets-completed (integer(0:MAX)).....	100
199	4.3.19	attributes-charset (charset).....	101
200	4.3.20	attributes-natural-language (naturalLanguage).....	101
201	4.4	Printer Description Attributes.....	101
202	4.4.1	printer-uri-supported (1setOf uri).....	103
203	4.4.2	uri-authentication-supported (1setOf type2 keyword).....	104
204	4.4.3	uri-security-supported (1setOf type2 keyword).....	104
205	4.4.4	printer-name (name(127)).....	105
206	4.4.5	printer-location (text(127)).....	106
207	4.4.6	printer-info (text(127)) .....	106
208	4.4.7	printer-more-info (uri) .....	106
209	4.4.8	printer-driver-installer (uri) .....	106
210	4.4.9	printer-make-and-model (text(127)).....	106
211	4.4.10	printer-more-info-manufacturer (uri).....	106
212	4.4.11	printer-state (type1 enum).....	107
213	4.4.12	printer-state-reasons (1setOf type2 keyword).....	107
214	4.4.13	printer-state-message (text(MAX)) .....	109
215	4.4.14	ipp-versions-supported (1setOf type2 keyword) .....	109
216	4.4.15	operations-supported (1setOf type2 enum) .....	110
217	4.4.16	multiple-document-jobs-supported (boolean) .....	111
218	4.4.17	charset-configured (charset) .....	111
219	4.4.18	charset-supported (1setOf charset) .....	111
220	4.4.19	natural-language-configured (naturalLanguage).....	111
221	4.4.20	generated-natural-language-supported (1setOf naturalLanguage).....	111
222	4.4.21	document-format-default (mimeMediaType) .....	112
223	4.4.22	document-format-supported (1setOf mimeMediaType).....	112
224	4.4.23	printer-is-accepting-jobs (boolean).....	112
225	4.4.24	queued-job-count (integer(0:MAX)) .....	112
226	4.4.25	printer-message-from-operator (text(127)).....	112
227	4.4.26	color-supported (boolean).....	113
228	4.4.27	reference-uri-schemes-supported (1setOf uriScheme) .....	113
229	4.4.28	pdl-override-supported (type2 keyword) .....	113
230	4.4.29	printer-up-time (integer(1:MAX)).....	113
231	4.4.30	printer-current-time (dateTime).....	114
232	4.4.31	multiple-operation-time-out (integer(1:MAX)).....	114
233	4.4.32	compression-supported (1setOf type3 keyword).....	115
234	4.4.33	job-k-octets-supported (rangeOfInteger(0:MAX)) .....	115
235	4.4.34	job-impressions-supported (rangeOfInteger(0:MAX)).....	115
236	4.4.35	job-media-sheets-supported (rangeOfInteger(0:MAX)).....	115
237	4.4.36	pages-per-minute (integer(0:MAX)) .....	115
238	4.4.37	pages-per-minute-color (integer(0:MAX)).....	115
239	5.	Conformance.....	116
240	5.1	Client Conformance Requirements .....	116

241	5.2	IPP Object Conformance Requirements.....	117
242	5.2.1	Objects.....	117
243	5.2.2	Operations.....	118
244	5.2.3	IPP Object Attributes.....	118
245	5.2.4	Versions.....	119
246	5.2.5	Extensions.....	119
247	5.2.6	Attribute Syntaxes.....	119
248	5.2.7	Security.....	120
249	5.3	Charset and Natural Language Requirements.....	120
250	6.	IANA Considerations.....	120
251	6.1	Typed 'keyword' and 'enum' Extensions.....	121
252	6.2	Attribute Extensibility.....	123
253	6.3	Attribute Syntax Extensibility.....	123
254	6.4	Operation Extensibility.....	124
255	6.5	Attribute Group Extensibility.....	124
256	6.6	Status Code Extensibility.....	124
257	6.7	Out-of-band Attribute Value Extensibility.....	125
258	6.8	Registration of MIME types/sub-types for document-formats.....	125
259	6.9	Registration of charsets for use in 'charset' attribute values.....	125
260	7.	Internationalization Considerations.....	125
261	8.	Security Considerations.....	129
262	8.1	Security Scenarios.....	129
263	8.1.1	Client and Server in the Same Security Domain.....	129
264	8.1.2	Client and Server in Different Security Domains.....	130
265	8.1.3	Print by Reference.....	130
266	8.2	URIs in Operation, Job, and Printer attributes.....	130
267	8.3	URIs for each authentication mechanisms.....	130
268	8.4	Restricted Queries.....	131
269	8.5	Operations performed by operators and system administrators.....	131
270	8.6	Queries on jobs submitted using non-IPP protocols.....	131
271	9.	References.....	132
272	10.	Author's Address.....	135
273	11.	Formats for IPP Registration Proposals.....	139
274	11.1	Type2 keyword attribute values registration.....	139
275	11.2	Type3 keyword attribute values registration.....	139
276	11.3	Type2 enum attribute values registration.....	140
277	11.4	Type3 enum attribute values registration.....	140
278	11.5	Attribute registration.....	140
279	11.6	Attribute Syntax registration.....	141
280	11.7	Operation registration.....	141

281	11.8	Attribute Group registration .....	141
282	11.9	Status code registration.....	142
283	11.10	Out-of-band Attribute Value registration .....	142
284	12.	APPENDIX A: Terminology.....	142
285	12.1	Conformance Terminology.....	143
286	12.1.1	NEED NOT .....	143
287	12.2	Model Terminology .....	143
288	12.2.1	Keyword .....	143
289	12.2.2	Attributes .....	143
290	12.2.2.1	Attribute Name.....	143
291	12.2.2.2	Attribute Group Name .....	143
292	12.2.2.3	Attribute Value.....	144
293	12.2.2.4	Attribute Syntax .....	144
294	12.2.3	Supports.....	144
295	12.2.4	print-stream page .....	145
296	12.2.5	impression.....	146
297	13.	APPENDIX B: Status Codes and Suggested Status Code Messages .....	146
298	13.1	Status Codes .....	147
299	13.1.1	Informational .....	147
300	13.1.2	Successful Status Codes .....	147
301	13.1.2.1	successful-ok (0x0000) .....	147
302	13.1.2.2	successful-ok-ignored-or-substituted-attributes (0x0001) .....	147
303	13.1.2.3	successful-ok-conflicting-attributes (0x0002) .....	147
304	13.1.3	Redirection Status Codes.....	148
305	13.1.4	Client Error Status Codes .....	148
306	13.1.4.1	client-error-bad-request (0x0400) .....	148
307	13.1.4.2	client-error-forbidden (0x0401) .....	148
308	13.1.4.3	client-error-not-authenticated (0x0402) .....	148
309	13.1.4.4	client-error-not-authorized (0x0403) .....	148
310	13.1.4.5	client-error-not-possible (0x0404) .....	149
311	13.1.4.6	client-error-timeout (0x0405) .....	149
312	13.1.4.7	client-error-not-found (0x0406) .....	149
313	13.1.4.8	client-error-gone (0x0407) .....	149
314	13.1.4.9	client-error-request-entity-too-large (0x0408) .....	150
315	13.1.4.10	client-error-request-value-too-long (0x0409) .....	150
316	13.1.4.11	client-error-document-format-not-supported (0x040A).....	150
317	13.1.4.12	client-error-attributes-or-values-not-supported (0x040B).....	150
318	13.1.4.13	client-error-uri-scheme-not-supported (0x040C).....	151
319	13.1.4.14	client-error-charset-not-supported (0x040D).....	151
320	13.1.4.15	client-error-conflicting-attributes (0x040E).....	151
321	13.1.4.16	client-error-compression-not-supported (0x040F).....	151
322	13.1.4.17	client-error-compression-error (0x0410) .....	151
323	13.1.4.18	client-error-document-format-error (0x0411).....	152



324	13.1.4.19	client-error-document-access-error (0x0412) .....	152
325	13.1.5	Server Error Status Codes.....	152
326	13.1.5.1	server-error-internal-error (0x0500).....	152
327	13.1.5.2	server-error-operation-not-supported (0x0501) .....	152
328	13.1.5.3	server-error-service-unavailable (0x0502).....	152
329	13.1.5.4	server-error-version-not-supported (0x0503).....	153
330	13.1.5.5	server-error-device-error (0x0504) .....	153
331	13.1.5.6	server-error-temporary-error (0x0505).....	153
332	13.1.5.7	server-error-not-accepting-jobs (0x0506) .....	154
333	13.1.5.8	server-error-busy (0x0507) .....	154
334	13.1.5.9	server-error-job-canceled (0x0508).....	154
335	13.1.5.10	server-error-multiple-document-jobs-not-supported (0x0509).....	154
336	13.2	Status Codes for IPP Operations .....	155
337	14.	APPENDIX C: "media" keyword values.....	157
338	15.	APPENDIX D: Processing IPP Attributes .....	175
339	15.1	Fidelity.....	176
340	15.2	Page Description Language (PDL) Override.....	177
341	15.3	Using Job Template Attributes During Document Processing.....	178
342	16.	APPENDIX E: Generic Directory Schema .....	179
343	17.	APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model and Semantics" Documents	182
344	18.	Full Copyright Statement.....	187
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 primarily on end user functionality  
349 with a few administrative operations included. This document is just one of a suite of documents that fully  
350 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 [HTPP] [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)

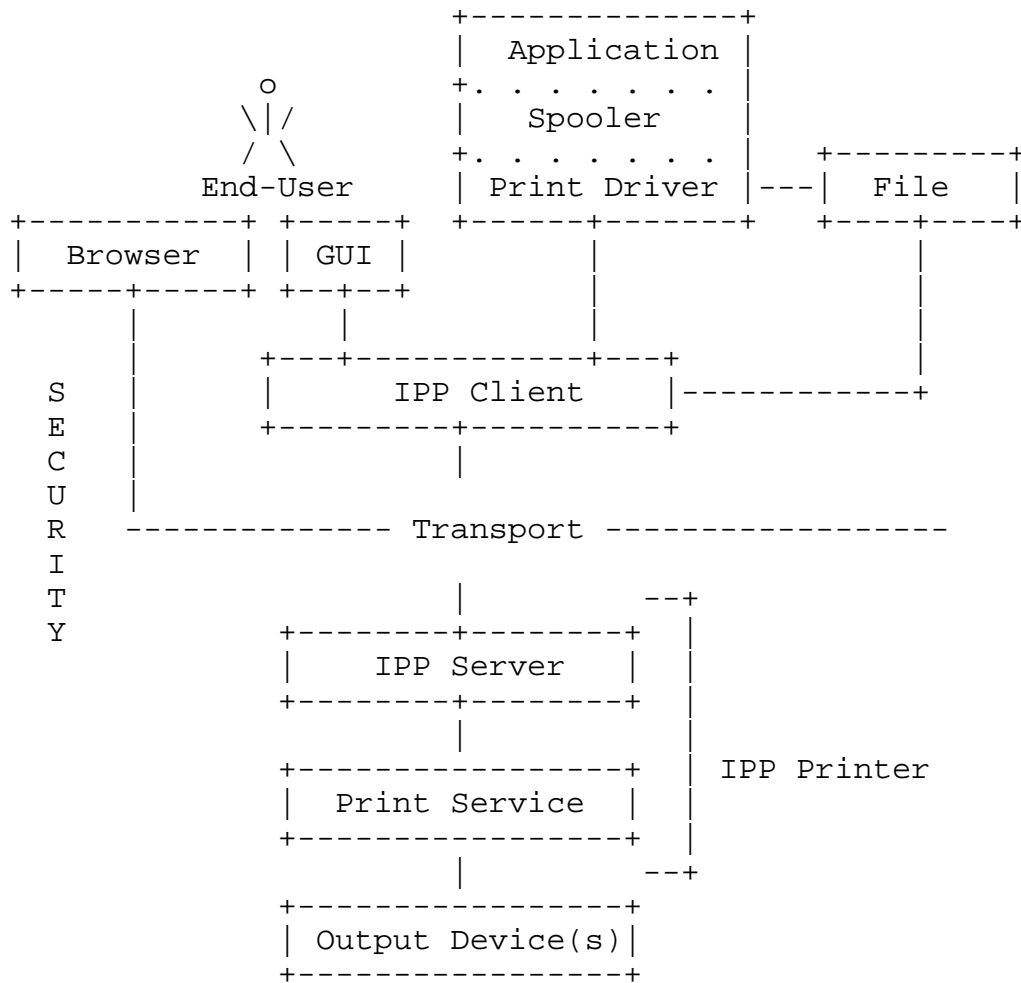
405

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 additional 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).

527 Legend:

528

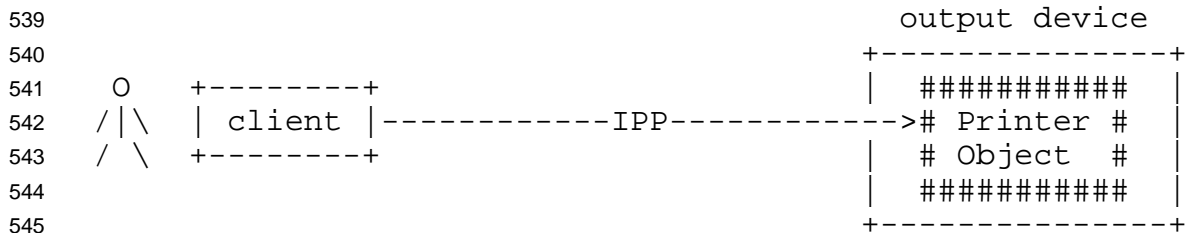
529 ##### indicates a Printer object which is  
 530 either embedded in an output device or is  
 531 hosted in a server. The Printer object  
 532 might or might not be capable of queuing/spooling.

533  
 534 any indicates any network protocol or direct  
 535 connect, including IPP

536

537

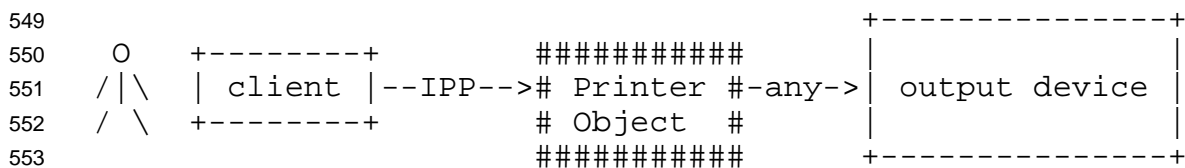
538 embedded printer:



546

547

548 hosted printer:



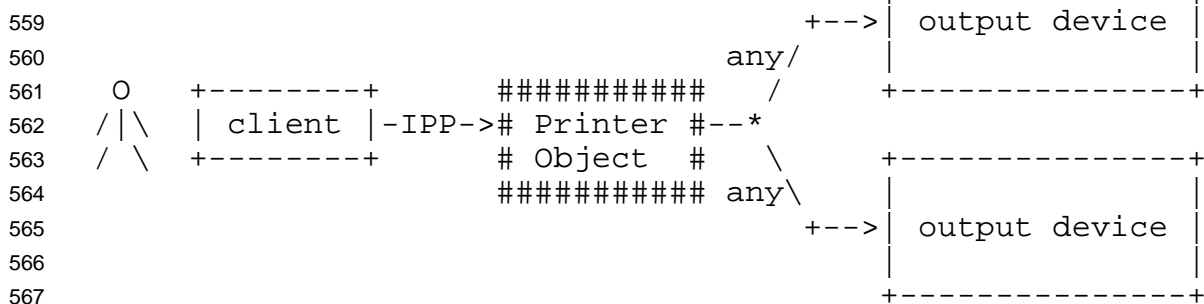
554

555

556

557

558 fan out:



568

569

570 **2.2 Job Object**

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

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

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

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

582

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

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

588

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

## 592 **2.3 Object Relationships**

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

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

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

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



## 608 2.4 Object Identity

609 All Printer and Job objects are identified by a Uniform Resource Identifier (URI) [RFC2396] so that they  
610 can be persistently and unambiguously referenced. Since every URL is a specialized form of a URI, even  
611 though the more generic term URI is used throughout the rest of this document, its usage is intended to  
612 cover the more specific notion of URL as well.

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

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

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

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

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

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

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

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

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

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

684 To summarize:

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

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

### 708 3. IPP Operations

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

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

721 The IPP/1.1 Printer operations are:

- 722 Print-Job (section 3.2.1)
- 723 Print-URI (section 3.2.2)
- 724 Validate-Job (section 3.2.3)
- 725 Create-Job (section 3.2.4)

726 Get-Printer-Attributes (section 3.2.5)  
727 Get-Jobs (section 3.2.6)  
728 Pause-Printer (section 3.3.5)  
729 Resume-Printer (section 3.3.6)  
730 Purge-Jobs (section 3.3.7)  
731

732 The Job operations are:

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

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

## 743 **3.1 Common Semantics**

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

### 746 **3.1.1 Required Parameters**

747 Every operation request contains the following REQUIRED parameters:

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

752  
753 Every operation response contains the following REQUIRED parameters:

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

758  
759 The "Encoding and Transport" document [IPP-PRO] defines special rules for the encoding of these  
760 parameters. All other operation elements are represented using the more generic encoding rules for  
761 attributes and groups of attributes.

### 762 3.1.2 Operation IDs and Request IDs

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

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

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

### 778 3.1.3 Attributes

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

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

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

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

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

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

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

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

### 838 **3.1.4 Character Set and Natural Language Operation Attributes**

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

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

### 851 3.1.4.1 Request Operation Attributes

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

854 "attributes-charset" (charset):

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

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

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

879  
880 See the 'charset' attribute syntax description in Section 4.1.7 for the syntax and semantic  
881 interpretation of the values of this attribute and for example values.

882

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

884 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that  
885 the client is supplying in this request. This attribute also identifies the natural language that the  
886 Printer object SHOULD use for all 'text' and 'name' attributes and status messages that the Printer  
887 object returns in the response to this request. See the 'naturalLanguage' attribute syntax description  
888 in section 4.1.8 for the syntax and semantic interpretation of the values of this attribute and for  
889 example values.

890

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

896

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

903

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

910

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

917

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

926



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

931

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

939

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

950

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

953

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

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

968

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

#### 974 **3.1.4.2 Response Operation Attributes**

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

977 "attributes-charset" (charset):

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

983

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

993

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

998

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

1000 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that  
1001 the IPP object is returning in this response. Unlike the "attributes-charset" operation attribute, the  
1002 IPP object **NEED NOT** return the same value as that supplied by the client in the request. The IPP  
1003 object **MAY** return the natural language of the Job object or the Printer's configured natural  
1004 language as identified by the Printer object's "natural-language-configured" attribute, rather than the  
1005 natural language supplied by the client. For any 'text' or 'name' attribute or status message in the  
1006 response that is in a different natural language than the value returned in the "attributes-natural-  
1007 language" operation attribute, the IPP object **MUST** use the Natural Language Override mechanism  
1008 (see sections 4.1.1.2 and 4.1.2.2) on each attribute value returned. The IPP object **MAY** use the  
1009 Natural Language Override mechanism redundantly, i.e., use it even when the value is in the same

1010 natural language as the value supplied in the "attributes-natural-language" operation attribute of the  
1011 response.

### 1012 3.1.5 Operation Targets

1013 All IPP operations are directed at IPP objects. For Printer operations, the operation is always directed at a  
1014 Printer object using one of its URIs (i.e., one of the values in the Printer object's "printer-uri-supported"  
1015 attribute). Even if the Printer object supports more than one URI, the client supplies only one URI as the  
1016 target of the operation. The client identifies the target object by supplying the correct URI in the "printer-  
1017 uri (uri)" operation attribute.

1018 For Job operations, the operation is directed at either:

- 1019 - The Job object itself using the Job object's URI. In this case, the client identifies the target object by  
1020 supplying the correct URI in the "job-uri (uri)" operation attribute.
- 1021 - The Printer object that created the Job object using both the Printer objects URI and the Job object's  
1022 Job ID. Since the Printer object that created the Job object generated the Job ID, it **MUST** be able to  
1023 correctly associate the client supplied Job ID with the correct Job object. The client supplies the  
1024 Printer object's URI in the "printer-uri (uri)" operation attribute and the Job object's Job ID in the  
1025 "job-id (integer(1:MAX))" operation attribute.

1026

1027 If the operation is directed at the Job object directly using the Job object's URI, the client **MUST NOT**  
1028 include the redundant "job-id" operation attribute.

1029 The operation target attributes are **REQUIRED** operation attributes that **MUST** be included in every  
1030 operation request. Like the charset and natural language attributes (see section 3.1.4), the operation target  
1031 attributes are specially ordered operation attributes. In all cases, the operation target attributes immediately  
1032 follow the "attributes-charset" and "attributes-natural-language" attributes within the operation attribute  
1033 group, however the specific ordering rules are:

- 1034 - In the case where there is only one operation target attribute (i.e., either only the "printer-uri" attribute  
1035 or only the "job-uri" attribute), that attribute **MUST** be the third attribute in the operation attributes  
1036 group.
- 1037 - In the case where Job operations use two operation target attributes (i.e., the "printer-uri" and "job-id"  
1038 attributes), the "printer-uri" attribute **MUST** be the third attribute and the "job-id" attribute **MUST**  
1039 be the fourth attribute.

1040

1041 In all cases, the target URIs contained within the body of IPP operation requests and responses must be in  
1042 absolute format rather than relative format (a relative URL identifies a resource with the scope of the HTTP  
1043 server, but does not include scheme, host or port).

1044 The following rules apply to the use of port numbers in URIs that identify IPP objects:

- 1045 1. If the URI scheme allows the port number to be explicitly included in the URI string, and a port  
1046 number is specified within the URI, then that port number **MUST** be used by the client to contact  
1047 the IPP object.

1048

1049

2. If the URI scheme allows the port number to be explicitly included in the URI string, and a port number is not specified within the URI, then default port number implied by that URI scheme MUST be used by the client to contact the IPP object.

1052

1053

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

1054

1055

1056

1057

Note: The IPP "Encoding and Transport document [IPP-PRO] shows a mapping of IPP onto HTTP/1.1 [RFC2616] and defines a new default port number for using IPP over HTTP/1.1.

1058

### 3.1.6 Operation Response Status Codes and Status Messages

1059

1060

1061

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

1062

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

1063

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

1064

1065

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

1066

1067

1068

1069

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

1070

1071

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

1072

1073

1074

If the client supplies unsupported values for the following parameters or Operation attributes, the Printer object MUST reject the operation, NEED NOT return the unsupported attribute value in the Unsupported 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

1075

1076 If the client supplies unsupported values for other attributes, or unsupported attributes, the Printer returns  
1077 the status code defined in section 3.1.7 on Unsupported Attributes.

### 1078 **3.1.6.2 "status-message" (text(255))**

1079 The OPTIONAL "status-message" operation attribute provides a short textual description of the status of  
1080 the operation. The "status-message" attribute's syntax is "text(255)", so the maximum length is 255 octets  
1081 (see section 4.1.1). The status message is intended for the human end user. If a response does include a  
1082 "status-message" attribute, an IPP client NEED NOT examine or display the messages, however it  
1083 SHOULD do so in some implementation specific manner. The "status-message" is especially useful for a  
1084 later version of a Printer object to return as supplemental information for the human user to accompany a  
1085 status code that an earlier version of a client might not understand.

1086 If the Printer object supports the "status-message" operation attribute, the Printer object MUST be able to  
1087 generate this message in any of the natural languages identified by the Printer object's "generated-natural-  
1088 language-supported" attribute (see the "attributes-natural-language" operation attribute specified in section  
1089 3.1.4.1. Section 13 suggests the text for the status message returned by the Printer for use with the English  
1090 natural language.

1091 As described in section 3.1.4.1 for any returned 'text' attribute, if there is a choice for generating this  
1092 message, the Printer object uses the natural language indicated by the value of the "attributes-natural-  
1093 language" in the client request if supported, otherwise the Printer object uses the value in the Printer  
1094 object's own "natural-language-configured" attribute.

1095 If the Printer object supports the "status-message" operation attribute, it SHOULD use the REQUIRED 'utf-  
1096 8' charset to return a status message for the following error status codes (see section 13): 'client-error-bad-  
1097 request', 'client-error-charset-not-supported', 'server-error-internal-error', 'server-error-operation-not-  
1098 supported', and 'server-error-version-not-supported'. In this case, it MUST set the value of the "attributes-  
1099 charset" operation attribute to 'utf-8' in the error response.

### 1100 **3.1.6.3 "detailed-status-message" (text(MAX))**

1101 The OPTIONAL "detailed-status-message" operation attribute provides additional more detailed technical  
1102 and implementation-specific information about the operation. The "detailed-status-message" attribute's  
1103 syntax is "text(MAX)", so the maximum length is 1023 octets (see section 4.1.1). If the Printer objects  
1104 supports the "detailed-status-message" operation attribute, the Printer NEED NOT localize the message,  
1105 since it is intended for use by the system administrator or other experienced technical persons. Localization  
1106 might obscure the technical meaning of such messages. Clients MUST NOT attempt to parse the value of  
1107 this attribute. See the "document-access-error" operation attribute (section 3.1.6.4) for additional errors that  
1108 a program can process.

### 1109 **3.1.6.4 "document-access-error" (text(MAX))**

1110 This OPTIONAL operation attribute provides additional information about any document access errors  
1111 encountered by the Printer before it returned a response to the Print-URI (section 3.2.2) or Send-URI

1112 (section 3.3.1) operation. For errors in the protocol identified by the URI scheme in the "document-uri"  
1113 operation attribute, such as 'http:' or 'ftp:', the error code is returned in parentheses, followed by the URI.  
1114 For example:

1115           (404) http://ftp.pwg.org/pub/pwg/ipp/new\_MOD/ipp-model-v11-990510.pdf  
1116

1117 Most Internet protocols use decimal error codes (unlike IPP), so the ASCII error code representation is in  
1118 decimal.

### 1119 **3.1.7 Unsupported Attributes**

1120 The Unsupported Attributes group contains attributes that are not supported by the operation. This group is  
1121 primarily for the job creation operations, but all operations can return this group.

1122 A Printer object **MUST** include an Unsupported Attributes group in a response if the status code is one of  
1123 the following: 'successful-ok-ignored-or-substituted-attributes', 'successful-ok-conflicting-attributes', 'client-  
1124 error-attributes-or-values-not-supported' or 'client-error-conflicting-attributes'.

1125 If the status code is one of the four specified in the preceding paragraph, the Unsupported Attributes group  
1126 **MUST** contain all of those attributes and only those attributes that are:

- 1127       a. an Operation or Job Template attribute supplied in the request, and
- 1128       b. unsupported by the printer. See below for details on the three categories "unsupported" attributes.

1129 If the status code is one of those in the table in section 3.1.6.1, the Unsupported Attributes group **NEED**  
1130 **NOT** contain the unsupported parameter or attribute indicated in that table.

1131 If the Printer object is not returning any Unsupported Attributes in the response, the Printer object  
1132 **SHOULD** omit Group 2 rather than sending an empty group. However, a client **MUST** be able to accept an  
1133 empty group.

1134 Unsupported attributes fall into three categories:

- 1135       1. The Printer object does not support the supplied attribute (no matter what the attribute syntax or  
1136       value).
- 1137       2. The Printer object does support the attribute, but does not support some or all of the particular  
1138       attribute syntaxes or values supplied by the client (i.e., the Printer object does not have those  
1139       attribute syntaxes or values in its corresponding "xxx-supported" attribute).
- 1140       3. The Printer object does support the attributes and values supplied, but the particular values are in  
1141       conflict with one another, because they violate a constraint, such as not being able to staple  
1142       transparencies.

1143 In the case of an unsupported attribute name, the Printer object returns the client-supplied attribute with a  
1144 substituted value of 'unsupported'. This value's syntax type is "out-of-band" and its encoding is defined by

1145 special rules for "out-of-band" values in the "Encoding and Transport" document [IPP-PRO]. Its value  
1146 indicates no support for the attribute itself (see the beginning of section 4.1).

1147 In the case of a supported attribute with one or more unsupported attribute syntaxes or values, the Printer  
1148 object simply returns the client-supplied attribute with the unsupported attribute syntaxes or values as  
1149 supplied by the client. This indicates support for the attribute, but no support for that particular attribute  
1150 syntax or value. If the client supplies a multi-valued attribute with more than one value and the Printer  
1151 object supports the attribute but only supports a subset of the client-supplied attribute syntaxes or values,  
1152 the Printer object **MUST** return only those attribute syntaxes or values that are unsupported.

1153 In the case of two (or more) supported attribute values that are in conflict with one another (although each  
1154 is supported independently, the values conflict when requested together within the same job), the Printer  
1155 object **MUST** return all the values that it ignores or substitutes to resolve the conflict, but not any of the  
1156 values that it is still using. The choice for exactly how to resolve the conflict is implementation dependent.  
1157 See sections 3.2.1.2 and 15. See The Implementer's Guide [IPP-IIG] for an example.

### 1158 **3.1.8 Versions**

1159 Each operation request and response carries with it a "version-number" parameter. Each value of the  
1160 "version-number" is in the form "X.Y" where X is the major version number and Y is the minor version  
1161 number. By including a version number in the client request, it allows the client to identify which version  
1162 of IPP it is interested in using, i.e., the version whose conformance requirements the client may be  
1163 depending upon the Printer to meet.

1164 If the IPP object does not support that major version number supplied by the client, i.e., the major version  
1165 field of the "version-number" parameter does not match any of the values of the Printer's "ipp-versions-  
1166 supported" (see section 4.4.14), the object **MUST** respond with a status code of 'server-error-version-not-  
1167 supported' along with the closest version number that is supported (see section 13.1.5.4). If the major  
1168 version number is supported, but the minor version number is not, the IPP object **SHOULD** accept and  
1169 attempt to perform the request (or reject the request if the operation is not supported), else it rejects the  
1170 request and returns the 'server-error-version-not-supported' status code. In all cases, the IPP object **MUST**  
1171 return the "version-number" that it supports that is closest to the version number supplied by the client in  
1172 the request.

1173 There is no version negotiation per se. However, if after receiving a 'server-error-version-not-supported'  
1174 status code from an IPP object, a client **SHOULD** try again with a different version number. A client **MAY**  
1175 also determine the versions supported either from a directory that conforms to Appendix E (see section 16)  
1176 or by querying the Printer object's "ipp-versions-supported" attribute (see section 4.4.14) to determine  
1177 which versions are supported.

1178 An IPP object implementation **MUST** support version '1.1', i.e., meet the conformance requirements for  
1179 IPP/1.1 as specified in this document and [IPP-PRO]. It is recommended that IPP object implementations  
1180 accept any request with the major version '1' (or reject the request if the operation is not supported).

1181 There is only one notion of "version number" that covers both IPP Model and IPP Protocol changes. Thus  
1182 the version number **MUST** change when introducing a new version of the Model and Semantics document  
1183 (this document) or a new version of the "Encoding and Transport" document [IPP-PRO].

1184 Changes to the major version number of the Model and Semantics document indicate structural or syntactic  
1185 changes that make it impossible for older version of IPP clients and Printer objects to correctly parse and  
1186 correctly process the new or changed attributes, operations and responses. If the major version number  
1187 changes, the minor version numbers is set to zero. As an example, adding the **REQUIRED** "ipp-attribute-  
1188 fidelity" attribute to version '1.1' (if it had not been part of version '1.0'), would have required a change to  
1189 the major version number, since an IPP/1.0 Printer would not have processed a request with the correct  
1190 semantics that contained the "ipp-attribute-fidelity" attribute that it did not know about. Items that might  
1191 affect the changing of the major version number include any changes to the Model and Semantics document  
1192 (this document) or the "Encoding and Transport" document [IPP-PRO] itself, such as:

- 1193 - reordering of ordered attributes or attribute sets
- 1194 - changes to the syntax of existing attributes
- 1195 - adding **REQUIRED** (for an IPP object to support) operation attribute groups
- 1196 - adding values to existing **REQUIRED** operation attributes
- 1197 - adding **REQUIRED** operations

1198

1199 Changes to the minor version number indicate the addition of new features, attributes and attribute values  
1200 that may not be understood by all IPP objects, but which can be ignored if not understood. Items that might  
1201 affect the changing of the minor version number include any changes to the model objects and attributes but  
1202 not the encoding and transport rules [IPP-PRO] (except adding attribute syntaxes). Examples of such  
1203 changes are:

- 1204 - grouping all extensions not included in a previous version into a new version
- 1205 - adding new attribute values
- 1206 - adding new object attributes
- 1207 - adding **OPTIONAL** (for an IPP object to support) operation attributes (i.e., those attributes that an IPP  
1208 object can ignore without confusing clients)
- 1209 - adding **OPTIONAL** (for an IPP object to support) operation attribute groups (i.e., those attributes that  
1210 an IPP object can ignore without confusing clients)
- 1211 - adding new attribute syntaxes
- 1212 - adding **OPTIONAL** operations
- 1213 - changing Job Description attributes or Printer Description attributes from **OPTIONAL** to **REQUIRED**  
1214 or vice versa.
- 1215 - adding **OPTIONAL** attribute syntaxes to an existing attribute.

1216 The encoding of the "version-number" **MUST NOT** change over any version number (either major or  
1217 minor). This rule guarantees that all future versions will be backwards compatible with all previous  
1218 versions (at least for checking the "version-number"). In addition, any protocol elements (attributes, error  
1219 codes, tags, etc.) that are not carried forward from one version to the next are deprecated so that they can  
1220 never be reused with new semantics.



1221 Implementations that support a certain version NEED NOT support ALL previous versions. As each new  
1222 version is defined (through the release of a new IPP specification document), that version will specify  
1223 which previous versions MUST and which versions SHOULD be supported in compliant implementations.

### 1224 3.1.9 Job Creation Operations

1225 In order to "submit a print job" and create a new Job object, a client issues a create request. A create  
1226 request is any one of following three operation requests:

- 1227 - The Print-Job Request: A client that wants to submit a print job with only a single document uses the  
1228 Print-Job operation. The operation allows for the client to "push" the document data to the Printer  
1229 object by including the document data in the request itself.  
1230
- 1231 - The Print-URI Request: A client that wants to submit a print job with only a single document (where  
1232 the Printer object "pulls" the document data instead of the client "pushing" the data to the Printer  
1233 object) uses the Print-URI operation. In this case, the client includes in the request only a URI  
1234 reference to the document data (not the document data itself).  
1235
- 1236 - The Create-Job Request: A client that wants to submit a print job with multiple documents uses the  
1237 Create-Job operation. This operation is followed by an arbitrary number (one or more) of Send-  
1238 Document and/or Send-URI operations (each creating another document for the newly create Job  
1239 object). The Send-Document operation includes the document data in the request (the client  
1240 "pushes" the document data to the printer), and the Send-URI operation includes only a URI  
1241 reference to the document data in the request (the Printer "pulls" the document data from the  
1242 referenced location). The last Send-Document or Send-URI request for a given Job object includes  
1243 a "last-document" operation attribute set to 'true' indicating that this is the last request.  
1244

1245 Throughout this model document, the term "create request" is used to refer to any of these three operation  
1246 requests.

1247 A Create-Job operation followed by only one Send-Document operation is semantically equivalent to a  
1248 Print-Job operation, however, for performance reasons, the client SHOULD use the Print-Job operation for  
1249 all single document jobs. Also, Print-Job is a REQUIRED operation (all implementations MUST support  
1250 it) whereas Create-Job is an OPTIONAL operation, hence some implementations might not support it.

1251 Job submission time is the point in time when a client issues a create request. The initial state of every Job  
1252 object is the 'pending', 'pending-held', or 'processing' state (see section 4.3.7). When the Printer object  
1253 begins processing the print job, the Job object's state moves to 'processing'. This is known as job  
1254 processing time. There are validation checks that must be done at job submission time and others that must  
1255 be performed at job processing time.

1256 At job submission time and at the time a Validate-Job operation is received, the Printer MUST do the  
1257 following:

- 1258 1. Process the client supplied attributes and either accept or reject the request
- 1259 2. Validate the syntax of and support for the scheme of any client supplied URI

1260

1261 At job submission time the Printer object **MUST** validate whether or not the supplied attributes, attribute  
1262 syntaxes, and values are supported by matching them with the Printer object's corresponding "xxx-  
1263 supported" attributes. See section 3.1.7 for details. [IPP-IIG] presents suggested steps for an IPP object to  
1264 either accept or reject any request and additional steps for processing create requests.

1265 At job submission time the Printer object **NEED NOT** perform the validation checks reserved for job  
1266 processing time such as:

- 1267 1. Validating the document data
- 1268 2. Validating the actual contents of any client supplied URI (resolve the reference and follow the link to  
1269 the document data)

1270

1271 At job submission time, these additional job processing time validation checks are essentially useless, since  
1272 they require actually parsing and interpreting the document data, are not guaranteed to be 100% accurate,  
1273 and **MUST** be done, yet again, at job processing time. Also, in the case of a URI, checking for availability  
1274 at job submission time does not guarantee availability at job processing time. In addition, at job processing  
1275 time, the Printer object might discover any of the following conditions that were not detectable at job  
1276 submission time:

- 1277 - runtime errors in the document data,
- 1278 - nested document data that is in an unsupported format,
- 1279 - the URI reference is no longer valid (i.e., the server hosting the document might be down), or
- 1280 - any other job processing error

1281

1282 At job submission time, a Printer object, especially a non-spooling Printer, **MAY** accept jobs that it does  
1283 not have enough space for. In such a situation, a Printer object **MAY** stop reading data from a client for an  
1284 indefinite period of time. A client **MUST** be prepared for a write operation to block for an indefinite period  
1285 of time (see section 5.1 on client conformance).

1286 When a Printer object has too little space for starting a new job, it **MAY** reject a new create request. In this  
1287 case, a Printer object **MUST** return a response (in reply to the rejected request) with a status-code of 'server-  
1288 error-busy' (see section 14.1.5.8) and it **MAY** close the connection before receiving all bytes of the  
1289 operation. A Printer **SHOULD** indicate that it is temporarily unable to accept jobs by setting the 'spool-  
1290 space-full' value in its "printer-state-reasons" attribute and removing the value when it can accept another  
1291 job (see section 4.4.12).

1292 When receiving a 'server-error-busy' status-code in an operation response, a client **MUST** be prepared for  
1293 the Printer object to close the connection before the client has sent all of the data (especially for the Print-  
1294 Job operation). A client **MUST** be prepared to keep submitting a create request until the IPP Printer object  
1295 accepts the create request.

1296 At job processing time, since the Printer object has already responded with a successful status code in the  
1297 response to the create request, if the Printer object detects an error, the Printer object is unable to inform the  
1298 end user of the error with an operation status code. In this case, the Printer, depending on the error, can set

1299 the job object's "job-state", "job-state-reasons", or "job-state-message" attributes to the appropriate value(s)  
1300 so that later queries can report the correct job status.

1301 Note: Asynchronous notification of events is outside the scope of this IPP/1.1 document.

1302

## 1303 **3.2 Printer Operations**

1304 All Printer operations are directed at Printer objects. A client **MUST** always supply the "printer-uri"  
1305 operation attribute in order to identify the correct target of the operation.

### 1306 **3.2.1 Print-Job Operation**

1307 This **REQUIRED** operation allows a client to submit a print job with only one document and supply the  
1308 document data (rather than just a reference to the data). See Section 15 for the suggested steps for  
1309 processing create operations and their Operation and Job Template attributes.

#### 1310 **3.2.1.1 Print-Job Request**

1311 The following groups of attributes are supplied as part of the Print-Job Request:

1312 Group 1: Operation Attributes

1313 Natural Language and Character Set:

1314 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.

1315 The Printer object **MUST** copy these values to the corresponding Job Description attributes  
1316 described in sections 4.3.19 and 4.3.20.

1317

1318 Target:

1319 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1320 section 3.1.5.

1321

1322 Requesting User Name:

1323 The "requesting-user-name" (name(MAX)) attribute **SHOULD** be supplied by the client as  
1324 described in section 8.3.

1325

1326 "job-name" (name(MAX)):

1327 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this attribute. It  
1328 contains the client supplied Job name. If this attribute is supplied by the client, its value is used for  
1329 the "job-name" attribute of the newly created Job object. The client **MAY** automatically include any  
1330 information that will help the end-user distinguish amongst his/her jobs, such as the name of the  
1331 application program along with information from the document, such as the document name,  
1332 document subject, or source file name. If this attribute is not supplied by the client, the Printer

1333 generates a name to use in the "job-name" attribute of the newly created Job object (see Section  
1334 4.3.5).

1335

1336 "ipp-attribute-fidelity" (boolean):

1337 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this attribute.  
1338 The value 'true' indicates that total fidelity to client supplied Job Template attributes and values is  
1339 required, else the Printer object **MUST** reject the Print-Job request. The value 'false' indicates that a  
1340 reasonable attempt to print the Job object is acceptable and the Printer object **MUST** accept the  
1341 Print-Job request. If not supplied, the Printer object assumes the value is 'false'. All Printer objects  
1342 **MUST** support both types of job processing. See section 15 for a full description of "ipp-attribute-  
1343 fidelity" and its relationship to other attributes, especially the Printer object's "pdl-override-  
1344 supported" attribute.

1345

1346 "document-name" (name(MAX)):

1347 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this attribute.  
1348 It contains the client supplied document name. The document name **MAY** be different than the Job  
1349 name. Typically, the client software automatically supplies the document name on behalf of the end  
1350 user by using a file name or an application generated name. If this attribute is supplied, its value can  
1351 be used in a manner defined by each implementation. Examples include: printed along with the Job  
1352 (job start sheet, page adornments, etc.), used by accounting or resource tracking management tools,  
1353 or even stored along with the document as a document level attribute. IPP/1.1 does not support the  
1354 concept of document level attributes.

1355

1356 "compression" (type3 keyword)

1357 The client **OPTIONALLY** supplies this attribute. The Printer object **MUST** support this attribute  
1358 and the "compression-supported" attribute (see section 4.4.32). The client supplied "compression"  
1359 operation attribute identifies the compression algorithm used on the document data. The following  
1360 cases exist:

1361

- 1362 a) If the client omits this attribute, the Printer object **MUST** assume that the data is not  
1363 compressed (i.e. the Printer follows the rules below as if the client supplied the  
1364 "compression" attribute with a value of 'none').
- 1365 b) If the client supplies this attribute, but the value is not supported by the Printer object,  
1366 i.e., the value is not one of the values of the Printer object's "compression-supported"  
1367 attribute, the Printer object **MUST** reject the request, and return the 'client-error-  
1368 compression-not-supported' status code. See section 3.1.7 for returning unsupported  
1369 attributes and values.
- 1370 c) If the client supplies the attribute and the Printer object supports the attribute value, the  
1371 Printer object uses the corresponding decompression algorithm on the document data.
- 1372 d) If the decompression algorithm fails before the Printer returns an operation response, the  
1373 Printer object **MUST** reject the request and return the 'client-error-compression-error'  
1374 status code.
- 1375 e) If the decompression algorithm fails after the Printer returns an operation response, the  
1376 Printer object **MUST** abort the job and add the 'compression-error' value to the job's  
"job-state-reasons" attribute.

1377 f) If the decompression algorithm succeeds, the document data MUST then have the format  
1378 specified by the job's "document-format" attribute, if supplied (see "document-format"  
1379 operation attribute definition below).  
1380

1381 "document-format" (mimeType) :

1382 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute.  
1383 The value of this attribute identifies the format of the supplied document data. The following cases  
1384 exist:

- 1385 a) If the client does not supply this attribute, the Printer object assumes that the document  
1386 data is in the format defined by the Printer object's "document-format-default" attribute.  
1387 (i.e. the Printer follows the rules below as if the client supplied the "document-format"  
1388 attribute with a value equal to the printer's default value).
- 1389 b) If the client supplies this attribute, but the value is not supported by the Printer object,  
1390 i.e., the value is not one of the values of the Printer object's "document-format-  
1391 supported" attribute, the Printer object MUST reject the request and return the 'client-  
1392 error-document-format-not-supported' status code.
- 1393 c) If the client supplies this attribute and its value is 'application/octet-stream' (i.e. to be  
1394 auto-sensed, see Section 4.1.9.1), and the format is not one of the document-formats that  
1395 the Printer can auto-sense, and this check occurs before the Printer returns an operation  
1396 response, then the Printer MUST reject the request and return the 'client-error-  
1397 document-format-not-supported' status code.
- 1398 d) If the client supplies this attribute, and the value is supported by the Printer object, the  
1399 Printer is capable of interpreting the document data.
- 1400 e) If interpreting of the document data fails before the Printer returns an operation response,  
1401 the Printer object MUST reject the request and return the 'client-error-document-format-  
1402 error' status code.
- 1403 f) If interpreting of the document data fails after the Printer returns an operation response,  
1404 the Printer object MUST abort the job and add the 'document-format-error' value to the  
1405 job's "job-state-reasons" attribute.  
1406

1407 "document-natural-language" (naturalLanguage):

1408 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1409 attribute. This attribute specifies the natural language of the document for those document-formats  
1410 that require a specification of the natural language in order to image the document unambiguously.  
1411 There are no particular values required for the Printer object to support.  
1412

1413 "job-k-octets" (integer(0:MAX))

1414 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1415 attribute and the "job-k-octets-supported" attribute (see section 4.4.33). The client supplied "job-k-  
1416 octets" operation attribute identifies the total size of the document(s) in K octets being submitted  
1417 (see section 4.3.17.1 for the complete semantics). If the client supplies the attribute and the Printer  
1418 object supports the attribute, the value of the attribute is used to populate the Job object's "job-k-  
1419 octets" Job Description attribute.  
1420

1421 For this attribute and the following two attributes ("job-impressions", and "job-media-sheets"), if the  
1422 client supplies the attribute, but the Printer object does not support the attribute, the Printer object  
1423 ignores the client-supplied value. If the client supplies the attribute and the Printer supports the  
1424 attribute, and the value is within the range of the corresponding Printer object's "xxx-supported"  
1425 attribute, the Printer object MUST use the value to populate the Job object's "xxx" attribute. If the  
1426 client supplies the attribute and the Printer supports the attribute, but the value is outside the range  
1427 of the corresponding Printer object's "xxx-supported" attribute, the Printer object MUST copy the  
1428 attribute and its value to the Unsupported Attributes response group, reject the request, and return  
1429 the 'client-error-attributes-or-values-not-supported' status code. If the client does not supply the  
1430 attribute, the Printer object MAY choose to populate the corresponding Job object attribute  
1431 depending on whether the Printer object supports the attribute and is able to calculate or discern the  
1432 correct value.

1433

1434 "job-impressions" (integer(0:MAX))

1435 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1436 attribute and the "job-impressions-supported" attribute (see section 4.4.34). The client supplied  
1437 "job-impressions" operation attribute identifies the total size in number of impressions of the  
1438 document(s) being submitted (see section 4.3.17.2 for the complete semantics).

1439

1440 See last paragraph under "job-k-octets".

1441

1442 "job-media-sheets" (integer(0:MAX))

1443 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
1444 attribute and the "job-media-sheets-supported" attribute (see section 4.4.35). The client supplied  
1445 "job-media-sheets" operation attribute identifies the total number of media sheets to be produced for  
1446 this job (see section 4.3.17.3 for the complete semantics).

1447

1448 See last paragraph under "job-k-octets".

1449

## 1450 Group 2: Job Template Attributes

1451 The client OPTIONALLY supplies a set of Job Template attributes as defined in section 4.2. If the  
1452 client is not supplying any Job Template attributes in the request, the client SHOULD omit Group 2  
1453 rather than sending an empty group. However, a Printer object MUST be able to accept an empty  
1454 group.

1455

## 1456 Group 3: Document Content

1457 The client MUST supply the document data to be processed.

1458

1459 In addition to the MANDATORY parameters required for every operation request, the simplest Print-Job  
1460 Request consists of just the "attributes-charset" and "attributes-natural-language" operation attributes; the  
1461 "printer-uri" target operation attribute; the Document Content and nothing else. In this simple case, the  
1462 Printer object:

- 1463 - creates a new Job object (the Job object contains a single document),  
1464 - stores a generated Job name in the "job-name" attribute in the natural language and charset requested  
1465 (see Section 3.1.4.1) (if those are supported, otherwise using the Printer object's default natural  
1466 language and charset), and  
1467 - at job processing time, uses its corresponding default value attributes for the supported Job Template  
1468 attributes that were not supplied by the client as IPP attribute or embedded instructions in the  
1469 document data.  
1470

### 1471 3.2.1.2 Print-Job Response

1472 The Printer object MUST return to the client the following sets of attributes as part of the Print-Job  
1473 Response:

#### 1474 Group 1: Operation Attributes

##### 1475 Status Message:

1476 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
1477 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
1478 attribute as described in sections 13 and 3.1.6. If the client supplies unsupported or conflicting Job  
1479 Template attributes or values, the Printer object MUST reject or accept the Print-Job request  
1480 depending on the whether the client supplied a 'true' or 'false' value for the "ipp-attribute-fidelity"  
1481 operation attribute. See the Implementer's Guide [IPP-IIG] for a complete description of the  
1482 suggested steps for processing a create request.  
1483

##### 1484 Natural Language and Character Set:

1485 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.  
1486

#### 1487 Group 2: Unsupported Attributes

1488 See section 3.1.7 for details on returning Unsupported Attributes.  
1489

1490 The value of the "ipp-attribute-fidelity" supplied by the client does not affect what attributes the  
1491 Printer object returns in this group. The value of "ipp-attribute-fidelity" only affects whether the  
1492 Print-Job operation is accepted or rejected. If the job is accepted, the client may query the job using  
1493 the Get-Job-Attributes operation requesting the unsupported attributes that were returned in the  
1494 create response to see which attributes were ignored (not stored on the Job object) and which  
1495 attributes were stored with other (substituted) values.  
1496

#### 1497 Group 3: Job Object Attributes

##### 1498 "job-uri" (uri):

1499 The Printer object MUST return the Job object's URI by returning the contents of the REQUIRED  
1500 "job-uri" Job object attribute. The client uses the Job object's URI when directing operations at the  
1501 Job object. The Printer object always uses its configured security policy when creating the new  
1502 URI. However, if the Printer object supports more than one URI, the Printer object also uses

1503 information about which URI was used in the Print-Job Request to generated the new URI so that  
1504 the new URI references the correct access channel. In other words, if the Print-Job Request comes  
1505 in over a secure channel, the Printer object MUST generate a Job URI that uses the secure channel  
1506 as well.

1507

1508 "job-id" (integer(1:MAX)):

1509 The Printer object MUST return the Job object's Job ID by returning the REQUIRED "job-id" Job  
1510 object attribute. The client uses this "job-id" attribute in conjunction with the "printer-uri" attribute  
1511 used in the Print-Job Request when directing Job operations at the Printer object.

1512

1513 "job-state":

1514 The Printer object MUST return the Job object's REQUIRED "job-state" attribute. The value of this  
1515 attribute (along with the value of the next attribute: "job-state-reasons") is taken from a "snapshot"  
1516 of the new Job object at some meaningful point in time (implementation defined) between when the  
1517 Printer object receives the Print-Job Request and when the Printer object returns the response.

1518

1519 "job-state-reasons":

1520 The Printer object MUST return the Job object's REQUIRED "job-state-reasons" attribute.

1521

1522 "job-state-message":

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

1527

1528 "number-of-intervening-jobs":

1529 The Printer object OPTIONALLY returns the Job object's OPTIONAL "number-of-intervening-  
1530 jobs" attribute. If the Printer object supports this attribute then it MUST be returned in the response.  
1531 If this attribute is not returned in the response, the client can assume that the "number-of-  
1532 intervening-jobs" attribute is not supported and will not be returned in a subsequent Job object  
1533 query.

1534

1535 Note: Since any printer state information which affects a job's state is reflected in the "job-state" and  
1536 "job-state-reasons" attributes, it is sufficient to return only these attributes and no specific printer  
1537 status attributes.

1538

1539 Note: In addition to the MANDATORY parameters required for every operation response, the simplest  
1540 response consists of the just the "attributes-charset" and "attributes-natural-language" operation attributes  
1541 and the "job-uri", "job-id", and "job-state" Job Object Attributes. In this simplest case, the status code is  
1542 'successful-ok' and there is no "status-message" or "detailed-status-message" operation attribute.



### 1543 3.2.2 Print-URI Operation

1544 This OPTIONAL operation is identical to the Print-Job operation (section 3.2.1) except that a client  
1545 supplies a URI reference to the document data using the "document-uri" (uri) operation attribute (in Group  
1546 1) rather than including the document data itself. Before returning the response, the Printer MUST validate  
1547 that the Printer supports the retrieval method (e.g., http, ftp, etc.) implied by the URI, and MUST check for  
1548 valid URI syntax. If the client-supplied URI scheme is not supported, i.e. the value is not in the Printer  
1549 object's "referenced-uri-scheme-supported" attribute, the Printer object MUST reject the request and return  
1550 the 'client-error-uri-scheme-not-supported' status code.

1551 The IPP Printer MAY validate the accessibility of the document as part of the operation or subsequently. If  
1552 the Printer determines an accessibility problem before returning an operation response, it rejects the request  
1553 and returns the 'client-error-document-access-error' status code. The Printer MAY also return a specific  
1554 document access error code using the "document-access-error" operation attribute (see section 3.1.6.4).

1555 If the Printer determines this document accessibility problem after accepting the request and returning an  
1556 operation response with one of the successful status codes, the Printer adds the 'document-access-error'  
1557 value to the job's "job-state-reasons" attribute and MAY populate the job's "job-document-access-errors"  
1558 Job Description attribute (see section 4.3.11). See The Implementer's Guide [IPP-IIG] for suggested  
1559 additional checks.

1560 If the Printer object supports this operation, it MUST support the "reference-uri-schemes-supported" Printer  
1561 attribute (see section 4.4.27).

1562 It is up to the IPP object to interpret the URI and subsequently "pull" the document from the source  
1563 referenced by the URI string.

### 1564 3.2.3 Validate-Job Operation

1565 This REQUIRED operation is similar to the Print-Job operation (section 3.2.1) except that a client supplies  
1566 no document data and the Printer allocates no resources (i.e., it does not create a new Job object). This  
1567 operation is used only to verify capabilities of a printer object against whatever attributes are supplied by  
1568 the client in the Validate-Job request. By using the Validate-Job operation a client can validate that an  
1569 identical Print-Job operation (with the document data) would be accepted. The Validate-Job operation also  
1570 performs the same security negotiation as the Print-Job operation (see section 8), so that a client can check  
1571 that the client and Printer object security requirements can be met before performing a Print-Job operation.

1572 The Validate-Job operation does not accept a "document-uri" attribute in order to allow a client to check  
1573 that the same Print-URI operation will be accepted, since the client doesn't send the data with the Print-URI  
1574 operation. The client SHOULD just issue the Print-URI request.

1575 The Printer object returns the same status codes, Operation Attributes (Group 1) and Unsupported  
1576 Attributes (Group 2) as the Print-Job operation. However, no Job Object Attributes (Group 3) are returned,  
1577 since no Job object is created.

### 1578 3.2.4 Create-Job Operation

1579 This OPTIONAL operation is similar to the Print-Job operation (section 3.2.1) except that in the Create-Job  
1580 request, a client does not supply document data or any reference to document data. Also, the client does not  
1581 supply any of the "document-name", "document-format", "compression", or "document-natural-language"  
1582 operation attributes. This operation is followed by one or more Send-Document or Send-URI operations.  
1583 In each of those operation requests, the client OPTIONALLY supplies the "document-name", "document-  
1584 format", and "document-natural-language" attributes for each document in the multi-document Job object.

1585 If a Printer object supports the Create-Job operation, it MUST also support the Send-Document operation  
1586 and also MAY support the Send-URI operation.

1587 If the Printer object supports this operation, it MUST support the "multiple-operation-time-out" Printer  
1588 attribute (see section 4.4.31).

1589 If the Printer object supports this operation, then it MUST support the "multiple-document-jobs-supported"  
1590 Printer Description attribute (see section 4.4.16) and indicate whether or not it supports multiple-document  
1591 jobs.

1592 If the Printer object supports this operation and supports multiple documents in a job, then it MUST support  
1593 the "multiple-document-handling" Job Template job attribute with at least one value (see section 4.2.4) and  
1594 the associated "multiple-document-handling-default" and "multiple-document-handling-supported" Job  
1595 Template Printer attributes (see section 4.2).

1596 After the Create-Job operation has completed, the value of the "job-state" attribute is similar to the "job-  
1597 state" after a Print-Job, even though no document-data has arrived. A Printer MAY set the 'job-data-  
1598 insufficient' value of the job's "job-state-reason" attribute to indicate that processing cannot begin until  
1599 sufficient data has arrived and set the "job-state" to either 'pending' or 'pending-held'. A non-spooling  
1600 printer that doesn't implement the 'pending' job state may even set the "job-state" to 'processing', even  
1601 though there is not yet any data to process. See sections 4.3.7 and 4.3.8.

### 1602 3.2.5 Get-Printer-Attributes Operation

1603 This REQUIRED operation allows a client to request the values of the attributes of a Printer object. In the  
1604 request, the client supplies the set of Printer attribute names and/or attribute group names in which the  
1605 requester is interested. In the response, the Printer object returns a corresponding attribute set with the  
1606 appropriate attribute values filled in.

1607 For Printer objects, the possible names of attribute groups are:

- 1608 - 'job-template': the subset of the Job Template attributes that apply to a Printer object (the last two  
1609 columns of the table in Section 4.2) that the implementation supports for Printer objects.
- 1610 - 'printer-description': the subset of the attributes specified in Section 4.4 that the implementation  
1611 supports for Printer objects.
- 1612 - 'all': the special group 'all' that includes all attributes that the implementation supports for Printer  
1613 objects.

1614

1615 Since a client MAY request specific attributes or named groups, there is a potential that there is some  
1616 overlap. For example, if a client requests, 'printer-name' and 'all', the client is actually requesting the  
1617 "printer-name" attribute twice: once by naming it explicitly, and once by inclusion in the 'all' group. In such  
1618 cases, the Printer object NEED NOT return each attribute only once in the response even if it is requested  
1619 multiple times. The client SHOULD NOT request the same attribute in multiple ways.

1620 It is NOT REQUIRED that a Printer object support all attributes belonging to a group (since some attributes  
1621 are OPTIONAL). However, it is REQUIRED that each Printer object support all group names.

### 1622 3.2.5.1 Get-Printer-Attributes Request

1623 The following sets of attributes are part of the Get-Printer-Attributes Request:

#### 1624 Group 1: Operation Attributes

1625 Natural Language and Character Set:

1626 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.

1627

1628 Target:

1629 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1630 section 3.1.5.

1631

1632 Requesting User Name:

1633 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1634 described in section 8.3.

1635

1636 "requested-attributes" (1setOf keyword) :

1637 The client OPTIONALLY supplies a set of attribute names and/or attribute group names in whose  
1638 values the requester is interested. The Printer object MUST support this attribute. If the client  
1639 omits this attribute, the Printer MUST respond as if this attribute had been supplied with a value of  
1640 'all'.

1641

1642 "document-format" (mimeMediaType) :

1643 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute.  
1644 This attribute is useful for a Printer object to determine the set of supported attribute values that  
1645 relate to the requested document format. The Printer object MUST return the attributes and values  
1646 that it uses to validate a job on a create or Validate-Job operation in which this document format is  
1647 supplied. The Printer object SHOULD return only (1) those attributes that are supported for the  
1648 specified format and (2) the attribute values that are supported for the specified document format.  
1649 By specifying the document format, the client can get the Printer object to eliminate the attributes  
1650 and values that are not supported for a specific document format. For example, a Printer object  
1651 might have multiple interpreters to support both 'application/postscript' (for PostScript) and  
1652 'text/plain' (for text) documents. However, for only one of those interpreters might the Printer  
1653 object be able to support "number-up" with values of '1', '2', and '4'. For the other interpreter it

1654 might be able to only support "number-up" with a value of '1'. Thus a client can use the Get-Printer-  
1655 Attributes operation to obtain the attributes and values that will be used to accept/reject a create job  
1656 operation.  
1657

1658 If the Printer object does not distinguish between different sets of supported values for each  
1659 different document format when validating jobs in the create and Validate-Job operations, it MUST  
1660 NOT distinguish between different document formats in the Get-Printer-Attributes operation. If the  
1661 Printer object does distinguish between different sets of supported values for each different  
1662 document format specified by the client, this specialization applies only to the following Printer  
1663 object attributes:

- 1664 - Printer attributes that are Job Template attributes ("xxx-default" "xxx-supported", and "xxx-  
1665 ready" in the Table in Section 4.2),
- 1666 - "pdl-override-supported",
- 1667 - "compression-supported",
- 1668 - "job-k-octets-supported",
- 1669 - "job-impressions-supported",
- 1670 - "job-media-sheets-supported"
- 1671 - "printer-driver-installer",
- 1672 - "color-supported", and
- 1673 - "reference-uri-schemes-supported"
- 1674

1675  
1676 The values of all other Printer object attributes (including "document-format-supported") remain  
1677 invariant with respect to the client supplied document format (except for new Printer description  
1678 attribute as registered according to section 6.2).  
1679

1680 If the client omits this "document-format" operation attribute, the Printer object MUST respond as if  
1681 the attribute had been supplied with the value of the Printer object's "document-format-default"  
1682 attribute. It is RECOMMENDED that the client always supply a value for "document-format",  
1683 since the Printer object's "document-format-default" may be 'application/octet-stream', in which case  
1684 the returned attributes and values are for the union of the document formats that the Printer can  
1685 automatically sense. For more details, see the description of the 'mimeType' attribute syntax  
1686 in section 4.1.9.  
1687

1688 If the client supplies a value for the "document-format" Operation attribute that is not supported by  
1689 the Printer, i.e., is not among the values of the Printer object's "document-format-supported"  
1690 attribute, the Printer object MUST reject the operation and return the 'client-error-document-format-  
1691 not-supported' status code.  
1692

### 1693 3.2.5.2 Get-Printer-Attributes Response

1694 The Printer object returns the following sets of attributes as part of the Get-Printer-Attributes Response:

1695 Group 1: Operation Attributes

1696 Status Message:

1697 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
1698 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
1699 attribute as described in sections 13 and 3.1.6.

1700

1701 Natural Language and Character Set:

1702 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.

1703

1704 Group 2: Unsupported Attributes

1705 See section 3.1.7 for details on returning Unsupported Attributes.

1706

1707 The response NEED NOT contain the "requested-attributes" operation attribute with any supplied  
1708 values (attribute keywords) that were requested by the client but are not supported by the IPP object.  
1709 If the Printer object does include unsupported attributes referenced in "requested-attributes" and  
1710 such attributes include group names, such as 'all', the unsupported attributes MUST NOT include  
1711 attributes described in the standard but not supported by the implementation.

1712

1713 Group 3: Printer Object Attributes

1714 This is the set of requested attributes and their current values. The Printer object ignores (does not  
1715 respond with) any requested attribute which is not supported. The Printer object MAY respond with  
1716 a subset of the supported attributes and values, depending on the security policy in force. However,  
1717 the Printer object MUST respond with the 'unknown' value for any supported attribute (including all  
1718 REQUIRED attributes) for which the Printer object does not know the value. Also the Printer  
1719 object MUST respond with the 'no-value' for any supported attribute (including all REQUIRED  
1720 attributes) for which the system administrator has not configured a value. See the description of the  
1721 "out-of-band" values in the beginning of Section 4.1.

1722

### 1723 3.2.6 Get-Jobs Operation

1724 This REQUIRED operation allows a client to retrieve the list of Job objects belonging to the target Printer  
1725 object. The client may also supply a list of Job attribute names and/or attribute group names. A group of  
1726 Job object attributes will be returned for each Job object that is returned.

1727 This operation is similar to the Get-Job-Attributes operation, except that this Get-Jobs operation returns  
1728 attributes from possibly more than one object (see the description of Job attribute group names in section  
1729 3.3.4).

#### 1730 3.2.6.1 Get-Jobs Request

1731 The client submits the Get-Jobs request to a Printer object.

1732 The following groups of attributes are part of the Get-Jobs Request:

## 1733 Group 1: Operation Attributes

## 1734 Natural Language and Character Set:

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

1736

## 1737 Target:

1738 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1739 section 3.1.5.

1740

## 1741 Requesting User Name:

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

1744

## 1745 "limit" (integer(1:MAX)):

1746 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1747 is an integer value that determines the maximum number of jobs that a client will receive from the  
1748 Printer even if "which-jobs" or "my-jobs" constrain which jobs are returned. The limit is a "stateless  
1749 limit" in that if the value supplied by the client is 'N', then only the first 'N' jobs are returned in the  
1750 Get-Jobs Response. There is no mechanism to allow for the next 'M' jobs after the first 'N' jobs. If  
1751 the client does not supply this attribute, the Printer object responds with all applicable jobs.

1752

## 1753 "requested-attributes" (1setOf keyword):

1754 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1755 is a set of Job attribute names and/or attribute groups names in whose values the requester is  
1756 interested. This set of attributes is returned for each Job object that is returned. The allowed  
1757 attribute group names are the same as those defined in the Get-Job-Attributes operation in section  
1758 3.3.4. If the client does not supply this attribute, the Printer MUST respond as if the client had  
1759 supplied this attribute with two values: 'job-uri' and 'job-id'.

1760

## 1761 "which-jobs" (type2 keyword):

1762 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1763 indicates which Job objects MUST be returned by the Printer object. The values for this attribute  
1764 are:

1765

1766 'completed': This includes any Job object whose state is 'completed', 'canceled', or 'aborted'.

1767 'not-completed': This includes any Job object whose state is 'pending', 'processing', 'processing-  
1768 stopped', or 'pending-held'.

1769

1770 A Printer object MUST support both values. However, if the implementation does not keep jobs in  
1771 the 'completed', 'canceled', and 'aborted' states, then it returns no jobs when the 'completed' value is  
1772 supplied.

1773

1774 If a client supplies some other value, the Printer object MUST copy the attribute and the  
1775 unsupported value to the Unsupported Attributes response group, reject the request, and return the  
1776 'client-error-attributes-or-values-not-supported' status code.

1777

1778

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

1779

1780

1781

"my-jobs" (boolean):

1782

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

1783

1784

1785

1786

1787

1788

### 3.2.6.2 Get-Jobs Response

1789

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

1790

1791

1792

1793

1794

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

1795

1796

1797

#### Group 1: Operation Attributes

1798

##### Status Message:

1799

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.

1800

1801

1802

1803

##### Natural Language and Character Set:

1804

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

1805

1806

#### Group 2: Unsupported Attributes

1807

See section 3.1.7 for details on returning Unsupported Attributes.

1808

1809

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.

1810

1811

1812

1813

1814

1815

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

1816 The Printer object responds with one set of Job Object Attributes for each returned Job object. The  
1817 Printer object ignores (does not respond with) any requested attribute or value which is not  
1818 supported or which is restricted by the security policy in force, including whether the requesting  
1819 user is the user that submitted the job (job originating user) or not (see section 8). However, the  
1820 Printer object MUST respond with the 'unknown' value for any supported attribute (including all  
1821 REQUIRED attributes) for which the Printer object does not know the value, unless it would violate  
1822 the security policy. See the description of the "out-of-band" values in the beginning of Section 4.1.  
1823

1824 Jobs are returned in the following order:

- 1825 - If the client requests all 'completed' Jobs (Jobs in the 'completed', 'aborted', or 'canceled' states),  
1826 then the Jobs are returned newest to oldest (with respect to actual completion time)
- 1827 - If the client requests all 'not-completed' Jobs (Jobs in the 'pending', 'processing', 'pending-held',  
1828 and 'processing-stopped' states), then Jobs are returned in relative chronological order of  
1829 expected time to complete (based on whatever scheduling algorithm is configured for the  
1830 Printer object).

### 1831 3.2.7 Pause-Printer Operation

1832 This OPTIONAL operation allows a client to stop the Printer object from scheduling jobs on all its devices.  
1833 Depending on implementation, the Pause-Printer operation MAY also stop the Printer from processing the  
1834 current job or jobs. Any job that is currently being printed is either stopped as soon as the implementation  
1835 permits or is completed, depending on implementation. The Printer object MUST still accept create  
1836 operations to create new jobs, but MUST prevent any jobs from entering the 'processing' state.

1837 If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and  
1838 vice-versa.

1839 The IPP Printer stops the current job(s) on its device(s) that were in the 'processing' or 'processing-stopped'  
1840 states as soon as the implementation permits. If the implementation will take appreciable time to stop, the  
1841 IPP Printer adds the 'moving-to-paused' value to the Printer object's "printer-state-reasons" attribute (see  
1842 section 4.4.12). When the device(s) have all stopped, the IPP Printer transitions the Printer object to the  
1843 'stopped' state, removes the 'moving-to-paused' value, if present, and adds the 'paused' value to the Printer  
1844 object's "printer-state-reasons" attribute.

1845 When the current job(s) complete that were in the 'processing' state, the IPP Printer transitions them to the  
1846 'completed' state. When the current job(s) stop in mid processing that were in the 'processing' state, the IPP  
1847 Printer transitions them to the 'processing-stopped' state and adds the 'printer-stopped' value to the job's  
1848 "job-state-reasons" attribute.

1849 For any jobs that are 'pending' or 'pending-held', the 'printer-stopped' value of the jobs' "job-state-reasons"  
1850 attribute also applies. However, the IPP Printer NEED NOT update those jobs' "job-state-reasons"  
1851 attributes and only need return the 'printer-stopped' value when those jobs are queried (so-called "lazy  
1852 evaluation").



1853 Whether the Pause-Printer operation affects jobs that were submitted to the device from other sources than  
 1854 the IPP Printer object in the same way that the Pause-Printer operation affects jobs that were submitted to  
 1855 the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP protocol is being used  
 1856 as a universal management protocol or just to manage IPP jobs, respectively.

1857 The IPP Printer **MUST** accept the request in any state and transition the Printer to the indicated new  
 1858 "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'

1859 *Access Rights:* The authenticated user (see section 8.3) performing this operation must be an operator or  
 1860 administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP Printer **MUST** reject the  
 1861 operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized'  
 1862 as appropriate.

### 1863 3.2.7.1 Pause-Printer Request

1864 The following groups of attributes are part of the Pause-Printer Request:

1865 Group 1: Operation Attributes

1866 Natural Language and Character Set:

1867 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.

1868 Target:

1869 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
 1870 section 3.1.5.

1871 Requesting User Name:

1872 The "requesting-user-name" (name(MAX)) attribute **SHOULD** be supplied by the client as  
 1873 described in section 8.3.

### 1874 3.2.7.2 Pause-Printer Response

1875 The following groups of attributes are part of the Pause-Printer Response:

1876 Group 1: Operation Attributes

1879 Status Message:

1880 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
1881 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
1882 attribute as described in sections 13 and 3.1.6.

1883

1884 Natural Language and Character Set:

1885 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.

1886

1887 Group 2: Unsupported Attributes

1888 See section 3.1.7 for details on returning Unsupported Attributes.

1889

### 1890 3.2.8 Resume-Printer Operation

1891 This operation allows a client to resume the Printer object scheduling jobs on all its devices. The Printer  
1892 object MUST remove the 'paused' and 'moving-to-paused' values from the Printer object's "printer-state-  
1893 reasons" attribute, if present. If there are no other reasons to keep a device paused (such as media-jam), the  
1894 IPP Printer is free to transition itself to the 'processing' or 'idle' states, depending on whether there are jobs  
1895 to be processed or not, respectively, and the device(s) resume processing jobs.

1896 If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and  
1897 vice-versa.

1898 The IPP Printer removes the 'printer-stopped' value from any job's "job-state-reasons" attributes contained  
1899 in that Printer.

1900 The IPP Printer MUST accept the request in any state, transition the Printer object to the indicated new state  
1901 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.

1902 *Access Rights:* The authenticated user (see section 8.3) performing this operation must be an operator or  
1903 administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP Printer MUST reject the  
1904 operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized'  
1905 as appropriate.

1906 The Resume-Printer Request and Resume-Printer Response have the same attribute groups and attributes as  
1907 the Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

### 1908 **3.2.9 Purge-Jobs Operation**

1909 This OPTIONAL operation allows a client to remove all jobs from an IPP Printer object, regardless of their  
1910 job states, including jobs in the Printer object's Job History (see Section 4.3.7.2). After a Purge-Jobs  
1911 operation has been performed, a Printer object MUST return no jobs in subsequent Get-Job-Attributes and  
1912 Get-Jobs responses (until new jobs are submitted).

1913 Whether the Purge-Jobs (and Get-Jobs) operation affects jobs that were submitted to the device from other  
1914 sources than the IPP Printer object in the same way that the Purge-Jobs operation affects jobs that were  
1915 submitted to the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP protocol  
1916 is being used as a universal management protocol or just to manage IPP jobs, respectively.

1917 Note: if an operator wants to cancel all jobs without clearing out the Job History, the operator uses the  
1918 Cancel-Job operation on each job instead of using the Purge-Jobs operation.

1919 The Printer object MUST accept this operation in any state and transition the Printer object to the 'idle'  
1920 state.

1921 *Access Rights:* The authenticated user (see section 8.3) performing this operation must be an operator or  
1922 administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP object MUST reject the  
1923 operation and return: client-error-forbidden, client-error-not-authenticated, and client-error-not-authorized  
1924 as appropriate.

1925 The Purge-Jobs Request and Purge-Jobs Response have the same attribute groups and attributes as the  
1926 Pause-Printer operation (see sections 3.2.7.1 and 3.2.7.2).

1927

## 1928 **3.3 Job Operations**

1929 All Job operations are directed at Job objects. A client MUST always supply some means of identifying the  
1930 Job object in order to identify the correct target of the operation. That job identification MAY either be a  
1931 single Job URI or a combination of a Printer URI with a Job ID. The IPP object implementation MUST  
1932 support both forms of identification for every job.

### 1933 **3.3.1 Send-Document Operation**

1934 This OPTIONAL operation allows a client to create a multi-document Job object that is initially "empty"  
1935 (contains no documents). In the Create-Job response, the Printer object returns the Job object's URI (the  
1936 "job-uri" attribute) and the Job object's 32-bit identifier (the "job-id" attribute). For each new document  
1937 that the client desires to add, the client uses a Send-Document operation. Each Send-Document Request  
1938 contains the entire stream of document data for one document.

1939 If the Printer supports this operation but does not support multiple documents per job, the Printer MUST  
1940 reject subsequent Send-Document operations supplied with data and return the 'server-error-multiple-  
1941 document-jobs-not-supported'. However, the Printer MUST accept the first document with a 'true' or 'false'

1942 value for the "last-document" operation attribute (see below), so that clients MAY always submit one  
1943 document jobs with a 'false' value for "last-document" in the first Send-Document and a 'true' for "last-  
1944 document" in the second Send-Document (with no data).

1945 Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow could  
1946 occur over an arbitrarily long period of time for a particular job, a client MUST send another send operation  
1947 within an IPP Printer defined minimum time interval after the receipt of the previous request for the job. If  
1948 a Printer object supports the Create-Job and Send-Document operations, the Printer object MUST support  
1949 the "multiple-operation-time-out" attribute (see section 4.4.31). This attribute indicates the minimum  
1950 number of seconds the Printer object will wait for the next send operation before taking some recovery  
1951 action.

1952 An IPP object MUST recover from an errant client that does not supply a send operation, sometime after  
1953 the minimum time interval specified by the Printer object's "multiple-operation-time-out" attribute. Such  
1954 recovery MAY include any of the following or other recovery actions:

- 1955 1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', add the  
1956 'aborted-by-system' value to the job's "job-state-reasons" attribute (see section 4.3.8), and clean up  
1957 all resources associated with the Job. In this case, if another send operation is finally received, the  
1958 Printer responds with an "client-error-not-possible" or "client-error-not-found" depending on  
1959 whether or not the Job object is still around when the send operation finally arrives.
- 1960 2. Assume that the last send operation received was in fact the last document (as if the "last-document"  
1961 flag had been set to 'true'), close the Job object, and proceed to process it (i.e., move the Job's state  
1962 to 'pending').
- 1963 3. Assume that the last send operation received was in fact the last document, close the Job, but move it  
1964 to the 'pending-held' and add the 'submission-interrupted' value to the job's "job-state-reasons"  
1965 attribute (see section 4.3.8). This action allows the user or an operator to determine whether to  
1966 continue processing the Job by moving it back to the 'pending' state using the Release-Job operation  
1967 (see section 3.3.6) or to cancel the job using the Cancel-Job operation (see section 3.3.3).

1968

1969 Each implementation is free to decide the "best" action to take depending on local policy, whether any  
1970 documents have been added, whether the implementation spools jobs or not, and/or any other piece of  
1971 information available to it. If the choice is to abort the Job object, it is possible that the Job object may  
1972 already have been processed to the point that some media sheet pages have been printed.

1973 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
1974 owner (as determined in the Create-Job operation) or an operator or administrator of the Printer object (see  
1975 Sections 1 and 8.5). Otherwise, the IPP object MUST reject the operation and return: 'client-error-  
1976 forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

### 1977 3.3.1.1 Send-Document Request

1978 The following attribute sets are part of the Send-Document Request:

1979 Group 1: Operation Attributes

1980 Natural Language and Character Set:  
1981 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.  
1982  
1983 Target:  
1984 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
1985 attribute(s) which define the target for this operation as described in section 3.1.5.  
1986  
1987 Requesting User Name:  
1988 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1989 described in section 8.3.  
1990  
1991 "document-name" (name(MAX)):  
1992 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
1993 contains the client supplied document name. The document name MAY be different than the Job  
1994 name. It might be helpful, but NEED NOT be unique across multiple documents in the same Job.  
1995 Typically, the client software automatically supplies the document name on behalf of the end user  
1996 by using a file name or an application generated name. See the description of the "document-name"  
1997 operation attribute in the Print-Job Request (section 3.2.1.1) for more information about this  
1998 attribute.  
1999  
2000 "compression" (type3 keyword)  
2001 See the description of "compression" for the Print-Job operation in Section 3.2.1.1.  
2002  
2003 "document-format" (mimeMediaType) :  
2004 See the description of "document-format" for the Print-Job operation in Section 3.2.1.1.  
2005  
2006 "document-natural-language" (naturalLanguage):  
2007 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this  
2008 attribute. This attribute specifies the natural language of the document for those document-formats  
2009 that require a specification of the natural language in order to image the document unambiguously.  
2010 There are no particular values required for the Printer object to support.  
2011  
2012 "last-document" (boolean):  
2013 The client MUST supply this attribute. The Printer object MUST support this attribute. It is a  
2014 boolean flag that is set to 'true' if this is the last document for the Job, 'false' otherwise.  
2015  
2016 Group 2: Document Content  
2017 The client MUST supply the document data if the "last-document" flag is set to 'false'. However,  
2018 since a client might not know that the previous document sent with a Send-Document (or Send-  
2019 URI) operation was the last document (i.e., the "last-document" attribute was set to 'false'), it is legal  
2020 to send a Send-Document request with no document data where the "last-document" flag is set to  
2021 'true'. Such a request MUST NOT increment the value of the Job object's "number-of-documents"  
2022 attribute, since no real document was added to the job. It is not an error for a client to submit a job

2023 with no actual document data, i.e., only a single Create-Job and Send-Document request with a  
2024 "last-document" operation attribute set to 'true' with no document data.

### 2025 **3.3.1.2 Send-Document Response**

2026 The following sets of attributes are part of the Send-Document Response:

#### 2027 Group 1: Operation Attributes

##### 2028 Status Message:

2029 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
2030 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
2031 attribute as described in sections 13 and 3.1.6.

2032

##### 2033 Natural Language and Character Set:

2034 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.

2035

#### 2036 Group 2: Unsupported Attributes

2037 See section 3.1.7 for details on returning Unsupported Attributes.

#### 2038 Group 3: Job Object Attributes

2039 This is the same set of attributes as described in the Print-Job response (see section 3.2.1.2).

2040

### 2041 **3.3.2 Send-URI Operation**

2042 This OPTIONAL operation is identical to the Send-Document operation (see section 3.3.1) except that a  
2043 client MUST supply a URI reference ("document-uri" operation attribute) rather than the document data  
2044 itself. If a Printer object supports this operation, clients can use both Send-URI or Send-Document  
2045 operations to add new documents to an existing multi-document Job object. However, if a client needs to  
2046 indicate that the previous Send-URI or Send-Document was the last document, the client MUST use the  
2047 Send-Document operation with no document data and the "last-document" flag set to 'true' (rather than  
2048 using a Send-URI operation with no "document-uri" operation attribute).

2049 If a Printer object supports this operation, it MUST also support the Print-URI operation (see section 3.2.2).

2050 The Printer object MUST validate the syntax and URI scheme of the supplied URI before returning a  
2051 response, just as in the Print-URI operation. The IPP Printer MAY validate the accessibility of the  
2052 document as part of the operation or subsequently (see section 3.2.2).

### 2053 **3.3.3 Cancel-Job Operation**

2054 This REQUIRED operation allows a client to cancel a Print Job from the time the job is created up to the  
2055 time it is completed, canceled, or aborted. Since a Job might already be printing by the time a Cancel-Job is  
2056 received, some media sheet pages might be printed before the job is actually terminated.

2057 The IPP object MUST accept or reject the request based on the job's current state and transition the job to  
 2058 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'

2059 Rule 1: If the implementation requires some measurable time to cancel the job in the 'processing' or  
 2060 'processing-stopped' job states, the IPP object MUST add the 'processing-to-stop-point' value to the job's  
 2061 "job-state-reasons" attribute and then transition the job to the 'canceled' state when the processing ceases  
 2062 (see section 4.3.8).

2063 Rule 2: If the Job object already has the 'processing-to-stop-point' value in its "job-state-reasons" attribute,  
 2064 then the Printer object MUST reject a Cancel-Job operation.

2065 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
 2066 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
 2067 object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
 2068 'client-error-not-authorized' as appropriate.

### 2069 3.3.3.1 Cancel-Job Request

2070 The following groups of attributes are part of the Cancel-Job Request:

#### 2071 Group 1: Operation Attributes

##### 2072 Natural Language and Character Set:

2073 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.  
 2074

##### 2075 Target:

2076 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
 2077 attribute(s) which define the target for this operation as described in section 3.1.5.  
 2078

##### 2079 Requesting User Name:

2080 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
 2081 described in section 8.3.  
 2082

2083 "message" (text(127)):  
2084 The client **OPTIONALLY** supplies this attribute. The Printer object **OPTIONALLY** supports this  
2085 attribute. It is a message to the operator. This "message" attribute is not the same as the "job-  
2086 message-from-operator" attribute. That attribute is used to report a message from the operator to the  
2087 end user that queries that attribute. This "message" operation attribute is used to send a message  
2088 from the client to the operator along with the operation request. It is an implementation decision of  
2089 how or where to display this message to the operator (if at all).  
2090

### 2091 **3.3.3.2 Cancel-Job Response**

2092 The following sets of attributes are part of the Cancel-Job Response:

#### 2093 Group 1: Operation Attributes

##### 2094 Status Message:

2095 In addition to the **REQUIRED** status code returned in every response, the response **OPTIONALLY**  
2096 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
2097 attribute as described in sections 13 and 3.1.6.  
2098

##### 2099 Natural Language and Character Set:

2100 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.  
2101

#### 2102 Group 2: Unsupported Attributes

2103 See section 3.1.7 for details on returning Unsupported Attributes.  
2104

2105 Once a successful response has been sent, the implementation guarantees that the Job will eventually end up  
2106 in the 'canceled' state. Between the time of the Cancel-Job operation is accepted and when the job enters the  
2107 'canceled' job-state (see section 4.3.7), the "job-state-reasons" attribute **SHOULD** contain the 'processing-to-  
2108 stop-point' value which indicates to later queries that although the Job might still be 'processing', it will  
2109 eventually end up in the 'canceled' state, not the 'completed' state.

### 2110 **3.3.4 Get-Job-Attributes Operation**

2111 This **REQUIRED** operation allows a client to request the values of attributes of a Job object and it is almost  
2112 identical to the Get-Printer-Attributes operation (see section 3.2.5). The only differences are that the  
2113 operation is directed at a Job object rather than a Printer object, there is no "document-format" operation  
2114 attribute used when querying a Job object, and the returned attribute group is a set of Job object attributes  
2115 rather than a set of Printer object attributes.

2116 For Jobs, the possible names of attribute groups are:

- 2117 - 'job-template': the subset of the Job Template attributes that apply to a Job object (the first column of  
2118 the table in Section 4.2) that the implementation supports for Job objects.



- 2119 - 'job-description': the subset of the Job Description attributes specified in Section 4.3 that the  
2120 implementation supports for Job objects.  
2121 - 'all': the special group 'all' that includes all attributes that the implementation supports for Job objects.  
2122

2123 Since a client MAY request specific attributes or named groups, there is a potential that there is some  
2124 overlap. For example, if a client requests, 'job-name' and 'job-description', the client is actually requesting  
2125 the "job-name" attribute once by naming it explicitly, and once by inclusion in the 'job-description' group.  
2126 In such cases, the Printer object NEED NOT return the attribute only once in the response even if it is  
2127 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

2128 It is NOT REQUIRED that a Job object support all attributes belonging to a group (since some attributes  
2129 are OPTIONAL). However it is REQUIRED that each Job object support all group names.

### 2130 3.3.4.1 Get-Job-Attributes Request

2131 The following groups of attributes are part of the Get-Job-Attributes Request when the request is directed at  
2132 a Job object:

2133 Group 1: Operation Attributes

2134 Natural Language and Character Set:

2135 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.  
2136

2137 Target:

2138 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
2139 attribute(s) which define the target for this operation as described in section 3.1.5.  
2140

2141 Requesting User Name:

2142 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
2143 described in section 8.3.  
2144

2145 "requested-attributes" (1setOf keyword) :

2146 The client OPTIONALLY supplies this attribute. The IPP object MUST support this attribute. It is  
2147 a set of attribute names and/or attribute group names in whose values the requester is interested. If  
2148 the client omits this attribute, the IPP object MUST respond as if this attribute had been supplied  
2149 with a value of 'all'.  
2150

### 2151 3.3.4.2 Get-Job-Attributes Response

2152 The Printer object returns the following sets of attributes as part of the Get-Job-Attributes Response:

2153 Group 1: Operation Attributes

2154 Status Message:

2155 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY  
2156 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation  
2157 attribute as described in sections 13 and 3.1.6.

2158

2159 Natural Language and Character Set:

2160 The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.  
2161 The "attributes-natural-language" MAY be the natural language of the Job object, rather than the  
2162 one requested.

2163

2164 Group 2: Unsupported Attributes

2165 See section 3.1.7 for details on returning Unsupported Attributes.

2166

2167 The response NEED NOT contain the "requested-attributes" operation attribute with any supplied  
2168 values (attribute keywords) that were requested by the client but are not supported by the IPP object.  
2169 If the Printer object does include unsupported attributes referenced in "requested-attributes" and  
2170 such attributes include group names, such as 'all', the unsupported attributes MUST NOT include  
2171 attributes described in the standard but not supported by the implementation.

2172

2173 Group 3: Job Object Attributes

2174 This is the set of requested attributes and their current values. The IPP object ignores (does not  
2175 respond with) any requested attribute or value which is not supported or which is restricted by the  
2176 security policy in force, including whether the requesting user is the user that submitted the job (job  
2177 originating user) or not (see section 8). However, the IPP object MUST respond with the 'unknown'  
2178 value for any supported attribute (including all REQUIRED attributes) for which the IPP object does  
2179 not know the value, unless it would violate the security policy. See the description of the "out-of-  
2180 band" values in the beginning of Section 4.1.

### 2181 3.3.5 Hold-Job Operation

2182 This OPTIONAL operation allows a client to hold a pending job in the queue so that it is not eligible for  
2183 scheduling. If the Hold-Job operation is supported, then the Release-Job operation MUST be supported,  
2184 and vice-versa. The OPTIONAL "job-hold-until" operation attribute allows a client to specify whether to  
2185 hold the job indefinitely or until a specified time period, if supported.

2186 The IPP object MUST accept or reject the request based on the job's current state and transition the job to  
2187 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'

2188 Rule 1: If the implementation supports multiple reasons for a job to be in the 'pending-held' state, the IPP  
2189 object MUST add the 'job-hold-until-specified' value to the job's "job-state-reasons" attribute.

2190 Rule 2: If the IPP object supports the "job-hold-until" operation attribute, but the specified time period has  
2191 already started (or is the 'no-hold' value) and there are no other reasons to hold the job, the IPP object  
2192 MUST make the job be a candidate for processing immediately (see Section 4.2.2) by putting the job in the  
2193 'pending' state.

2194 Note: In order to keep the Hold-Job operation simple, such a request is rejected when the job is in the  
2195 'processing' or 'processing-stopped' states. If an operation is needed to hold jobs while in these states, it will  
2196 be added as an additional operation, rather than overloading the Hold-Job operation. Then it is clear to  
2197 clients by querying the Printer object's "operations-supported" (see Section 4.4.15) and the Job object's  
2198 "job-state" (see Section 4.3.7) attributes which operations are possible.

2199 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
2200 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
2201 object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
2202 'client-error-not-authorized' as appropriate.

### 2203 3.3.5.1 Hold-Job Request

2204 The groups and operation attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with the  
2205 addition of the following Group 1 Operation attribute:

2206 "job-hold-until" (type3 keyword | name(MAX)):

2207 The client OPTIONALLY supplies this Operation attribute. The IPP object MUST support this  
2208 operation attribute in a Hold-Job request, if it supports the "job-hold-until" Job template attribute in  
2209 create operations. See section 4.2.2. The IPP object SHOULD support the "job-hold-until" Job  
2210 Template attribute for use in job create operations with at least the 'indefinite' value, if it supports  
2211 the Hold-Job operation. Otherwise, a client cannot create a job and hold it immediately (without  
2212 picking some supported time period in the future).

2213 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP  
2214 object copies the supplied operation attribute to the Job object, replacing the job's previous "job-

2215 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied  
2216 named time period.

2217 If supplied, but either the "job-hold-until" Operation attribute itself or the value supplied is not  
2218 supported, the IPP object accepts the request, returns the unsupported attribute or value in the  
2219 Unsupported Attributes Group according to section 3.1.7, returns the 'successful-ok-ignored-or-  
2220 substituted-attributes, and holds the job indefinitely until a client performs a subsequent Release-Job  
2221 operation.

2222 If the client (1) supplies a value that specifies a time period that has already started or the 'no-hold'  
2223 value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until" operation  
2224 attribute and there are no other reasons to hold the job, the IPP object **MUST** accept the operation  
2225 and make the job be a candidate for processing immediately (see Section 4.2.2).

2226 If the client does not supply a "job-hold-until" Operation attribute in the request, the IPP object  
2227 **MUST** populate the job object with a "job-hold-until" attribute with the 'indefinite' value (if IPP  
2228 object supports the "job-hold-until" attribute) and hold the job indefinitely, until a client performs a  
2229 Release-Job operation.

### 2230 **3.3.5.2 Hold-Job Response**

2231 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

### 2232 **3.3.6 Release-Job Operation**

2233 This **OPTIONAL** operation allows a client to release a previously held job so that it is again eligible for  
2234 scheduling. If the Hold-Job operation is supported, then the Release-Job operation **MUST** be supported,  
2235 and vice-versa.

2236 This operation removes the "job-hold-until" job attribute, if present, from the job object that had been  
2237 supplied in the create or most recent Hold-Job or Restart-Job operation and removes its effect on the job.  
2238 The IPP object **MUST** remove the 'job-hold-until-specified' value from the job's "job-state-reasons"  
2239 attribute, if present. See section 4.3.8.

2240 The IPP object **MUST** accept or reject the request based on the job's current state and transition the job to  
2241 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'

2242 Rule 1: If there are other reasons to keep the job in the 'pending-held' state, such as 'resources-are-not-  
2243 ready', the job remains in the 'pending-held' state. Thus the 'pending-held' state is not just for jobs that have  
2244 the 'job-hold-until' applied to them, but are for any reason to keep the job from being a candidate for  
2245 scheduling and processing, such as 'resources-are-not-ready'. See the "job-hold-until" attribute (section  
2246 4.2.2).

2247 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
2248 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
2249 object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
2250 'client-error-not-authorized' as appropriate.

2251 The Release-Job Request and Release-Job Response have the same attribute groups and attributes as the  
2252 Cancel-Job operation (see section 3.3.3.1 and 3.3.3.2).

### 2253 3.3.7 Restart-Job Operation

2254 This OPTIONAL operation allows a client to restart a job that is retained in the queue after processing has  
2255 completed (see section 4.3.7.2).

2256 The job is moved to the 'pending' or 'pending-held' job state and restarts at the beginning on the same IPP  
2257 Printer object with the same attribute values. If any of the documents in the job were passed by reference  
2258 (Print-URI or Send-URI), the Printer MUST re-fetch the data, since the semantics of Restart-Job are to  
2259 repeat all Job processing. The Job Description attributes that accumulate job progress, such as "job-  
2260 impressions-completed", "job-media-sheets-completed", and "job-k-octets-processed", MUST be reset to 0  
2261 so that they give an accurate record of the job from its restart point. The job object MUST continue to use  
2262 the same "job-uri" and "job-id" attribute values.

2263 Note: If in the future an operation is needed that does not reset the job progress attributes, then a new  
2264 operation will be defined which makes a copy of the job, assigns a new "job-uri" and "job-id" to the copy  
2265 and resets the job progress attributes in the new copy only.

2266 The IPP object MUST accept or reject the request based on the job's current state, transition the job to the  
2267 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' or 'pending-held'	'successful-ok' - job is started over.
'completed'	'completed'	'client-error-not-possible' - see Rule 1
'canceled'	'pending' or 'pending-held'	'successful-ok' - job is started over.
'canceled'	'canceled'	'client-error-not-possible' - see Rule 1
'aborted'	'pending' or 'pending-held'	'successful-ok' - job is started over.
'aborted'	'aborted'	'client-error-not-possible' - see Rule 1

2268

2269 Rule 1: If the Job Retention Period has expired for the job in this state, then the IPP object rejects the  
2270 operation. See section 4.3.7.2.

2271 Note: In order to prevent a user from inadvertently restarting a job in the middle, the Restart-Job request is  
2272 rejected when the job is in the 'processing' or 'processing-stopped' states. If in the future an operation is  
2273 needed to hold or restart jobs while in these states, it will be added as an additional operation, rather than  
2274 overloading the Restart-Job operation, so that it is clear that the user intended that the current job not be  
2275 completed.

2276 *Access Rights:* The authenticated user (see section 8.3) performing this operation must either be the job  
2277 owner or an operator or administrator of the Printer object (see Sections 1 and 8.5). Otherwise, the IPP  
2278 object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or  
2279 'client-error-not-authorized' as appropriate.

### 2280 3.3.7.1 Restart-Job Request

2281 The groups and attributes are the same as for a Cancel-Job request (see section 3.3.3.1), with the addition of  
2282 the following Group 1 Operation attribute:

2283 "job-hold-until" (type3 keyword | name(MAX)):

2284 The client OPTIONALLY supplies this attribute. The IPP object MUST support this Operation  
2285 attribute in a Restart-Job request, if it supports the "job-hold-until" Job Template attribute in create  
2286 operations. See section 4.2.2. Otherwise, the IPP object NEED NOT support the "job-hold-until"  
2287 Operation attribute in a Restart-Job request.

2288 If supplied and supported as specified in the Printer's "job-hold-until-supported" attribute, the IPP  
2289 object copies the supplied Operation attribute to the Job object, replacing the job's previous "job-  
2290 hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied  
2291 named time period. See section 4.2.2.

2292 If supplied, but the value is not supported, the IPP object accepts the request, returns the  
2293 unsupported attribute or value in the Unsupported Attributes Group according to section 3.1.7,

2294 returns the 'successful-ok-ignored-or-substituted-attributes' status code, and holds the job  
2295 indefinitely until a client performs a subsequent Release-Job operation.

2296 If supplied, but the "job-hold-until" Operation attribute itself is not supported, the IPP object accepts  
2297 the request, returns the unsupported attribute with the out-of-band 'unsupported' value in the  
2298 Unsupported Attributes Group according to section 3.1.7, returns the 'successful-ok-ignored-or-  
2299 substituted-attributes' status code, and restarts the job, i.e., ignores the "job-hold-until" attribute.

2300 If the client (1) supplies a value that specifies a time period that has already started or the 'no-hold'  
2301 value (meaning don't hold the job) and (2) the IPP object supports the "job-hold-until" operation  
2302 attribute and there are no other reasons to hold the job, the IPP object makes the job a candidate for  
2303 processing immediately (see Section 4.2.2).

2304 If the client does not supply a "job-hold-until" operation attribute in the request, the IPP object  
2305 removes the "job-hold-until" attribute, if present, from the job. If there are no other reasons to hold  
2306 the job, the Restart-Job operation makes the job a candidate for processing immediately (see Section  
2307 4.2.2).

### 2308 3.3.7.2 Restart-Job Response

2309 The groups and attributes are the same as for a Cancel-Job response (see section 3.3.3.2).

2310 Note: In the future an OPTIONAL Modify-Job or Set-Job-Attributes operation may be specified that  
2311 allows the client to modify other attributes before releasing the restarted job.

## 2312 4. Object Attributes

2313 This section describes the attributes with their corresponding attribute syntaxes and values that are part of  
2314 the IPP model. The sections below show the objects and their associated attributes which are included  
2315 within the scope of this protocol. Many of these attributes are derived from other relevant documents:

- 2316 - Document Printing Application (DPA) [ISO10175]
- 2317 - RFC 1759 Printer MIB [RFC1759]

2318 Each attribute is uniquely identified in this document using a "keyword" (see section 12.2.1) which is the  
2319 name of the attribute. The keyword is included in the section header describing that attribute.

2320 Note: Not only are keywords used to identify attributes, but one of the attribute syntaxes described below is  
2321 "keyword" so that some attributes have keyword values. Therefore, these attributes are defined as having  
2322 an attribute syntax that is a set of keywords.

## 2323 4.1 Attribute Syntaxes

2324 This section defines the basic attribute syntax types that all clients and IPP objects MUST be able to accept  
2325 in responses and accept in requests, respectively. Each attribute description in sections 3 and 4 includes the  
2326 name of attribute syntax(es) in the heading (in parentheses). A conforming implementation of an attribute  
2327 MUST include the semantics of the attribute syntax(es) so identified. Section 6.3 describes how the  
2328 protocol can be extended with new attribute syntaxes.

2329 The attribute syntaxes are specified in the following sub-sections, where the sub-section heading is the  
2330 keyword name of the attribute syntax inside the single quotes. In operation requests and responses each  
2331 attribute value MUST be represented as one of the attribute syntaxes specified in the sub-section heading  
2332 for the attribute. In addition, the value of an attribute in a response (but not in a request) MAY be one of  
2333 the "out-of-band" values whose special encoding rules are defined in the "Encoding and Transport"  
2334 document [IPP-PRO]. Standard "out-of-band" values are:

2335 'unknown': The attribute is supported by the IPP object, but the value is unknown to the IPP object for  
2336 some reason.

2337 'unsupported': The attribute is unsupported by the IPP object. This value MUST be returned only as the  
2338 value of an attribute in the Unsupported Attributes Group.

2339 'no-value': The attribute is supported by the Printer object, but the administrator has not yet configured a  
2340 value.

2341

2342 All attributes in a request MUST have one or more values as defined in Sections 4.2 to 4.4. Thus clients  
2343 MUST NOT supply attributes with "out-of-band" values for operations defined in this document. All  
2344 attributes in a response MUST have one or more values as defined in Sections 4.2 to 4.4 or a single "out-of-  
2345 band" value.

2346 Most attributes are defined to have a single attribute syntax. However, a few attributes (e.g., "job-sheet",  
2347 "media", "job-hold-until") are defined to have several attribute syntaxes, depending on the value. These  
2348 multiple attribute syntaxes are separated by the "|" character in the sub-section heading to indicate the  
2349 choice. Since each value MUST be tagged as to its attribute syntax in the protocol, a single-valued attribute  
2350 instance may have any one of its attribute syntaxes and a multi-valued attribute instance may have a mixture  
2351 of its defined attribute syntaxes.

### 2352 4.1.1 'text'

2353 A text attribute is an attribute whose value is a sequence of zero or more characters encoded in a maximum  
2354 of 1023 ('MAX') octets. MAX is the maximum length for each value of any text attribute. However, if an  
2355 attribute will always contain values whose maximum length is much less than MAX, the definition of that  
2356 attribute will include a qualifier that defines the maximum length for values of that attribute. For example:  
2357 the "printer-location" attribute is specified as "printer-location (text(127))". In this case, text values for  
2358 "printer-location" MUST NOT exceed 127 octets; if supplied with a longer text string via some external  
2359 interface (other than the protocol), implementations are free to truncate to this shorter length limitation.



2360 In this document, all text attributes are defined using the 'text' syntax. However, 'text' is used only for  
2361 brevity; the formal interpretation of 'text' is: 'textWithoutLanguage | textWithLanguage'. That is, for any  
2362 attribute defined in this document using the 'text' attribute syntax, all IPP objects and clients MUST support  
2363 both the 'textWithoutLanguage' and 'textWithLanguage' attribute syntaxes. However, in actual usage and  
2364 protocol execution, objects and clients accept and return only one of the two syntax per attribute. The  
2365 syntax 'text' never appears "on-the-wire".

2366 Both 'textWithoutLanguage' and 'textWithLanguage' are needed to support the real world needs of  
2367 interoperability between sites and systems that use different natural languages as the basis for human  
2368 communication. Generally, one natural language applies to all text attributes in a given request or response.  
2369 The language is indicated by the "attributes-natural-language" operation attribute defined in section 3.1.4 or  
2370 "attributes-natural-language" job attribute defined in section 4.3.20, and there is no need to identify the  
2371 natural language for each text string on a value-by-value basis. In these cases, the attribute syntax  
2372 'textWithoutLanguage' is used for text attributes. In other cases, the client needs to supply or the Printer  
2373 object needs to return a text value in a natural language that is different from the rest of the text values in  
2374 the request or response. In these cases, the client or Printer object uses the attribute syntax  
2375 'textWithLanguage' for text attributes (this is the Natural Language Override mechanism described in  
2376 section 3.1.4).

2377 The 'textWithoutLanguage' and 'textWithLanguage' attribute syntaxes are described in more detail in the  
2378 following sections.

#### 2379 **4.1.1.1 'textWithoutLanguage'**

2380 The 'textWithoutLanguage' syntax indicates a value that is sequence of zero or more characters encoded in a  
2381 maximum of 1023 (MAX) octets. Text strings are encoded using the rules of some charset. The Printer  
2382 object MUST support the UTF-8 charset [RFC2279] and MAY support additional charsets to represent  
2383 'text' values, provided that the charsets are registered with IANA [IANA-CS]. See Section 4.1.7 for the  
2384 definition of the 'charset' attribute syntax, including restricted semantics and examples of charsets.

#### 2385 **4.1.1.2 'textWithLanguage'**

2386 The 'textWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2387 'textWithoutLanguage' part encoded in a maximum of 1023 (MAX) octets plus an additional  
2388 'naturalLanguage' (see section 4.1.8) part that overrides the natural language in force. The  
2389 'naturalLanguage' part explicitly identifies the natural language that applies to the text part of that value and  
2390 that value alone. For any give text attribute, the 'textWithoutLanguage' part is limited to the maximum  
2391 length defined for that 'text' attribute, and the 'naturalLanguage' part is always limited to 63 (additional)  
2392 octets. Using the 'textWithLanguage' attribute syntax rather than the normal 'textWithoutLanguage' syntax  
2393 is the so-called Natural Language Override mechanism and MUST be supported by all IPP objects and  
2394 clients.

2395 If the attribute is multi-valued (1setOf text), then the 'textWithLanguage' attribute syntax MUST be used to  
2396 explicitly specify each attribute value whose natural language needs to be overridden. Other values in a

2397 multi-valued 'text' attribute in a request or a response revert to the natural language of the operation  
2398 attribute.

2399 In a create request, the Printer object **MUST** accept and store with the Job object any natural language in the  
2400 "attributes-natural-language" operation attribute, whether the Printer object supports that natural language  
2401 or not. Furthermore, the Printer object **MUST** accept and store any 'textWithLanguage' attribute value,  
2402 whether the Printer object supports that natural language or not. These requirements are independent of the  
2403 value of the "ipp-attribute-fidelity" operation attribute that the client **MAY** supply.

2404 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2405 indicating English, but the value of the "job-name" attribute is in French, the client **MUST** use the  
2406 'textWithLanguage' attribute syntax with the following two values:

2407 'fr': Natural Language Override indicating French

2408 'Rapport Mensuel': the job name in French

2409

2410 See the "Encoding and Transport" document [IPP-PRO] section 3.11 for the encoding of the two parts and  
2411 Appendix A for a detailed example of the 'textWithLanguage' attribute syntax.

#### 2412 4.1.2 'name'

2413 This syntax type is used for user-friendly strings, such as a Printer name, that, for humans, are more  
2414 meaningful than identifiers. Names are never translated from one natural language to another. The 'name'  
2415 attribute syntax is essentially the same as 'text', including the **REQUIRED** support of UTF-8 except that the  
2416 sequence of characters is limited so that its encoded form **MUST NOT** exceed 255 (MAX) octets.

2417 Also like 'text', 'name' is really an abbreviated notation for either 'nameWithoutLanguage' or  
2418 'nameWithLanguage'. That is, all IPP objects and clients **MUST** support both the 'nameWithoutLanguage'  
2419 and 'nameWithLanguage' attribute syntaxes. However, in actual usage and protocol execution, objects and  
2420 clients accept and return only one of the two syntax per attribute. The syntax 'name' never appears "on-the-  
2421 wire".

2422 Only the 'text' and 'name' attribute syntaxes permit the Natural Language Override mechanism.

2423 Some attributes are defined as 'type3 keyword | name'. These attributes support values that are either type3  
2424 keywords or names. This dual-syntax mechanism enables a site administrator to extend these attributes to  
2425 legally include values that are locally defined by the site administrator. Such names are not registered with  
2426 IANA.

##### 2427 4.1.2.1 'nameWithoutLanguage'

2428 The nameWithoutLanguage' syntax indicates a value that is sequence of zero or more characters encoded in  
2429 a maximum of 255 (MAX) octets.

#### 2430 4.1.2.2 'nameWithLanguage'

2431 The 'nameWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2432 'nameWithoutLanguage' part encoded in a maximum of 1023 (MAX) octets plus an additional  
2433 'naturalLanguage' (see section 4.1.8) part that overrides the natural language in force. The  
2434 'naturalLanguage' part explicitly identifies the natural language that applies to that name value and that  
2435 name value alone. For any give text attribute, the 'textWithoutLanguage' part is limited to the maximum  
2436 length defined for that 'text' attribute, and the 'naturalLanguage' part is always limited to 63 (additional)  
2437 octets. Using the 'textWithLanguage' attribute syntax rather than the normal 'textWithoutLanguage' syntax  
2438 is the so-called Natural Language Override mechanism and MUST be supported by all IPP objects and  
2439 clients.

2440 The 'nameWithLanguage' attribute syntax behaves the same as the 'textWithLanguage' syntax. Using the  
2441 'textWithLanguage' attribute syntax rather than the normal 'textWithoutLanguage' syntax is the so-called  
2442 Natural Language Override mechanism and MUST be supported by all IPP objects and clients. If a name is  
2443 in a language that is different than the rest of the object or operation, then this 'nameWithLanguage' syntax  
2444 is used rather than the generic 'nameWithoutLanguage' syntax.

2445 If the attribute is multi-valued (1setOf text), then the 'nameWithLanguage' attribute syntax MUST be used  
2446 to explicitly specify each attribute value whose natural language needs to be overridden. Other values in a  
2447 multi-valued 'name' attribute in a request or a response revert to the natural language of the operation  
2448 attribute.

2449 In a create request, the Printer object MUST accept and store with the Job object any natural language in the  
2450 "attributes-natural-language" operation attribute, whether the Printer object supports that natural language  
2451 or not. Furthermore, the Printer object MUST accept and store any 'nameWithLanguage' attribute value,  
2452 whether the Printer object supports that natural language or not. These requirements are independent of the  
2453 value of the "ipp-attribute-fidelity" operation attribute that the client MAY supply.

2454 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2455 indicating English, but the "printer-name" attribute is in German, the client MUST use the  
2456 'nameWithLanguage' attribute syntax as follows:

2457 'de': Natural Language Override indicating German

2458 'Farbdrucker': the Printer name in German

2459

2460 See the "Encoding and Transport" document [IPP-PRO] section 3.11 for the encoding of the two parts and  
2461 Appendix A for a detailed example of the 'nameWithLanguage' attribute syntax.

#### 2462 4.1.2.3 Matching 'name' attribute values

2463 For purposes of matching two 'name' attribute values for equality, such as in job validation (where a client-  
2464 supplied value for attribute "xxx" is checked to see if the value is among the values of the Printer object's  
2465 corresponding "xxx-supported" attribute), the following match rules apply:

2466 1. 'keyword' values never match 'name' values.

2467 2. 'name' (nameWithoutLanguage and nameWithLanguage) values match if (1) the name parts  
2468 match and (2) the Associated Natural-Language parts (see section 3.1.4.1) match. The matching  
2469 rules are:

2470 a. the name parts match if the two names are identical character by character, except it is  
2471 RECOMMENDED that case be ignored. For example: 'Ajax-letter-head-white' MUST  
2472 match 'Ajax-letter-head-white' and SHOULD match 'ajax-letter-head-white' and 'AJAX-  
2473 LETTER-HEAD-WHITE'.

2474 b. the Associated Natural-Language parts match if the shorter of the two meets the syntactic  
2475 requirements of RFC 1766 [RFC1766] and matches byte for byte with the longer. For  
2476 example, 'en' matches 'en', 'en-us' and 'en-gb', but matches neither 'fr' nor 'e'.

#### 2477 4.1.3 'keyword'

2478 The 'keyword' attribute syntax is a sequence of characters, length: 1 to 255, containing only the US-ASCII  
2479 [ASCII] encoded values for lowercase letters ("a" - "z"), digits ("0" - "9"), hyphen ("-"), dot ("."), and  
2480 underscore ("\_"). The first character MUST be a lowercase letter. Furthermore, keywords MUST be in  
2481 U.S. English.

2482 This syntax type is used for enumerating semantic identifiers of entities in the abstract protocol, i.e., entities  
2483 identified in this document. Keywords are used as attribute names or values of attributes. Unlike 'text' and  
2484 'name' attribute values, 'keyword' values MUST NOT use the Natural Language Override mechanism, since  
2485 they MUST always be US-ASCII and U.S. English.

2486 Keywords are for use in the protocol. A user interface will likely provide a mapping between protocol  
2487 keywords and displayable user-friendly words and phrases which are localized to the natural language of  
2488 the user. While the keywords specified in this document MAY be displayed to users whose natural  
2489 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since  
2490 the user interface is outside the scope of this document.

2491 In the definition for each attribute of this syntax type, the full set of defined keyword values for that  
2492 attribute are listed.

2493 When a keyword is used to represent an attribute (its name), it MUST be unique within the full scope of all  
2494 IPP objects and attributes. When a keyword is used to represent a value of an attribute, it MUST be unique  
2495 just within the scope of that attribute. That is, the same keyword MUST NOT be used for two different  
2496 values within the same attribute to mean two different semantic ideas. However, the same keyword MAY  
2497 be used across two or more attributes, representing different semantic ideas for each attribute. Section 6.1  
2498 describes how the protocol can be extended with new keyword values. Examples of attribute name  
2499 keywords:

2500 "job-name"  
2501 "attributes-charset"  
2502

2503 Note: This document uses "type1", "type2", and "type3" prefixes to the "keyword" basic syntax to indicate  
2504 different levels of review for extensions (see section 6.1).

#### 2505 **4.1.4 'enum'**

2506 The 'enum' attribute syntax is an enumerated integer value that is in the range from 1 to  $2^{31} - 1$  (MAX).  
2507 Each value has an associated 'keyword' name. In the definition for each attribute of this syntax type, the full  
2508 set of possible values for that attribute are listed. This syntax type is used for attributes for which there are  
2509 enum values assigned by other standards, such as SNMP MIBs. A number of attribute enum values in this  
2510 document are also used for corresponding attributes in other standards [RFC1759]. This syntax type is not  
2511 used for attributes to which the administrator may assign values. Section 6.1 describes how the protocol  
2512 can be extended with new enum values.

2513 Enum values are for use in the protocol. A user interface will provide a mapping between protocol enum  
2514 values and displayable user-friendly words and phrases which are localized to the natural language of the  
2515 user. While the enum symbols specified in this document MAY be displayed to users whose natural  
2516 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since  
2517 the user interface is outside the scope of this document.

2518 Note: SNMP MIBs use '2' for 'unknown' which corresponds to the IPP "out-of-band" value 'unknown'. See  
2519 the description of the "out-of-band" values at the beginning of Section 4.1. Therefore, attributes of type  
2520 'enum' start at '3'.

2521 Note: This document uses "type1", "type2", and "type3" prefixes to the "enum" basic syntax to indicate  
2522 different levels of review for extensions (see section 6.1).

#### 2523 **4.1.5 'uri'**

2524 The 'uri' attribute syntax is any valid Uniform Resource Identifier or URI [RFC2396]. Most often, URIs are  
2525 simply Uniform Resource Locators or URLs. The maximum length of URIs used as values of IPP  
2526 attributes is 1023 octets. Although most other IPP attribute syntax types allow for only lower-cased values,  
2527 this attribute syntax type conforms to the case-sensitive and case-insensitive rules specified in [RFC2396].  
2528 See also [IPP-IIG] for a discussion of case in URIs.

#### 2529 **4.1.6 'uriScheme'**

2530 The 'uriScheme' attribute syntax is a sequence of characters representing a URI scheme according to RFC  
2531 2396 [RFC2396]. Though RFC 2396 requires that the values be case-insensitive, IPP requires all lower  
2532 case values in IPP attributes to simplify comparing by IPP clients and Printer objects.

2533 Standard values for this syntax type are the following keywords:

2534 'ipp': for IPP schemed URIs (e.g., "ipp:...")  
2535 'http': for HTTP schemed URIs (e.g., "http:...")  
2536 'https': for use with HTTPS schemed URIs (e.g., "https:...") (not on IETF standards track)  
2537 'ftp': for FTP schemed URIs (e.g., "ftp:...")

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

2539 'file': for file schemed URIs (e.g., "file:...")

2540

2541 A Printer object MAY support any URI 'scheme' that has been registered with IANA [IANA-MT]. The  
2542 maximum length of URI 'scheme' values used to represent IPP attribute values is 63 octets.

#### 2543 4.1.7 'charset'

2544 The 'charset' attribute syntax is a standard identifier for a charset. A charset is a coded character set and  
2545 encoding scheme. Charsets are used for labeling certain document contents and 'text' and 'name' attribute  
2546 values. The syntax and semantics of this attribute syntax are specified in RFC 2046 [RFC2046] and  
2547 contained in the IANA character-set Registry [IANA-CS] according to the IANA procedures [RFC2278].  
2548 Though RFC 2046 requires that the values be case-insensitive US-ASCII, IPP requires all lower case values  
2549 in IPP attributes to simplify comparing by IPP clients and Printer objects. When a character-set in the  
2550 IANA registry has more than one name (alias), the name labeled as "(preferred MIME name)", if present,  
2551 MUST be used.

2552 The maximum length of 'charset' values used to represent IPP attribute values is 63 octets.

2553 Some examples are:

2554 'utf-8': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as the UTF-8  
2555 [RFC2279] transfer encoding scheme in which US-ASCII is a subset charset.

2556 'us-ascii': 7-bit American Standard Code for Information Interchange (ASCII), ANSI X3.4-1986  
2557 [ASCII]. That standard defines US-ASCII, but RFC 2045 [RFC2045] eliminates most of the control  
2558 characters from conformant usage in MIME and IPP.

2559 'iso-8859-1': 8-bit One-Byte Coded Character Set, Latin Alphabet Nr 1 [ISO8859-1]. That standard  
2560 defines a coded character set that is used by Latin languages in the Western Hemisphere and  
2561 Western Europe. US-ASCII is a subset charset.

2562

2563 Some attribute descriptions MAY place additional requirements on charset values that may be used, such as  
2564 REQUIRED values that MUST be supported or additional restrictions, such as requiring that the charset  
2565 have US-ASCII as a subset charset.

#### 2566 4.1.8 'naturalLanguage'

2567 The 'naturalLanguage' attribute syntax is a standard identifier for a natural language and optionally a  
2568 country. The values for this syntax type are defined by RFC 1766 [RFC1766]. Though RFC 1766 requires  
2569 that the values be case-insensitive US-ASCII, IPP requires all lower case to simplify comparing by IPP  
2570 clients and Printer objects. Examples include:

2571 'en': for English

2572 'en-us': for US English

2573 'fr': for French

2574 'de': for German

2575

2576 The maximum length of 'naturalLanguage' values used to represent IPP attribute values is 63 octets.

#### 2577 **4.1.9 'mimeMediaType'**

2578 The 'mimeMediaType' attribute syntax is the Internet Media Type (sometimes called MIME type) as  
2579 defined by RFC 2046 [RFC2046] and registered according to the procedures of RFC 2048 [RFC2048] for  
2580 identifying a document format. The value MAY include a charset, or other, parameter, depending on the  
2581 specification of the Media Type in the IANA Registry [IANA-MT]. Although most other IPP syntax types  
2582 allow for only lower-cased values, this syntax type allows for mixed-case values which are case-insensitive.

2583 Examples are:

2584 'text/html': An HTML document

2585 'text/plain': A plain text document in US-ASCII (RFC 2046 indicates that in the absence of the charset  
2586 parameter MUST mean US-ASCII rather than simply unspecified) [RFC2046].

2587 'text/plain; charset=US-ASCII': A plain text document in US-ASCII [52, 56].

2588 'text/plain; charset=ISO-8859-1': A plain text document in ISO 8859-1 (Latin 1) [ISO8859-1].

2589 'text/plain; charset=utf-8': A plain text document in ISO 10646 represented as UTF-8 [RFC2279]

2590 'application/postscript': A PostScript document [RFC2046]

2591 'application/vnd.hp-PCL': A PCL document [IANA-MT] (charset escape sequence embedded in the  
2592 document data)

2593 'application/pdf': Portable Document Format - see IANA MIME Media Type registry

2594 'application/octet-stream': Auto-sense - see section 4.1.9.1

2595 The maximum length of a 'mimeMediaType' value to represent IPP attribute values is 255 octets.

#### 2596 **4.1.9.1 Application/octet-stream -- Auto-Sensing the document format**

2597 One special type is 'application/octet-stream'. If the Printer object supports this value, the Printer object  
2598 MUST be capable of auto-sensing the format of the document data using an implementation-dependent  
2599 method that examines some number of octets 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 When a Printer performs auto-sensing of a document in a submitted job, it is RECOMMENDED that the  
2629 Printer indicate to the user that such auto-sensing has occurred and which document-format was auto-  
2630 sensed by printing that information on the job's job-start-sheet.

#### 2631 **4.1.10 'octetString'**

2632 The 'octetString' attribute syntax is a sequence of octets encoded in a maximum of 1023 octets which is  
2633 indicated in sub-section headers using the notation: octetString(MAX). This syntax type is used for opaque  
2634 data.

#### 2635 **4.1.11 'boolean'**

2636 The 'boolean' attribute syntax has only two values: 'true' and 'false'.

#### 2637 **4.1.12 'integer'**

2638 The 'integer' attribute syntax is an integer value that is in the range from  $-2^{31}$  (MIN) to  $2^{31} - 1$  (MAX).  
2639 Each individual attribute may specify the range constraint explicitly in sub-section headers if the range is  
2640 different from the full range of possible integer values. For example: job-priority (integer(1:100)) for the  
2641 "job-priority" attribute. However, the enforcement of that additional constraint is up to the IPP objects, not  
2642 the protocol.

#### 2643 **4.1.13 'rangeOfInteger'**

2644 The 'rangeOfInteger' attribute syntax is an ordered pair of integers that defines an inclusive range of integer  
2645 values. The first integer specifies the lower bound and the second specifies the upper bound. If a range



2646 constraint is specified in the header description for an attribute in this document whose attribute syntax is  
2647 'rangeOfInteger' (i.e., 'X:Y' indicating X as a minimum value and Y as a maximum value), then the  
2648 constraint applies to both integers.

#### 2649 **4.1.14 'dateTime'**

2650 The 'dateTime' attribute syntax is a standard, fixed length, 11 octet representation of the "DateAndTime"  
2651 syntax as defined in RFC 2579 [RFC2579]. RFC 2579 also identifies an 8 octet representation of a  
2652 "DateAndTime" value, but IPP objects MUST use the 11 octet representation. A user interface will provide  
2653 a mapping between protocol dateTime values and displayable user-friendly words or presentation values  
2654 and phrases which are localized to the natural language and date format of the user, including time zone.

#### 2655 **4.1.15 'resolution'**

2656 The 'resolution' attribute syntax specifies a two-dimensional resolution in the indicated units. It consists of  
2657 3 values: a cross feed direction resolution (positive integer value), a feed direction resolution (positive  
2658 integer value), and a units value. The semantics of these three components are taken from the Printer MIB  
2659 [RFC1759] suggested values. That is, the cross feed direction component resolution component is the same  
2660 as the prtMarkerAddressabilityXFeedDir object in the Printer MIB, the feed direction component resolution  
2661 component is the same as the prtMarkerAddressabilityFeedDir in the Printer MIB, and the units component  
2662 is the same as the prtMarkerAddressabilityUnit object in the Printer MIB (namely, '3' indicates dots per inch  
2663 and '4' indicates dots per centimeter). All three values MUST be present even if the first two values are the  
2664 same. Example: '300', '600', '3' indicates a 300 dpi cross-feed direction resolution, a 600 dpi feed direction  
2665 resolution, since a '3' indicates dots per inch (dpi).

#### 2666 **4.1.16 '1setOf X'**

2667 The '1setOf X' attribute syntax is 1 or more values of attribute syntax type X. This syntax type is used for  
2668 multi-valued attributes. The syntax type is called '1setOf' rather than just 'setOf' as a reminder that the set  
2669 of values MUST NOT be empty (i.e., a set of size 0). Sets are normally unordered. However each attribute  
2670 description of this type may specify that the values MUST be in a certain order for that attribute.

## 2671 **4.2 Job Template Attributes**

2672 Job Template attributes describe job processing behavior. Support for Job Template attributes by a Printer  
2673 object is OPTIONAL (see section 12.2.3 for a description of support for OPTIONAL attributes). Also,  
2674 clients OPTIONALLY supply Job Template attributes in create requests.

2675 Job Template attributes conform to the following rules. For each Job Template attribute called "xxx":

- 2676 1. If the Printer object supports "xxx" then it MUST support both a "xxx-default" attribute (unless there  
2677 is a "No" in the table below) and a "xxx-supported" attribute. If the Printer object doesn't support  
2678 "xxx", then it MUST support neither an "xxx-default" attribute nor an "xxx-supported" attribute,  
2679 and it MUST treat an attribute "xxx" supplied by a client as unsupported. An attribute "xxx" may be

2680 supported for some document formats and not supported for other document formats. For example,  
2681 it is expected that a Printer object would only support "orientation-requested" for some document  
2682 formats (such as 'text/plain' or 'text/html') but not others (such as 'application/postscript').  
2683

2684 2. "xxx" is OPTIONALLY supplied by the client in a create request. If "xxx" is supplied, the client is  
2685 indicating a desired job processing behavior for this Job. When "xxx" is not supplied, the client is  
2686 indicating that the Printer object apply its default job processing behavior at job processing time if  
2687 the document content does not contain an embedded instruction indicating an xxx-related behavior.  
2688

2689 Since an administrator MAY change the default value attribute after a Job object has been submitted  
2690 but before it has been processed, the default value used by the Printer object at job processing time  
2691 may be different that the default value in effect at job submission time.  
2692

2693 3. The "xxx-supported" attribute is a Printer object attribute that describes which job processing  
2694 behaviors are supported by that Printer object. A client can query the Printer object to find out what  
2695 xxx-related behaviors are supported by inspecting the returned values of the "xxx-supported"  
2696 attribute.  
2697

2698 Note: The "xxx" in each "xxx-supported" attribute name is singular, even though an "xxx-  
2699 supported" attribute usually has more than one value, such as "job-sheet-supported", unless the  
2700 "xxx" Job Template attribute is plural, such as "finishings" or "sides". In such cases the "xxx-  
2701 supported" attribute names are: "finishings-supported" and "sides-supported".  
2702

2703 4. The "xxx-default" default value attribute describes what will be done at job processing time when no  
2704 other job processing information is supplied by the client (either explicitly as an IPP attribute in the  
2705 create request or implicitly as an embedded instruction within the document data).  
2706

2707 If an application wishes to present an end user with a list of supported values from which to choose, the  
2708 application SHOULD query the Printer object for its supported value attributes. The application SHOULD  
2709 also query the default value attributes. If the application then limits selectable values to only those value  
2710 that are supported, the application can guarantee that the values supplied by the client in the create request  
2711 all fall within the set of supported values at the Printer. When querying the Printer, the client MAY  
2712 enumerate each attribute by name in the Get-Printer-Attributes Request, or the client MAY just name the  
2713 "job-template" group in order to get the complete set of supported attributes (both supported and default  
2714 attributes).

2715 The "finishings" attribute is an example of a Job Template attribute. It can take on a set of values such as  
2716 'staple', 'punch', and/or 'cover'. A client can query the Printer object for the "finishings-supported" attribute  
2717 and the "finishings-default" attribute. The supported attribute contains a set of supported values. The  
2718 default value attribute contains the finishing value(s) that will be used for a new Job if the client does not  
2719 supply a "finishings" attribute in the create request and the document data does not contain any  
2720 corresponding finishing instructions. If the client does supply the "finishings" attribute in the create  
2721 request, the IPP object validates the value or values to make sure that they are a subset of the supported  
2722 values identified in the Printer object's "finishings-supported" attribute. See section 3.1.7.

2723 The table below summarizes the names and relationships for all Job Template attributes. The first column  
2724 of the table (labeled "Job Attribute") shows the name and syntax for each Job Template attribute in the Job  
2725 object. These are the attributes that can optionally be supplied by the client in a create request. The last  
2726 two columns (labeled "Printer: Default Value Attribute" and "Printer: Supported Values Attribute") show  
2727 the name and syntax for each Job Template attribute in the Printer object (the default value attribute and the  
2728 supported values attribute). A "No" in the table means the Printer MUST NOT support the attribute (that is,  
2729 the attribute is simply not applicable). For brevity in the table, the 'text' and 'name' entries do not show the  
2730 maximum length for each attribute.

2731	+=====+		
2732	Job Attribute	Printer: Default Value	Printer: Supported
2733		Attribute	Values Attribute
2734	+=====+		
2735	job-priority	job-priority-default	job-priority-supported
2736	(integer 1:100)	(integer 1:100)	(integer 1:100)
2737	+-----+		
2738	job-hold-until	job-hold-until-	job-hold-until-
2739	(type3 keyword	default	supported
2740	name)	(type3 keyword	(1setOf (
2741		name)	type3 keyword   name))
2742	+-----+		
2743	job-sheets	job-sheets-default	job-sheets-supported
2744	(type3 keyword	(type3 keyword	(1setOf (
2745	name)	name)	type3 keyword   name))
2746	+-----+		
2747	multiple-document-	multiple-document-	multiple-document-
2748	handling	handling-default	handling-supported
2749	(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2750	+-----+		
2751	copies	copies-default	copies-supported
2752	(integer (1:MAX))	(integer (1:MAX))	(rangeOfInteger
2753			(1:MAX))
2754	+-----+		
2755	finishings	finishings-default	finishings-supported
2756	(1setOf type2 enum)	(1setOf type2 enum)	(1setOf type2 enum)
2757	+-----+		
2758	page-ranges	No	page-ranges-
2759	(1setOf		supported (boolean)
2760	rangeOfInteger		
2761	(1:MAX))		
2762	+-----+		
2763	sides	sides-default	sides-supported
2764	(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2765	+-----+		
2766	number-up	number-up-default	number-up-supported
2767	(integer (1:MAX))	(integer (1:MAX))	(1setOf (integer
2768			(1:MAX)
2769			rangeOfInteger
2770			(1:MAX))
2771	+-----+		
2772	orientation-	orientation-requested-	orientation-requested-
2773	requested	default	supported
2774	(type2 enum)	(type2 enum)	(1setOf type2 enum)
2775	+-----+		
2776	media	media-default	media-supported
2777	(type3 keyword	(type3 keyword	(1setOf (
2778	name)	name)	type3 keyword   name))
2779			

2780			media-ready
2781			(1setOf (
2782			type3 keyword   name))
2783	+-----+-----+-----+		
2784	printer-resolution	printer-resolution-	printer-resolution-
2785	(resolution)	default	supported
2786		(resolution)	(1setOf resolution)
2787	+-----+-----+-----+		
2788	print-quality	print-quality-default	print-quality-
2789	(type2 enum)	(type2 enum)	supported
2790			(1setOf type2 enum)
2791	+-----+-----+-----+		
2792			
2793			

#### 2794 4.2.1 job-priority (integer(1:100))

2795 This attribute specifies a priority for scheduling the Job. A higher value specifies a higher priority. The  
 2796 value 1 indicates the lowest possible priority. The value 100 indicates the highest possible priority. Among  
 2797 those jobs that are ready to print, a Printer MUST print all jobs with a priority value of n before printing  
 2798 those with a priority value of n-1 for all n.

2799 If the Printer object supports this attribute, it MUST always support the full range from 1 to 100. No  
 2800 administrative restrictions are permitted. This way an end-user can always make full use of the entire range  
 2801 with any Printer object. If privileged jobs are implemented outside IPP/1.1, they MUST have priorities  
 2802 higher than 100, rather than restricting the range available to end-users.

2803 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
 2804 object MUST use the value of the Printer object's "job-priority-default" at job submission time (unlike most  
 2805 Job Template attributes that are used if necessary at job processing time).

2806 The syntax for the "job-priority-supported" is also integer(1:100). This single integer value indicates the  
 2807 number of priority levels supported. The Printer object MUST take the value supplied by the client and  
 2808 map it to the closest integer in a sequence of n integers values that are evenly distributed over the range  
 2809 from 1 to 100 using the formula:

2810 
$$\text{roundToNearestInt}((100x+50)/n)$$

2811 where n is the value of "job-priority-supported" and x ranges from 0 through n-1.

2812 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  
 2813 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,  
 2814 and 95; if n = 100, the sequence of values is: 1, 2, 3, ... 100.

2815 If the value of the Printer object's "job-priority-supported" is 10 and the client supplies values in the range 1  
 2816 to 10, the Printer object maps them to 5, in the range 11 to 20, the Printer object maps them to 15, etc.

**2817 4.2.2 job-hold-until (type3 keyword | name (MAX))**

2818 This attribute specifies the named time period during which the Job **MUST** become a candidate for printing.

2819 Standard keyword values for named time periods are:

2820 'no-hold': immediately, if there are not other reasons to hold the job

2821 'indefinite': - the job is held indefinitely, until a client performs a Release-Job (section 3.3.6)

2822 'day-time': during the day

2823 'evening': evening

2824 'night': night

2825 'weekend': weekend

2826 'second-shift': second-shift (after close of business)

2827 'third-shift': third-shift (after midnight)

2828

2829 An administrator **MUST** associate allowable print times with a named time period (by means outside the  
2830 scope of this IPP/1.1 document). An administrator is encouraged to pick names that suggest the type of  
2831 time period. An administrator **MAY** define additional values using the 'name' or 'keyword' attribute syntax,  
2832 depending on implementation.

2833 If the value of this attribute specifies a time period that is in the future, the Printer **SHOULD** add the 'job-  
2834 hold-until-specified' value to the job's "job-state-reasons" attribute, **MUST** move the job to the 'pending-  
2835 held' state, and **MUST NOT** schedule the job for printing until the specified time-period arrives.

2836 When the specified time period arrives, the Printer **MUST** remove the 'job-hold-until-specified' value from  
2837 the job's "job-state-reason" attribute, if present. If there are no other job state reasons that keep the job in  
2838 the 'pending-held' state, the Printer **MUST** consider the job as a candidate for processing by moving the job  
2839 to the 'pending' state.

2840 If this job attribute value is the named value 'no-hold', or the specified time period has already started, the  
2841 job **MUST** be a candidate for processing immediately.

2842 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
2843 object **MUST** use the value of the Printer object's "job-hold-until-default" at job submission time (unlike  
2844 most Job Template attributes that are used if necessary at job processing time).

**2845 4.2.3 job-sheets (type3 keyword | name(MAX))**

2846 This attribute determines which job start/end sheet(s), if any, **MUST** be printed with a job.

2847 Standard keyword values are:

2848 'none': no job sheet is printed

2849 'standard': one or more site specific standard job sheets are printed, e.g. a single start sheet or both start  
2850 and end sheet is printed

2851

2852 An administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending  
2853 on implementation.

2854 The effect of this attribute on jobs with multiple documents MAY be affected by the "multiple-document-  
2855 handling" job attribute (section 4.2.4), depending on the job sheet semantics.

#### 2856 **4.2.4 multiple-document-handling (type2 keyword)**

2857 This attribute is relevant only if a job consists of two or more documents. This attribute MUST be  
2858 supported if the Printer supports multiple documents per job (see sections 3.2.4 and 3.3.1). The attribute  
2859 controls finishing operations and the placement of one or more print-stream pages into impressions and  
2860 onto media sheets. When the value of the "copies" attribute exceeds 1, it also controls the order in which  
2861 the copies that result from processing the documents are produced. For the purposes of this explanations, if  
2862 "a" represents an instance of document data, then the result of processing the data in document "a" is a  
2863 sequence of media sheets represented by "a(\*)".

2864 Standard keyword values are:

2865 'single-document': If a Job object has multiple documents, say, the document data is called a and b, then  
2866 the result of processing all the document data (a and then b) MUST be treated as a single sequence  
2867 of media sheets for finishing operations; that is, finishing would be performed on the concatenation  
2868 of the sequences a(\*),b(\*). The Printer object MUST NOT force the data in each document instance  
2869 to be formatted onto a new print-stream page, nor to start a new impression on a new media sheet. If  
2870 more than one copy is made, the ordering of the sets of media sheets resulting from processing the  
2871 document data MUST be a(\*), b(\*), a(\*), b(\*), ..., and the Printer object MUST force each copy  
2872 (a(\*),b(\*)) to start on a new media sheet.

2873 'separate-documents-uncollated-copies': If a Job object has multiple documents, say, the document data  
2874 is called a and b, then the result of processing the data in each document instance MUST be treated  
2875 as a single sequence of media sheets for finishing operations; that is, the sets a(\*) and b(\*) would  
2876 each be finished separately. The Printer object MUST force each copy of the result of processing the  
2877 data in a single document to start on a new media sheet. If more than one copy is made, the ordering  
2878 of the sets of media sheets resulting from processing the document data MUST be a(\*), a(\*), ...,  
2879 b(\*), b(\*) ... .

2880 'separate-documents-collated-copies': If a Job object has multiple documents, say, the document data is  
2881 called a and b, then the result of processing the data in each document instance MUST be treated as  
2882 a single sequence of media sheets for finishing operations; that is, the sets a(\*) and b(\*) would each  
2883 be finished separately. The Printer object MUST force each copy of the result of processing the data  
2884 in a single document to start on a new media sheet. If more than one copy is made, the ordering of  
2885 the sets of media sheets resulting from processing the document data MUST be a(\*), b(\*), a(\*), b(\*),  
2886 ... .

2887 'single-document-new-sheet': Same as 'single-document', except that the Printer object MUST ensure  
2888 that the first impression of each document instance in the job is placed on a new media sheet. This  
2889 value allows multiple documents to be stapled together with a single staple where each document  
2890 starts on a new sheet.

2891

2892 The 'single-document' value is the same as 'separate-documents-collated-copies' with respect to ordering of  
 2893 print-stream pages, but not media sheet generation, since 'single-document' will put the first page of the  
 2894 next document on the back side of a sheet if an odd number of pages have been produced so far for the job,  
 2895 while 'separate-documents-collated-copies' always forces the next document or document copy on to a new  
 2896 sheet. In addition, if the "finishings" attribute specifies 'staple', then with 'single-document', documents a  
 2897 and b are stapled together as a single document with no regard to new sheets, with 'single-document-new-  
 2898 sheet', documents a and b are stapled together as a single document, but document b starts on a new sheet,  
 2899 but with 'separate-documents-uncollated-copies' and 'separate-documents-collated-copies', documents a and  
 2900 b are stapled separately.

2901 Note: None of these values provide means to produce uncollated sheets within a document, i.e., where  
 2902 multiple copies of sheet n are produced before sheet n+1 of the same document.

2903 The relationship of this attribute and the other attributes that control document processing is described in  
 2904 section 15.3.

#### 2905 **4.2.5 copies (integer(1:MAX))**

2906 This attribute specifies the number of copies to be printed.

2907 On many devices the supported number of collated copies will be limited by the number of physical output  
 2908 bins on the device, and may be different from the number of uncollated copies which can be supported.

2909 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
 2910 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
 2911 control document processing is described in section 15.3.

#### 2912 **4.2.6 finishings (1setOf type2 enum)**

2913 This attribute identifies the finishing operations that the Printer uses for each copy of each printed  
 2914 document in the Job. For Jobs with multiple documents, the "multiple-document-handling" attribute  
 2915 determines what constitutes a "copy" for purposes of finishing.

2916 Standard enum values are:

2917	Value	Symbolic Name and Description
2918		
2919	'3'	'none': Perform no finishing
2920	'4'	'staple': Bind the document(s) with one or more staples. The exact number and placement of
2921		the staples is site-defined.
2922	'5'	'punch': This value indicates that holes are required in the finished document. The exact
2923		number and placement of the holes is site-defined The punch specification MAY be
2924		satisfied (in a site- and implementation-specific manner) either by drilling/punching,
2925		or by substituting pre-drilled media.



- 2926 '6' 'cover': This value is specified when it is desired to select a non-printed (or pre-printed)  
2927 cover for the document. This does not supplant the specification of a printed cover  
2928 (on cover stock medium) by the document itself.
- 2929 '7' 'bind': This value indicates that a binding is to be applied to the document; the type and  
2930 placement of the binding is site-defined.
- 2931
- 2932 '8' 'saddle-stitch': Bind the document(s) with one or more staples (wire stitches) along the  
2933 middle fold. The exact number and placement of the staples and the middle fold is  
2934 implementation and/or site-defined.
- 2935 '9' 'edge-stitch': Bind the document(s) with one or more staples (wire stitches) along one edge.  
2936 The exact number and placement of the staples is implementation and/or site-  
2937 defined.
- 2938 '10'-'19' reserved for future generic finishing enum values.

2939 The following values are more specific; they indicate a corner or an edge as if the document were a portrait  
2940 document (see below):

- 2941 '20' 'staple-top-left': Bind the document(s) with one or more staples in the top left corner.
- 2942 '21' 'staple-bottom-left': Bind the document(s) with one or more staples in the bottom left  
2943 corner.
- 2944 '22' 'staple-top-right': Bind the document(s) with one or more staples in the top right corner.
- 2945 '23' 'staple-bottom-right': Bind the document(s) with one or more staples in the bottom right  
2946 corner.
- 2947 '24' 'edge-stitch-left': Bind the document(s) with one or more staples (wire stitches) along the  
2948 left edge. The exact number and placement of the staples is implementation and/or  
2949 site-defined.
- 2950 '25' 'edge-stitch-top': Bind the document(s) with one or more staples (wire stitches) along the  
2951 top edge. The exact number and placement of the staples is implementation and/or  
2952 site-defined.
- 2953 '26' 'edge-stitch-right': Bind the document(s) with one or more staples (wire stitches) along the  
2954 right edge. The exact number and placement of the staples is implementation and/or  
2955 site-defined.
- 2956 '27' 'edge-stitch-bottom': Bind the document(s) with one or more staples (wire stitches) along  
2957 the bottom edge. The exact number and placement of the staples is implementation  
2958 and/or site-defined.
- 2959 '28' 'staple-dual-left': Bind the document(s) with two staples (wire stitches) along the left edge  
2960 assuming a portrait document (see above).
- 2961 '29' 'staple-dual-top': Bind the document(s) with two staples (wire stitches) along the top edge  
2962 assuming a portrait document (see above).
- 2963 '30' 'staple-dual-right': Bind the document(s) with two staples (wire stitches) along the right  
2964 edge assuming a portrait document (see above).
- 2965 '31' 'staple-dual-bottom': Bind the document(s) with two staples (wire stitches) along the bottom  
2966 edge assuming a portrait document (see above).

2967 The 'staple-xxx' values are specified with respect to the document as if the document were a portrait  
2968 document. If the document is actually a landscape or a reverse-landscape document, the client supplies the  
2969 appropriate transformed value. For example, to position a staple in the upper left hand corner of a

2970 landscape document when held for reading, the client supplies the 'staple-bottom-left' value (since  
2971 landscape is defined as a +90 degree rotation of the image with respect to the media from portrait, i.e., anti-  
2972 clockwise). On the other hand, to position a staple in the upper left hand corner of a reverse-landscape  
2973 document when held for reading, the client supplies the 'staple-top-right' value (since reverse-landscape is  
2974 defined as a -90 degree rotation of the image with respect to the media from portrait, i.e., clockwise).

2975 The angle (vertical, horizontal, angled) of each staple with respect to the document depends on the  
2976 implementation which may in turn depend on the value of the attribute.

2977 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
2978 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
2979 control document processing is described in section 15.3.

2980 If the client supplies a value of 'none' along with any other combination of values, it is the same as if only  
2981 that other combination of values had been supplied (that is the 'none' value has no effect).

#### 2982 **4.2.7 page-ranges (1setOf rangeOfInteger (1:MAX))**

2983 This attribute identifies the range(s) of print-stream pages that the Printer object uses for each copy of each  
2984 document which are to be printed. Nothing is printed for any pages identified that do not exist in the  
2985 document(s). Ranges MUST be in ascending order, for example: 1-3, 5-7, 15-19 and MUST NOT overlap,  
2986 so that a non-spooling Printer object can process the job in a single pass. If the ranges are not ascending or  
2987 are overlapping, the IPP object MUST reject the request and return the 'client-error-bad-request' status code.  
2988 The attribute is associated with print-stream pages not application-numbered pages (for example, the page  
2989 numbers found in the headers and or footers for certain word processing applications).

2990 For Jobs with multiple documents, the "multiple-document-handling" attribute determines what constitutes  
2991 a "copy" for purposes of the specified page range(s). When "multiple-document-handling" is 'single-  
2992 document', the Printer object MUST apply each supplied page range once to the concatenation of the print-  
2993 stream pages. For example, if there are 8 documents of 10 pages each, the page-range '41:60' prints the  
2994 pages in the 5th and 6th documents as a single document and none of the pages of the other documents are  
2995 printed. When "multiple-document-handling" is 'separate-documents-uncollated-copies' or 'separate-  
2996 documents-collated-copies', the Printer object MUST apply each supplied page range repeatedly to each  
2997 document copy. For the same job, the page-range '1:3, 10:10' would print the first 3 pages and the 10th  
2998 page of each of the 8 documents in the Job, as 8 separate documents.

2999 In most cases, the exact pages to be printed will be generated by a device driver and this attribute would not  
3000 be required. However, when printing an archived document which has already been formatted, the end user  
3001 may elect to print just a subset of the pages contained in the document. In this case, if page-range = n.m is  
3002 specified, the first page to be printed will be page n. All subsequent pages of the document will be printed  
3003 through and including page m.

3004 "page-ranges-supported" is a boolean value indicating whether or not the printer is capable of supporting  
3005 the printing of page ranges. This capability may differ from one PDL to another. There is no "page-ranges-  
3006 default" attribute. If the "page-ranges" attribute is not supplied by the client, all pages of the document will  
3007 be printed.

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

#### 3011 **4.2.8 sides (type2 keyword)**

3012 This attribute specifies how print-stream pages are to be imposed upon the sides of an instance of a selected  
3013 medium, i.e., an impression.

3014 The standard keyword values are:

3015 'one-sided': imposes each consecutive print-stream page upon the same side of consecutive media  
3016 sheets.

3017 'two-sided-long-edge': imposes each consecutive pair of print-stream pages upon front and back sides of  
3018 consecutive media sheets, such that the orientation of each pair of print-stream pages on the medium  
3019 would be correct for the reader as if for binding on the long edge. This imposition is sometimes  
3020 called 'duplex' or 'head-to-head'.

3021 'two-sided-short-edge': imposes each consecutive pair of print-stream pages upon front and back sides  
3022 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the  
3023 medium would be correct for the reader as if for binding on the short edge. This imposition is  
3024 sometimes called 'tumble' or 'head-to-toe'.

3025

3026 'two-sided-long-edge', 'two-sided-short-edge', 'tumble', and 'duplex' all work the same for portrait or  
3027 landscape. However 'head-to-toe' is 'tumble' in portrait but 'duplex' in landscape. 'head-to-head' also  
3028 switches between 'duplex' and 'tumble' when using portrait and landscape modes.

3029 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
3030 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
3031 control document processing is described in section 15.3.

#### 3032 **4.2.9 number-up (integer(1:MAX))**

3033 This attribute specifies the number of print-stream pages to impose upon a single side of an instance of a  
3034 selected medium. For example, if the value is:

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

3042

3043 This attribute primarily controls the translation, scaling and rotation of print-stream pages.

3044

3045 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
 3046 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
 3047 control document processing is described in section 15.3.

#### 3048 **4.2.10 orientation-requested (type2 enum)**

3049 This attribute indicates the desired orientation for printed print-stream pages; it does not describe the  
 3050 orientation of the client-supplied print-stream pages.

3051 For some document formats (such as 'application/postscript'), the desired orientation of the print-stream  
 3052 pages is specified within the document data. This information is generated by a device driver prior to the  
 3053 submission of the print job. Other document formats (such as 'text/plain') do not include the notion of  
 3054 desired orientation within the document data. In the latter case it is possible for the Printer object to bind  
 3055 the desired orientation to the document data after it has been submitted. It is expected that a Printer object  
 3056 would only support "orientations-requested" for some document formats (e.g., 'text/plain' or 'text/html') but  
 3057 not others (e.g., 'application/postscript'). This is no different than any other Job Template attribute since  
 3058 section 4.2, item 1, points out that a Printer object may support or not support any Job Template attribute  
 3059 based on the document format supplied by the client. However, a special mention is made here since it is  
 3060 very likely that a Printer object will support "orientation-requested" for only a subset of the supported  
 3061 document formats.

3062 Standard enum values are:

3063	Value	Symbolic Name and Description
3064		
3065	'3'	'portrait': The content will be imaged across the short edge of the medium.
3066	'4'	'landscape': The content will be imaged across the long edge of the medium. Landscape is
3067		defined to be a rotation of the print-stream page to be imaged by +90 degrees with
3068		respect to the medium (i.e. anti-clockwise) from the portrait orientation. Note: The
3069		+90 direction was chosen because simple finishing on the long edge is the same edge
3070		whether portrait or landscape
3071	'5'	'reverse-landscape': The content will be imaged across the long edge of the medium.
3072		Reverse-landscape is defined to be a rotation of the print-stream page to be imaged
3073		by -90 degrees with respect to the medium (i.e. clockwise) from the portrait
3074		orientation. Note: The 'reverse-landscape' value was added because some
3075		applications rotate landscape -90 degrees from portrait, rather than +90 degrees.
3076	'6'	'reverse-portrait': The content will be imaged across the short edge of the medium. Reverse-
3077		portrait is defined to be a rotation of the print-stream page to be imaged by 180
3078		degrees with respect to the medium from the portrait orientation. Note: The 'reverse-
3079		portrait' value was added for use with the "finishings" attribute in cases where the
3080		opposite edge is desired for finishing a portrait document on simple finishing devices
3081		that have only one finishing position. Thus a 'text/plain' portrait document can be
3082		stapled "on the right" by a simple finishing device as is common use with some
3083		middle eastern languages such as Hebrew.
3084		

3085 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-document-  
3086 handling" job attribute (section 4.2.4) and the relationship of this attribute and the other attributes that  
3087 control document processing is described in section 15.3.

#### 3088 **4.2.11 media (type3 keyword | name(MAX))**

3089 This attribute identifies the medium that the Printer uses for all impressions of the Job.

3090 The values for "media" include medium-names, medium-sizes, input-trays and electronic forms so that one  
3091 attribute specifies the media. If a Printer object supports a medium name as a value of this attribute, such a  
3092 medium name implicitly selects an input-tray that contains the specified medium. If a Printer object  
3093 supports a medium size as a value of this attribute, such a medium size implicitly selects a medium name  
3094 that in turn implicitly selects an input-tray that contains the medium with the specified size. If a Printer  
3095 object supports an input-tray as the value of this attribute, such an input-tray implicitly selects the medium  
3096 that is in that input-tray at the time the job prints. This case includes manual-feed input-trays. If a Printer  
3097 object supports an electronic form as the value of this attribute, such an electronic form implicitly selects a  
3098 medium-name that in turn implicitly selects an input-tray that contains the medium specified by the  
3099 electronic form. The electronic form also implicitly selects an image that the Printer MUST merge with the  
3100 document data as its prints each page.

3101 Standard keyword values are taken from ISO DPA [ISO10175], the Printer MIB [RFC1759], and ASME-  
3102 Y14.1M [ASME-Y14.1M] and are listed in section 14. An administrator MAY define additional values  
3103 using the 'name' or 'keyword' attribute syntax, depending on implementation.

3104 There is also an additional Printer attribute named "media-ready" which differs from "media-supported" in  
3105 that legal values only include the subset of "media-supported" values that are physically loaded and ready  
3106 for printing with no operator intervention required. If an IPP object supports "media-supported", it NEED  
3107 NOT support "media-ready".

3108 The relationship of this attribute and the other attributes that control document processing is described in  
3109 section 15.3.

#### 3110 **4.2.12 printer-resolution (resolution)**

3111 This attribute identifies the resolution that Printer uses for the Job.

#### 3112 **4.2.13 print-quality (type2 enum)**

3113 This attribute specifies the print quality that the Printer uses for the Job.

3114 The standard enum values are:

3115	Value	Symbolic Name and Description
3116		
3117	'3'	'draft': lowest quality available on the printer
3118	'4'	'normal': normal or intermediate quality on the printer

3119 '5' 'high': highest quality available on the printer  
3120

### 3121 **4.3 Job Description Attributes**

3122 The attributes in this section form the attribute group called "job-description". The following table  
3123 summarizes these attributes. The third column indicates whether the attribute is a REQUIRED attribute  
3124 that MUST be supported by Printer objects. If it is not indicated as REQUIRED, then it is OPTIONAL.  
3125 The maximum size in octets for 'text' and 'name' attributes is indicated in parentheses.

	Attribute	Syntax	REQUIRED?
3126			
3127			
3128			
3129	job-uri	uri	REQUIRED
3130			
3131	job-id	integer(1:MAX)	REQUIRED
3132			
3133	job-printer-uri	uri	REQUIRED
3134			
3135	job-more-info	uri	
3136			
3137	job-name	name (MAX)	REQUIRED
3138			
3139	job-originating-user-name	name (MAX)	REQUIRED
3140			
3141	job-state	type1 enum	REQUIRED
3142			
3143	job-state-reasons	1setOf type2 keyword	REQUIRED
3144			
3145	job-state-message	text (MAX)	
3146			
3147	job-detailed-status- messages	1setOf text (MAX)	
3148			
3149			
3150	job-document-access-errors	1setOf text (MAX)	
3151			
3152	number-of-documents	integer (0:MAX)	
3153			
3154	output-device-assigned	name (127)	
3155			
3156	time-at-creation	integer (MIN:MAX)	REQUIRED
3157			
3158	time-at-processing	integer (MIN:MAX)	REQUIRED
3159			
3160	time-at-completed	integer (MIN:MAX)	REQUIRED
3161			
3162	job-printer-up-time	integer (1:MAX)	REQUIRED
3163			
3164	date-time-at-creation	dateTime	
3165			
3166	date-time-at-processing	dateTime	
3167			
3168	date-time-at-completed	dateTime	
3169			
3170	number-of-intervening-jobs	integer (0:MAX)	
3171			
3172	job-message-from-operator	text (127)	
3173			
3174	job-k-octets	integer (0:MAX)	

3175	+	-----	+	-----	+	-----	+
3176		job-impressions		integer (0:MAX)			
3177	+	-----	+	-----	+	-----	+
3178		job-media-sheets		integer (0:MAX)			
3179	+	-----	+	-----	+	-----	+
3180		job-k-octets-processed		integer (0:MAX)			
3181	+	-----	+	-----	+	-----	+
3182		job-impressions-completed		integer (0:MAX)			
3183	+	-----	+	-----	+	-----	+
3184		job-media-sheets-completed		integer (0:MAX)			
3185	+	-----	+	-----	+	-----	+
3186		attributes-charset		charset		REQUIRED	
3187	+	-----	+	-----	+	-----	+
3188		attributes-natural-language		naturalLanguage		REQUIRED	
3189	+	-----	+	-----	+	-----	+
3190							
3191							

#### 3192 4.3.1 job-uri (uri)

3193 This REQUIRED attribute contains the URI for the job. The Printer object, on receipt of a new job,  
 3194 generates a URI which identifies the new Job. The Printer object returns the value of the "job-uri" attribute  
 3195 as part of the response to a create request. The precise format of a Job URI is implementation dependent.  
 3196 If the Printer object supports more than one URI and there is some relationship between the newly formed  
 3197 Job URI and the Printer object's URI, the Printer object uses the Printer URI supplied by the client in the  
 3198 create request. For example, if the create request comes in over a secure channel, the new Job URI MUST  
 3199 use the same secure channel. This can be guaranteed because the Printer object is responsible for  
 3200 generating the Job URI and the Printer object is aware of its security configuration and policy as well as the  
 3201 Printer URI used in the create request.

3202 For a description of this attribute and its relationship to "job-id" and "job-printer-uri" attribute, see the  
 3203 discussion in section 2.4 on "Object Identity".

#### 3204 4.3.2 job-id (integer(1:MAX))

3205 This REQUIRED attribute contains the ID of the job. The Printer, on receipt of a new job, generates an ID  
 3206 which identifies the new Job on that Printer. The Printer returns the value of the "job-id" attribute as part of  
 3207 the response to a create request. The 0 value is not included to allow for compatibility with SNMP index  
 3208 values which also cannot be 0.

3209 For a description of this attribute and its relationship to "job-uri" and "job-printer-uri" attribute, see the  
 3210 discussion in section 2.4 on "Object Identity".

#### 3211 4.3.3 job-printer-uri (uri)

3212 This REQUIRED attribute identifies the Printer object that created this Job object. When a Printer object  
 3213 creates a Job object, it populates this attribute with the Printer object URI that was used in the create



3214 request. This attribute permits a client to identify the Printer object that created this Job object when only  
3215 the Job object's URI is available to the client. The client queries the creating Printer object to determine  
3216 which languages, charsets, operations, are supported for this Job.

3217 For a description of this attribute and its relationship to "job-uri" and "job-id" attribute, see the discussion in  
3218 section 2.4 on "Object Identity".

#### 3219 **4.3.4 job-more-info (uri)**

3220 Similar to "printer-more-info", this attribute contains the URI referencing some resource with more  
3221 information about this Job object, perhaps an HTML page containing information about the Job.

#### 3222 **4.3.5 job-name (name(MAX))**

3223 This REQUIRED attribute is the name of the job. It is a name that is more user friendly than the "job-uri"  
3224 attribute value. It does not need to be unique between Jobs. The Job's "job-name" attribute is set to the  
3225 value supplied by the client in the "job-name" operation attribute in the create request (see Section 3.2.1.1).  
3226 If, however, the "job-name" operation attribute is not supplied by the client in the create request, the Printer  
3227 object, on creation of the Job, MUST generate a name. The printer SHOULD generate the value of the  
3228 Job's "job-name" attribute from the first of the following sources that produces a value: 1) the "document-  
3229 name" operation attribute of the first (or only) document, 2) the "document-URI" attribute of the first (or  
3230 only) document, or 3) any other piece of Job specific and/or Document Content information.

#### 3231 **4.3.6 job-originating-user-name (name(MAX))**

3232 This REQUIRED attribute contains the name of the end user that submitted the print job. The Printer  
3233 object sets this attribute to the most authenticated printable name that it can obtain from the authentication  
3234 service over which the IPP operation was received. Only if such is not available, does the Printer object use  
3235 the value supplied by the client in the "requesting-user-name" operation attribute of the create operation  
3236 (see Sections 4.4.2, 4.4.3, and 8).

3237 Note: The Printer object needs to keep an internal originating user id of some form, typically as a credential  
3238 of a principal, with the Job object. Since such an internal attribute is implementation-dependent and not of  
3239 interest to clients, it is not specified as a Job Description attribute. This originating user id is used for  
3240 authorization checks (if any) on all subsequent operations.

#### 3241 **4.3.7 job-state (type1 enum)**

3242 This REQUIRED attribute identifies the current state of the job. Even though the IPP protocol defines  
3243 seven values for job states (plus the out-of-band 'unknown' value - see Section 4.1), implementations only  
3244 need to support those states which are appropriate for the particular implementation. In other words, a  
3245 Printer supports only those job states implemented by the output device and available to the Printer object  
3246 implementation.

3247 Standard enum values are:

3248	Values	Symbolic Name and Description
3249		
3250	'3'	'pending': The job is a candidate to start processing, but is not yet processing.
3251		
3252	'4'	'pending-held': The job is not a candidate for processing for any number of reasons but will return to the 'pending' state as soon as the reasons are no longer present. The job's "job-state-reason" attribute MUST indicate why the job is no longer a candidate for processing.
3253		
3254		
3255		
3256		
3257	'5'	'processing': One or more of:
3258		
3259		1. the job is using, or is attempting to use, one or more purely software processes that are analyzing, creating, or interpreting a PDL, etc.,
3260		
3261		2. the job is using, or is attempting to use, one or more hardware devices that are interpreting a PDL, making marks on a medium, and/or performing finishing, such as stapling, etc.,
3262		
3263		
3264		3. the Printer object has made the job ready for printing, but the output device is not yet printing it, either because the job hasn't reached the output device or because the job is queued in the output device or some other spooler, awaiting the output device to print it.
3265		
3266		
3267		
3268		
3269		When the job is in the 'processing' state, the entire job state includes the detailed status represented in the Printer object's "printer-state", "printer-state-reasons", and "printer-state-message" attributes.
3270		
3271		
3272		Implementations MAY, though they NEED NOT, include additional values in the job's "job-state-reasons" attribute to indicate the progress of the job, such as adding the 'job-printing' value to indicate when the output device is actually making marks on paper and/or the 'processing-to-stop-point' value to indicate that the IPP object is in the process of canceling or aborting the job. Most implementations won't bother with this nuance.
3273		
3274		
3275		
3276		
3277		
3278		
3279	'6'	'processing-stopped': The job has stopped while processing for any number of reasons and will return to the 'processing' state as soon as the reasons are no longer present.
3280		
3281		
3282		The job's "job-state-reason" attribute MAY indicate why the job has stopped processing. For example, if the output device is stopped, the 'printer-stopped' value MAY be included in the job's "job-state-reasons" attribute.
3283		
3284		
3285		
3286		Note: When an output device is stopped, the device usually indicates its condition in human readable form locally at the device. A client can obtain more complete device status remotely by querying the Printer object's "printer-state", "printer-state-reasons" and "printer-state-message" attributes.
3287		
3288		
3289		

3290

3291 '7' 'canceled': The job has been canceled by a Cancel-Job operation and the Printer object has  
 3292 completed canceling the job and all job status attributes have reached their final  
 3293 values for the job. While the Printer object is canceling the job, the job remains in its  
 3294 current state, but the job's "job-state-reasons" attribute SHOULD contain the  
 3295 'processing-to-stop-point' value and one of the 'canceled-by-user', 'canceled-by-  
 3296 operator', or 'canceled-at-device' value. When the job moves to the 'canceled' state,  
 3297 the 'processing-to-stop-point' value, if present, MUST be removed, but the 'canceled-  
 3298 by-xxx', if present, MUST remain.

3299

3300 '8' 'aborted': The job has been aborted by the system, usually while the job was in the  
 3301 'processing' or 'processing-stopped' state and the Printer has completed aborting the  
 3302 job and all job status attributes have reached their final values for the job. While the  
 3303 Printer object is aborting the job, the job remains in its current state, but the job's  
 3304 "job-state-reasons" attribute SHOULD contain the 'processing-to-stop-point' and  
 3305 'aborted-by-system' values. When the job moves to the 'aborted' state, the  
 3306 'processing-to-stop-point' value, if present, MUST be removed, but the 'aborted-by-  
 3307 system' value, if present, MUST remain.

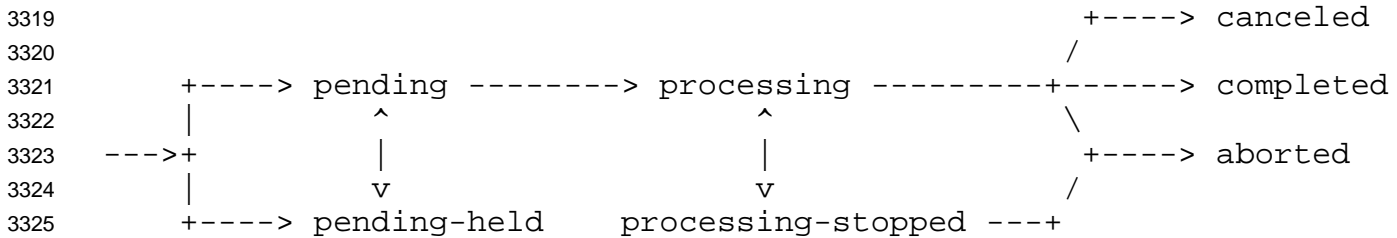
3308

3309 '9' 'completed': The job has completed successfully or with warnings or errors after processing  
 3310 and all of the job media sheets have been successfully stacked in the appropriate  
 3311 output bin(s) and all job status attributes have reached their final values for the job.  
 3312 The job's "job-state-reasons" attribute SHOULD contain one of: 'completed-  
 3313 successfully', 'completed-with-warnings', or 'completed-with-errors' values.

3314

3315 The final value for this attribute MUST be one of: 'completed', 'canceled', or 'aborted' before the Printer  
 3316 removes the job altogether. The length of time that jobs remain in the 'canceled', 'aborted', and 'completed'  
 3317 states depends on implementation. See section 4.3.7.2.

3318 The following figure shows the normal job state transitions.



3327 Normally a job progresses from left to right. Other state transitions are unlikely, but are not forbidden. Not  
 3328 shown are the transitions to the 'canceled' state from the 'pending', 'pending-held', and 'processing-stopped'  
 3329 states.

3330 Jobs reach one of the three terminal states: 'completed', 'canceled', or 'aborted', after the jobs have  
3331 completed all activity, including stacking output media, after the jobs have completed all activity, and all  
3332 job status attributes have reached their final values for the job.

#### 3333 **4.3.7.1 Forwarding Servers**

3334 As with all other IPP attributes, if the implementation cannot determine the correct value for this attribute,  
3335 it SHOULD respond with the out-of-band value 'unknown' (see section 4.1) rather than try to guess at some  
3336 possibly incorrect value and give the end user the wrong impression about the state of the Job object. For  
3337 example, if the implementation is just a gateway into some printing system from which it can normally get  
3338 status, but temporarily is unable, then the implementation should return the 'unknown' value. However, if  
3339 the implementation is a gateway to a printing system that never provides detailed status about the print job,  
3340 the implementation MAY set the IPP Job object's state to 'completed', provided that it also sets the 'queued-  
3341 in-device' value in the job's "job-state-reasons" attribute (see section 4.3.8).

#### 3342 **4.3.7.2 Partitioning of Job States**

3343 This section partitions the 7 job states into phases: Job Not Completed, Job Retention, Job History, and Job  
3344 Removal. This section also explains the 'job-restartable' value of the "job-state-reasons" Job Description  
3345 attribute for use with the Restart-Job operation.

3346 Job Not Completed: When a job is in the 'pending', 'pending-held', 'processing', or 'processing-stopped'  
3347 states, the job is not completed.

3348 Job Retention: When a job enters one of the three terminal job states: 'completed', 'canceled', or 'aborted',  
3349 the IPP Printer object MAY "retain" the job in a restartable condition for an implementation-defined time  
3350 period. This time period MAY be zero seconds and MAY depend on the terminal job state. This phase is  
3351 called Job Retention. While in the Job Retention phase, the job's document data is retained and a client  
3352 may restart the job using the Restart-Job operation. If the IPP object supports the Restart-Job operation,  
3353 then it SHOULD indicate that the job is restartable by adding the 'job-restartable' value to the job's "job-  
3354 state-reasons" attribute (see Section 4.3.8) during the Job Retention phase.

3355 Job History: After the Job Retention phase expires for a job, the Printer object deletes the document data  
3356 for the job and the job becomes part of the Job History. The Printer object MAY also delete any number of  
3357 the job attributes. Since the job is no longer restartable, the Printer object MUST remove the 'job-  
3358 restartable' value from the job's "job-state-reasons" attribute, if present.

3359 Job Removal: After the job has remained in the Job History for an implementation-defined time, such as  
3360 when the number of jobs exceeds a fixed number or after a fixed time period (which MAY be zero  
3361 seconds), the IPP Printer removes the job from the system.

3362 Using the Get-Jobs operation and supplying the 'not-completed' value for the "which-jobs" operation  
3363 attribute, a client is requesting jobs in the Job Not Completed phase. Using the Get-Jobs operation and  
3364 supplying the 'completed' value for the "which-jobs" operation attribute, a client is requesting jobs in the  
3365 Job Retention and Job History phases. Using the Get-Job-Attributes operation, a client is requesting a job

3366 in any phase except Job Removal. After Job Removal, the Get-Job-Attributes and Get-Jobs operations no  
3367 longer are capable of returning any information about a job.

#### 3368 **4.3.8 job-state-reasons (1setOf type2 keyword)**

3369 This REQUIRED attribute provides additional information about the job's current state, i.e., information  
3370 that augments the value of the job's "job-state" attribute.

3371 These values MAY be used with any job state or states for which the reason makes sense. Some of these  
3372 value definitions indicate conformance requirements; the rest are OPTIONAL. Furthermore, when  
3373 implemented, the Printer MUST return these values when the reason applies and MUST NOT return them  
3374 when the reason no longer applies whether the value of the Job's "job-state" attribute changed or not. When  
3375 the Job does not have any reasons for being in its current state, the value of the Job's "job-state-reasons"  
3376 attribute MUST be 'none'.

3377 Note: While values cannot be added to the 'job-state' attribute without impacting deployed clients that take  
3378 actions upon receiving "job-state" values, it is the intent that additional "job-state-reasons" values can be  
3379 defined and registered without impacting such deployed clients. In other words, the "job-state-reasons"  
3380 attribute is intended to be extensible.

3381 The following standard keyword values are defined. For ease of understanding, the values are presented in  
3382 the order in which the reasons are likely to occur (if implemented), starting with the 'job-incoming' value:

3383 'none': There are no reasons for the job's current state. This state reason is semantically equivalent to  
3384 "job-state-reasons" without any value and MUST be used when there is no other value, since the  
3385 1setOf attribute syntax requires at least one value.

3386 'job-incoming': Either (1) the Printer has accepted the Create-Job operation and is expecting additional  
3387 Send-Document and/or Send-URI operations, or (2) the Printer is retrieving/accepting document  
3388 data as a result of a Print-Job, Print-URI, Send-Document or Send-URI operation.

3389 'job-data-insufficient': The Create-Job operation has been accepted by the Printer, but the Printer is  
3390 expecting additional document data before it can move the job into the 'processing' state. If a Printer  
3391 starts processing before it has received all data, the Printer removes the 'job-data-insufficient'  
3392 reason, but the 'job-incoming' remains. If a Printer starts processing after it has received all data, the  
3393 Printer removes the 'job-data-insufficient' reason and the 'job-incoming' at the same time.

3394 'document-access-error': After accepting a Print-URI or Send-URI request, the Printer could not access  
3395 one or more documents passed by reference. This reason is intended to cover any file access  
3396 problem, including file does not exist and access denied because of an access control problem. The  
3397 Printer MAY also indicate the document access error using the "job-document-access-errors" Job  
3398 Description attribute (see section 4.3.11). Whether the Printer aborts the job and moves the job to  
3399 the 'aborted' job state or prints all documents that are accessible and moves the job to the 'completed'  
3400 job state and adds the 'completed-with-errors' value in the job's "job-state-reasons" attribute depends  
3401 on implementation and/or site policy. This value SHOULD be supported if the Print-URI or Send-  
3402 URI operations are supported.

3403 'submission-interrupted': The job was not completely submitted for some unforeseen reason, such as:  
3404 (1) the Printer has crashed before the job was closed by the client, (2) the Printer or the document  
3405 transfer method has crashed in some non-recoverable way before the document data was entirely

3406 transferred to the Printer, (3) the client crashed or failed to close the job before the time-out period.  
3407 See section 4.4.31.

3408 'job-outgoing': The Printer is transmitting the job to the output device.

3409 'job-hold-until-specified': The value of the job's "job-hold-until" attribute was specified with a time  
3410 period that is still in the future. The job MUST NOT be a candidate for processing until this reason  
3411 is removed and there are no other reasons to hold the job. This value SHOULD be supported if the  
3412 "job-hold-until" Job Template attribute is supported.

3413 'resources-are-not-ready': At least one of the resources needed by the job, such as media, fonts, resource  
3414 objects, etc., is not ready on any of the physical printer's for which the job is a candidate. This  
3415 condition MAY be detected when the job is accepted, or subsequently while the job is pending or  
3416 processing, depending on implementation. The job may remain in its current state or be moved to  
3417 the 'pending-held' state, depending on implementation and/or job scheduling policy.

3418 'printer-stopped-partly': The value of the Printer's "printer-state-reasons" attribute contains the value  
3419 'stopped-partly'.

3420 'printer-stopped': The value of the Printer's "printer-state" attribute is 'stopped'.

3421 'job-interpreting': Job is in the 'processing' state, but more specifically, the Printer is interpreting the  
3422 document data.

3423 'job-queued': Job is in the 'processing' state, but more specifically, the Printer has queued the document  
3424 data.

3425 'job-transforming': Job is in the 'processing' state, but more specifically, the Printer is interpreting  
3426 document data and producing another electronic representation.

3427 'job-queued-for-marker': Job is in any of the 'pending-held', 'pending', or 'processing' states, but more  
3428 specifically, the Printer has completed enough processing of the document to be able to start  
3429 marking and the job is waiting for the marker. Systems that require human intervention to release  
3430 jobs using the Release-Job operation, put the job into the 'pending-held' job state. Systems that  
3431 automatically select a job to use the marker put the job into the 'pending' job state or keep the job in  
3432 the 'processing' job state while waiting for the marker, depending on implementation. All  
3433 implementations put the job into (or back into) the 'processing' state when marking does begin.

3434 'job-printing': The output device is marking media. This value is useful for Printers which spend a great  
3435 deal of time processing (1) when no marking is happening and then want to show that marking is  
3436 now happening or (2) when the job is in the process of being canceled or aborted while the job  
3437 remains in the 'processing' state, but the marking has not yet stopped so that impression or sheet  
3438 counts are still increasing for the job.

3439 'job-canceled-by-user': The job was canceled by the owner of the job using the Cancel-Job request, i.e.,  
3440 by a user whose authenticated identity is the same as the value of the originating user that created  
3441 the Job object, or by some other authorized end-user, such as a member of the job owner's security  
3442 group. This value SHOULD be supported.

3443 'job-canceled-by-operator': The job was canceled by the operator using the Cancel-Job request, i.e., by a  
3444 user who has been authenticated as having operator privileges (whether local or remote). If the  
3445 security policy is to allow anyone to cancel anyone's job, then this value may be used when the job  
3446 is canceled by other than the owner of the job. For such a security policy, in effect, everyone is an  
3447 operator as far as canceling jobs with IPP is concerned. This value SHOULD be supported if the  
3448 implementation permits canceling by other than the owner of the job.

3449 'job-canceled-at-device': The job was canceled by an unidentified local user, i.e., a user at a console at  
3450 the device. This value SHOULD be supported if the implementation supports canceling jobs at the  
3451 console.

3452 'aborted-by-system': The job (1) is in the process of being aborted, (2) has been aborted by the system  
3453 and placed in the 'aborted' state, or (3) has been aborted by the system and placed in the 'pending-  
3454 held' state, so that a user or operator can manually try the job again. This value SHOULD be  
3455 supported.

3456 'unsupported-compression': The job was aborted by the system because the Printer determined while  
3457 attempting to decompress the document-data's that the compression is actually not among those  
3458 supported by the Printer. This value MUST be supported, since "compressions is a REQUIRED  
3459 operation attribute.

3460 'compression-error': The job was aborted by the system because the Printer encountered an error in the  
3461 document-data while decompressing it. If the Printer posts this reason, the document-data has  
3462 already passed any tests that would have led to the 'unsupported-compression' job-state-reason.

3463 'unsupported-document-format': The job was aborted by the system because the document-data's  
3464 document-format is not among those supported by the Printer. If the client specifies the document-  
3465 format as 'application/octet-stream', the printer MAY abort the job and post this reason even though  
3466 the format is a member of the "document-format-supported" printer attribute, but not among the  
3467 auto-sensed document-formats. This value MUST be supported, since "document-format" is a  
3468 REQUIRED operation attribute.

3469 'document-format-error': The job was aborted by the system because the Printer encountered an error in  
3470 the document-data while processing it. If the Printer posts this reason, the document-data has  
3471 already passed any tests that would have led to the 'unsupported-document-format' job-state-reason.

3472 'processing-to-stop-point': The requester has issued a Cancel-Job operation or the Printer object has  
3473 aborted the job, but is still performing some actions on the job until a specified stop point occurs or  
3474 job termination/cleanup is completed.

3475 If the implementation requires some measurable time to cancel the job in the 'processing' or  
3476 'processing-stopped' job states, the IPP object MUST use this value to indicate that the Printer object  
3477 is still performing some actions on the job while the job remains in the 'processing' or 'processing-  
3478 stopped' state. After all the job's job description attributes have stopped incrementing, the Printer  
3479 object moves the job from the 'processing' state to the 'canceled' or 'aborted' job states.

3480 'service-off-line': The Printer is off-line and accepting no jobs. All 'pending' jobs are put into the  
3481 'pending-held' state. This situation could be true if the service's or document transform's input is  
3482 impaired or broken.

3483 'job-completed-successfully': The job completed successfully. This value SHOULD be supported.

3484 'job-completed-with-warnings': The job completed with warnings. This value SHOULD be supported  
3485 if the implementation detects warnings.

3486 'job-completed-with-errors': The job completed with errors (and possibly warnings too). This value  
3487 SHOULD be supported if the implementation detects errors.

3488 'job-restartable' - This job is retained (see section 4.3.7.2) and is currently able to be restarted using the  
3489 Restart-Job operation (see section 3.3.7). If 'job-restartable' is a value of the job's 'job-state-reasons'  
3490 attribute, then the IPP object MUST accept a Restart-Job operation for that job. This value  
3491 SHOULD be supported if the Restart-Job operation is supported.

3492 'queued-in-device': The job has been forwarded to a device or print system that is unable to send back  
3493 status. The Printer sets the job's "job-state" attribute to 'completed' and adds the 'queued-in-device'  
3494 value to the job's "job-state-reasons" attribute to indicate that the Printer has no additional  
3495 information about the job and never will have any better information. See section 4.3.7.1.

#### 3496 **4.3.9 job-state-message (text(MAX))**

3497 This attribute specifies information about the "job-state" and "job-state-reasons" attributes in human  
3498 readable text. If the Printer object supports this attribute, the Printer object MUST be able to generate this  
3499 message in any of the natural languages identified by the Printer's "generated-natural-language-supported"  
3500 attribute (see the "attributes-natural-language" operation attribute specified in Section 3.1.4.1).

3501 The value SHOULD NOT contain additional information not contained in the values of the "job-state" and  
3502 "job-states-reasons" attributes, such as interpreter error information. Otherwise, application programs  
3503 might attempt to parse the (localized text). For such additional information such as interpreter errors for  
3504 application program consumption or specific document access errors, new attributes with keyword values,  
3505 needs to be developed and registered.

#### 3506 **4.3.10 job-detailed-status-messages (1setOf text(MAX))**

3507 This attribute specifies additional detailed and technical information about the job. The Printer NEED  
3508 NOT localize the message(s), since they are intended for use by the system administrator or other  
3509 experienced technical persons. Localization might obscure the technical meaning of such messages.  
3510 Clients MUST NOT attempt to parse the value of this attribute. See "job-document-access-errors" (section  
3511 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 canceled 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 canceled 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-   1setOf type2 keyword   REQUIRED
3680	supported
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-   uri
3695	manufacturer
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	multiple-document-jobs-   boolean
3708	supported
3709	+-----+-----+-----+
3710	charset-configured   charset   REQUIRED
3711	+-----+-----+-----+
3712	charset-supported   1setOf charset   REQUIRED
3713	+-----+-----+-----+
3714	natural-language-configured   naturalLanguage   REQUIRED
3715	+-----+-----+-----+
3716	generated-natural-language-   1setOf naturalLanguage   REQUIRED
3717	supported
3718	+-----+-----+-----+
3719	document-format-default   mimeType   REQUIRED
3720	+-----+-----+-----+

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

#### 3756 4.4.1 printer-uri-supported (1setOf uri)

3757 This REQUIRED Printer attribute contains at least one URI for the Printer object. It OPTIONALLY  
3758 contains more than one URI for the Printer object. An administrator determines a Printer object's URI(s)  
3759 and configures this attribute to contain those URIs by some means outside the scope of this IPP/1.1  
3760 document. The precise format of this URI is implementation dependent and depends on the protocol. See  
3761 the next two sections for a description of the "uri-security-supported" and "uri-authentication-supported"  
3762 attributes, both of which are the REQUIRED companion attributes to this "printer-uri-supported" attribute.  
3763 See section 2.4 on Printer object identity and section 8.2 on security and URIs for more information.

#### 3764 4.4.2 uri-authentication-supported (1setOf type2 keyword)

3765 This REQUIRED Printer attribute MUST have the same cardinality (contain the same number of values) as  
3766 the "printer-uri-supported" attribute. This attribute identifies the Client Authentication mechanism  
3767 associated with each URI listed in the "printer-uri-supported" attribute. The Printer object uses the specified  
3768 mechanism to identify the authenticated user (see section 8.3) . The "i th" value in "uri-authentication-  
3769 supported" corresponds to the "i th" value in "printer-uri-supported" and it describes the authentication  
3770 mechanisms used by the Printer when accessed via that URI. See [IPP-PRO] for more details on Client  
3771 Authentication.

3772 The following standard keyword values are defined:

3773 'none': There is no authentication mechanism associated with the URI. The Printer object assumes that  
3774 the authenticated user is "anonymous".

3775 'requesting-user-name': When a client performs an operation whose target is the associated URI, the  
3776 Printer object assumes that the authenticated user is specified by the "requesting-user-name"  
3777 Operation attribute (see section 8.3). If the "requesting-user-name" attribute is absent in a request,  
3778 the Printer object assumes that the authenticated user is "anonymous".

3779 'basic': When a client performs an operation whose target is the associated URI, the Printer object  
3780 challenges the client with HTTP basic authentication [RFC2617]. The Printer object assumes that  
3781 the authenticated user is the name received via the basic authentication mechanism.

3782 'digest': When a client performs an operation whose target is the associated URI, the Printer object  
3783 challenges the client with HTTP digest authentication [RFC2617]. The Printer object assumes that  
3784 the authenticated user is the name received via the digest authentication mechanism.

3785 'certificate': When a client performs an operation whose target is the associated URI, the Printer object  
3786 expects the client to provide a certificate. The Printer object assumes that the authenticated user is  
3787 the textual name contained within the certificate.

#### 3788 4.4.3 uri-security-supported (1setOf type2 keyword)

3789 This REQUIRED Printer attribute MUST have the same cardinality (contain the same number of values) as  
3790 the "printer-uri-supported" attribute. This attribute identifies the security mechanisms used for each URI  
3791 listed in the "printer-uri-supported" attribute. The "i th" value in "uri-security-supported" corresponds to  
3792 the "i th" value in "printer-uri-supported" and it describes the security mechanisms used for accessing the  
3793 Printer object via that URI. See [IPP-PRO] for more details on security mechanisms.

3794 The following standard keyword values are defined:

3795 'none': There are no secure communication channel protocols in use for the given URI.

3796 'ssl3': SSL3 [SSL] is the secure communications channel protocol in use for the given URI.

3797 'tls': TLS [RFC2246] is the secure communications channel protocol in use for the given URI.  
3798

3799 This attribute is orthogonal to the definition of a Client Authentication mechanism. Specifically, 'none'  
3800 does not exclude Client Authentication. See section 4.4.2.



3801 Consider the following example. For a single Printer object, an administrator configures the "printer-uri-  
3802 supported", "uri-authentication-supported" and "uri-security-supported" attributes as follows:

```
3803     "printer-uri-supported": 'xxx://acme.com/open-use-printer', 'xxx://acme.com/restricted-use-printer',  
3804     'xxx://acme.com/private-printer'  
3805     "uri-authentication-supported": 'none', 'digest', 'basic'  
3806     "uri-security-supported": 'none', 'none', 'tls'  
3807
```

3808 Note: 'xxx' is not a valid scheme. See the IPP/1.1 "Transport and Encoding" document [IPP-PRO] for the  
3809 actual URI schemes to be used in object target attributes.

3810 In this case, one Printer object has three URIs.

- 3811 - For the first URI, 'xxx://acme.com/open-use-printer', the value 'none' in "uri-security-supported"  
3812 indicates that there is no secure channel protocol configured to run under HTTP. The value of 'none'  
3813 in "uri-authentication-supported" indicates that all users are 'anonymous'. There will be no  
3814 challenge and the Printer will ignore "requesting-user-name".
- 3815 - For the second URI, 'xxx://acme.com/restricted-use-printer', the value 'none' in "uri-security-  
3816 supported" indicates that there is no secure channel protocol configured to run under HTTP. The  
3817 value of 'digest' in "uri-authentication-supported" indicates that the Printer will issue a challenge and  
3818 that the Printer will use the name supplied by the digest mechanism to determine the authenticated  
3819 user (see section 8.3).
- 3820 - For the third URI, 'xxx://acme.com/private-printer', the value 'tls' in "uri-security-supported" indicates  
3821 that TLS is being used to secure the channel. The client SHOULD be prepared to use TLS framing  
3822 to negotiate an acceptable ciphersuite to use while communicating with the Printer object. In this  
3823 case, the name implies the use of a secure communications channel, but the fact is made explicit by  
3824 the presence of the 'tls' value in "uri-security-supported". The client does not need to resort to  
3825 understanding which security it must use by following naming conventions or by parsing the URI to  
3826 determine which security mechanisms are implied. The value of 'basic' in "uri-authentication-  
3827 supported" indicates that the Printer will issue a challenge and that the Printer will use the name  
3828 supplied by the digest mechanism to determine the authenticated user (see section 8.3) . Because  
3829 this challenge occurs in a tls session, the channel is secure.

3830

3831 It is expected that many IPP Printer objects will be configured to support only one channel (either  
3832 configured to use TLS access or not) and only one authentication mechanism. Such Printer objects only  
3833 have one URI listed in the "printer-uri-supported" attribute. No matter the configuration of the Printer  
3834 object (whether it has only one URI or more than one URI), a client MUST supply only one URI in the  
3835 target "printer-uri" operation attribute.

#### 3836 4.4.4 printer-name (name(127))

3837 This REQUIRED Printer attribute contains the name of the Printer object. It is a name that is more end-  
3838 user friendly than a URI. An administrator determines a printer's name and sets this attribute to that name.  
3839 This name may be the last part of the printer's URI or it may be unrelated. In non-US-English locales, a  
3840 name may contain characters that are not allowed in a URI.

**3841 4.4.5 printer-location (text(127))**

3842 This Printer attribute identifies the location of the device. This could include things like: "in Room 123A,  
3843 second floor of building XYZ".

**3844 4.4.6 printer-info (text(127))**

3845 This Printer attribute identifies the descriptive information about this Printer object. This could include  
3846 things like: "This printer can be used for printing color transparencies for HR presentations", or "Out of  
3847 courtesy for others, please print only small (1-5 page) jobs at this printer", or even "This printer is going  
3848 away on July 1, 1997, please find a new printer".

**3849 4.4.7 printer-more-info (uri)**

3850 This Printer attribute contains a URI used to obtain more information about this specific Printer object. For  
3851 example, this could be an HTTP type URI referencing an HTML page accessible to a Web Browser. The  
3852 information obtained from this URI is intended for end user consumption. Features outside the scope of IPP  
3853 can be accessed from this URI. The information is intended to be specific to this printer instance and site  
3854 specific services (e.g. job pricing, services offered, end user assistance). The device manufacturer may  
3855 initially populate this attribute.

**3856 4.4.8 printer-driver-installer (uri)**

3857 This Printer attribute contains a URI to use to locate the driver installer for this Printer object. This  
3858 attribute is intended for consumption by automata. The mechanics of print driver installation is outside the  
3859 scope of this IPP/1.1 document. The device manufacturer may initially populate this attribute.

**3860 4.4.9 printer-make-and-model (text(127))**

3861 This Printer attribute identifies the make and model of the device. The device manufacturer may initially  
3862 populate this attribute.

**3863 4.4.10 printer-more-info-manufacturer (uri)**

3864 This Printer attribute contains a URI used to obtain more information about this type of device. The  
3865 information obtained from this URI is intended for end user consumption. Features outside the scope of  
3866 IPP can be accessed from this URI (e.g., latest firmware, upgrades, print drivers, optional features available,  
3867 details on color support). The information is intended to be germane to this printer without regard to site  
3868 specific modifications or services. The device manufacturer may initially populate this attribute.

#### 3869 4.4.11 printer-state (type1 enum)

3870 This REQUIRED Printer attribute identifies the current state of the device. The "printer-state reasons"  
3871 attribute augments the "printer-state" attribute to give more detailed information about the Printer in the  
3872 given printer state.

3873 A Printer object need only update this attribute before responding to an operation which requests the  
3874 attribute; the Printer object NEED NOT update this attribute continually, since asynchronous event  
3875 notification is not part of IPP/1.1. A Printer NEED NOT implement all values if they are not applicable to  
3876 a given implementation.

3877 The following standard enum values are defined:

3878 Value	Symbolic Name and Description
------------	-------------------------------

3879

3880 '3'	'idle': Indicates that new jobs can start processing without waiting.
----------	---

3881 '4'	'processing': Indicates that jobs are processing; new jobs will wait before processing.
----------	---

3882 '5'	'stopped': Indicates that no jobs can be processed and intervention is required.
----------	--

3883 Values of "printer-state-reasons", such as 'spool-area-full' and 'stopped-partly', MAY be used to provide  
3884 further information.

#### 3885 4.4.12 printer-state-reasons (1setOf type2 keyword)

3886 This REQUIRED Printer attribute supplies additional detail about the device's state. Some of the these  
3887 value definitions indicate conformance requirements; the rest are OPTIONAL.

3888 Each keyword value MAY have a suffix to indicate its level of severity. The three levels are: report (least  
3889 severe), warning, and error (most severe).

3890 - '-report': This suffix indicates that the reason is a "report". An implementation may choose to omit  
3891 some or all reports. Some reports specify finer granularity about the printer state; others serve as a  
3892 precursor to a warning. A report MUST contain nothing that could affect the printed output.

3893 - '-warning': This suffix indicates that the reason is a "warning". An implementation may choose to omit  
3894 some or all warnings. Warnings serve as a precursor to an error. A warning MUST contain nothing  
3895 that prevents a job from completing, though in some cases the output may be of lower quality.

3896 - '-error': This suffix indicates that the reason is an "error". An implementation MUST include all  
3897 errors. If this attribute contains one or more errors, printer MUST be in the stopped state.  
3898

3899 If the implementation does not add any one of the three suffixes, all parties MUST assume that the reason is  
3900 an "error".

3901 If a Printer object controls more than one output device, each value of this attribute MAY apply to one or  
3902 more of the output devices. An error on one output device that does not stop the Printer object as a whole  
3903 MAY appear as a warning in the Printer's "printer-state-reasons attribute". If the "printer-state" for such a  
3904 Printer has a value of 'stopped', then there MUST be an error reason among the values in the "printer-state-  
3905 reasons" attribute.

3906 The following standard keyword values are defined:

3907 'other': The device has detected an error other than one listed in this document.

3908 'none': There are not reasons. This state reason is semantically equivalent to "printer-state-reasons"  
3909 without any value and MUST be used, since the lsetOf attribute syntax requires at least one value.

3910 'media-needed': A tray has run out of media.

3911 'media-jam': The device has a media jam.

3912 'moving-to-paused': Someone has paused the Printer object using the Pause-Printer operation (see  
3913 section 3.2.7) or other means, but the device(s) are taking an appreciable time to stop. Later, when  
3914 all output has stopped, the "printer-state" becomes 'stopped', and the 'paused' value replaces the  
3915 'moving-to-paused' value in the "printer-state-reasons" attribute. This value MUST be supported, if  
3916 the Pause-Printer operation is supported and the implementation takes significant time to pause a  
3917 device in certain circumstances.

3918 'paused': Someone has paused the Printer object using the Pause-Printer operation (see section 3.2.7) or  
3919 other means and the Printer object's "printer-state" is 'stopped'. In this state, a Printer MUST NOT  
3920 produce printed output, but it MUST perform other operations requested by a client. If a Printer had  
3921 been printing a job when the Printer was paused, the Printer MUST resume printing that job when  
3922 the Printer is no longer paused and leave no evidence in the printed output of such a pause. This  
3923 value MUST be supported, if the Pause-Printer operation is supported.

3924 'shutdown': Someone has removed a Printer object from service, and the device may be powered down  
3925 or physically removed. In this state, a Printer object MUST NOT produce printed output, and unless  
3926 the Printer object is realized by a print server that is still active, the Printer object MUST perform no  
3927 other operations requested by a client, including returning this value. If a Printer object had been  
3928 printing a job when it was shutdown, the Printer NEED NOT resume printing that job when the  
3929 Printer is no longer shutdown. If the Printer resumes printing such a job, it may leave evidence in  
3930 the printed output of such a shutdown, e.g. the part printed before the shutdown may be printed a  
3931 second time after the shutdown.

3932 'connecting-to-device': The Printer object has scheduled a job on the output device and is in the process  
3933 of connecting to a shared network output device (and might not be able to actually start printing the  
3934 job for an arbitrarily long time depending on the usage of the output device by other servers on the  
3935 network).

3936 'timed-out': The server was able to connect to the output device (or is always connected), but was unable  
3937 to get a response from the output device.

3938 'stopping': The Printer object is in the process of stopping the device and will be stopped in a while.  
3939 When the device is stopped, the Printer object will change the Printer object's state to 'stopped'. The  
3940 'stopping-warning' reason is never an error, even for a Printer with a single output device. When an  
3941 output-device ceases accepting jobs, the Printer will have this reason while the output device  
3942 completes printing.

3943 'stopped-partly': When a Printer object controls more than one output device, this reason indicates that  
3944 one or more output devices are stopped. If the reason is a report, fewer than half of the output  
3945 devices are stopped. If the reason is a warning, fewer than all of the output devices are stopped.

3946 'toner-low': The device is low on toner.

3947 'toner-empty': The device is out of toner.

3948 'spool-area-full': The limit of persistent storage allocated for spooling has been reached. The Printer is  
3949 temporarily unable to accept more jobs. The Printer will remove this value when it is able to accept

3950 more jobs. This value SHOULD be used by a non-spooling Printer that only accepts one or a small  
3951 number jobs at a time or a spooling Printer that has filled the spool space.  
3952 'cover-open': One or more covers on the device are open.  
3953 'interlock-open': One or more interlock devices on the printer are unlocked.  
3954 'door-open': One or more doors on the device are open.  
3955 'input-tray-missing': One or more input trays are not in the device.  
3956 'media-low': At least one input tray is low on media.  
3957 'media-empty': At least one input tray is empty.  
3958 'output-tray-missing': One or more output trays are not in the device  
3959 'output-area-almost-full': One or more output area is almost full (e.g. tray, stacker, collator).  
3960 'output-area-full': One or more output area is full. (e.g. tray, stacker, collator)  
3961 'marker-supply-low': The device is low on at least one marker supply. (e.g. toner, ink, ribbon)  
3962 'marker-supply-empty': The device is out of at least one marker supply. (e.g. toner, ink, ribbon)  
3963 'marker-waste-almost-full': The device marker supply waste receptacle is almost full.  
3964 'marker-waste-full': The device marker supply waste receptacle is full.  
3965 'fuser-over-temp': The fuser temperature is above normal.  
3966 'fuser-under-temp': The fuser temperature is below normal.  
3967 'opc-near-eol': The optical photo conductor is near end of life.  
3968 'opc-life-over': The optical photo conductor is no longer functioning.  
3969 'developer-low': The device is low on developer.  
3970 'developer-empty': The device is out of developer.  
3971 'interpreter-resource-unavailable': An interpreter resource is unavailable (i.e. font, form)  
3972

#### 3973 **4.4.13 printer-state-message (text(MAX))**

3974 This Printer attribute specifies information about the "printer-state" and "printer-state-reasons" attributes in  
3975 human readable text. If the Printer object supports this attribute, the Printer object MUST be able to  
3976 generate this message in any of the natural languages identified by the Printer's "generated-natural-  
3977 language-supported" attribute (see the "attributes-natural-language" operation attribute specified in Section  
3978 3.1.4.1).

#### 3979 **4.4.14 ipp-versions-supported (1setOf type2 keyword)**

3980 This REQUIRED attribute identifies the IPP protocol version(s) that this Printer supports, including major  
3981 and minor versions, i.e., the version numbers for which this Printer implementation meets the conformance  
3982 requirements. For version number validation, the Printer matches the (two-octet binary) "version-number"  
3983 parameter supplied by the client in each request (see sections 3.1.1 and 3.1.8) with the (US-ASCII) keyword  
3984 values of this attribute.

3985 The following standard keyword values are defined:

3986 '1.0': Meets the conformance requirement of IPP version 1.0 as specified in RFC 2566 [RFC2566] and  
3987 RFC 2565 [RFC2565] including any extensions registered according to Section 6 and any extension  
3988 defined in this version or any future version of the IPP "Model and Semantics" document or the IPP

3989 "Encoding and Transport" document following the rules, if any, when the "version-number"  
 3990 parameter is '1.0'.  
 3991 '1.1': Meets the conformance requirement of IPP version 1.1 as specified in this document and [IPP-  
 3992 PRO] including any extensions registered according to Section 6 and any extension defined in any  
 3993 future versions of the IPP "Model and Semantics" document or the IPP Encoding and Transport  
 3994 document following the rules, if any, when the "version-number" parameter is '1.1'.

#### 3995 4.4.15 operations-supported (1setOf type2 enum)

3996 This REQUIRED Printer attribute specifies the set of supported operations for this Printer object and  
 3997 contained Job objects.

3998 This attribute is encoded as any other enum attribute syntax according to [IPP-PRO] as 32-bits. However,  
 3999 all 32-bit enum values for this attribute MUST NOT exceed 0x00008FFF, since these same values are also  
 4000 passed in two octets in the "operation-id" parameter (see section 3.1.1) in each Protocol request with the  
 4001 two high order octets omitted in order to indicate the operation being performed [IPP-PRO].

4002 The following standard enum and "operation-id" (see section 3.1.2) values are defined:

4003	Value	Operation Name
4004	-----	-----
4005		
4006	0x0000	reserved, not used
4007	0x0001	reserved, not used
4008	0x0002	Print-Job
4009	0x0003	Print-URI
4010	0x0004	Validate-Job
4011	0x0005	Create-Job
4012	0x0006	Send-Document
4013	0x0007	Send-URI
4014	0x0008	Cancel-Job
4015	0x0009	Get-Job-Attributes
4016	0x000A	Get-Jobs
4017	0x000B	Get-Printer-Attributes
4018	0x000C	Hold-Job
4019	0x000D	Release-Job
4020	0x000E	Restart-Job
4021	0x000F	reserved for a future operation
4022	0x0010	Pause-Printer
4023	0x0011	Resume-Printer
4024	0x0012	Purge-Jobs
4025	0x0013-0x3FFF	reserved for future IETF standards track operations (see section 6.4)
4026	0x4000-0x8FFF	reserved for vendor extensions (see section 6.4)
4027		

**4028 4.4.16 multiple-document-jobs-supported (boolean)**

4029 This Printer attribute indicates whether or not the Printer supports more than one document per job, i.e.,  
4030 more than one Send-Document or Send-Data operation with document data. If the Printer supports the  
4031 Create-Job and Send-Document operations (see section 3.2.4 and 3.3.1), it MUST support this attribute.

**4032 4.4.17 charset-configured (charset)**

4033 This REQUIRED Printer attribute identifies the charset that the Printer object has been configured to  
4034 represent 'text' and 'name' Printer attributes that are set by the operator, system administrator, or  
4035 manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info" (text), and "printer-  
4036 make-and-model" (text). Therefore, the value of the Printer object's "charset-configured" attribute MUST  
4037 also be among the values of the Printer object's "charset-supported" attribute.

**4038 4.4.18 charset-supported (1setOf charset)**

4039 This REQUIRED Printer attribute identifies the set of charsets that the Printer and contained Job objects  
4040 support in attributes with attribute syntax 'text' and 'name'. At least the value 'utf-8' MUST be present, since  
4041 IPP objects MUST support the UTF-8 [RFC2279] charset. If a Printer object supports a charset, it means  
4042 that for all attributes of syntax 'text' and 'name' the IPP object MUST (1) accept the charset in requests and  
4043 return the charset in responses as needed.

4044 If more charsets than UTF-8 are supported, the IPP object MUST perform charset conversion between the  
4045 charsets as described in Section 3.1.4.2.

**4046 4.4.19 natural-language-configured (naturalLanguage)**

4047 This REQUIRED Printer attribute identifies the natural language that the Printer object has been configured  
4048 to represent 'text' and 'name' Printer attributes that are set by the operator, system administrator, or  
4049 manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info" (text), and "printer-  
4050 make-and-model" (text). When returning these Printer attributes, the Printer object MAY return them in the  
4051 configured natural language specified by this attribute, instead of the natural language requested by the  
4052 client in the "attributes-natural-language" operation attribute. See Section 3.1.4.1 for the specification of  
4053 the OPTIONAL multiple natural language support. Therefore, the value of the Printer object's "natural-  
4054 language-configured" attribute MUST also be among the values of the Printer object's "generated-natural-  
4055 language-supported" attribute.

**4056 4.4.20 generated-natural-language-supported (1setOf naturalLanguage)**

4057 This REQUIRED Printer attribute identifies the natural language(s) that the Printer object and contained  
4058 Job objects support in attributes with attribute syntax 'text' and 'name'. The natural language(s) supported  
4059 depends on implementation and/or configuration. Unlike charsets, IPP objects MUST accept requests with  
4060 any natural language or any Natural Language Override whether the natural language is supported or not.

4061 If a Printer object supports a natural language, it means that for any of the attributes for which the Printer or  
4062 Job object generates messages, i.e., for the "job-state-message" and "printer-state-message" attributes and  
4063 Operation Messages (see Section 3.1.5) in operation responses, the Printer and Job objects MUST be able  
4064 to generate messages in any of the Printer's supported natural languages. See section 3.1.4 for the definition  
4065 of 'text' and 'name' attributes in operation requests and responses.

4066 Note: A Printer object that supports multiple natural languages, often has separate catalogs of messages,  
4067 one for each natural language supported.

#### 4068 **4.4.21 document-format-default (mimeMediaType)**

4069 This REQUIRED Printer attribute identifies the document format that the Printer object has been  
4070 configured to assume if the client does not supply a "document-format" operation attribute in any of the  
4071 operation requests that supply document data. The standard values for this attribute are Internet Media  
4072 types (sometimes called MIME types). For further details see the description of the 'mimeMediaType'  
4073 attribute syntax in Section 4.1.9.

#### 4074 **4.4.22 document-format-supported (1setOf mimeMediaType)**

4075 This REQUIRED Printer attribute identifies the set of document formats that the Printer object and  
4076 contained Job objects can support. For further details see the description of the 'mimeMediaType' attribute  
4077 syntax in Section 4.1.9.

#### 4078 **4.4.23 printer-is-accepting-jobs (boolean)**

4079 This REQUIRED Printer attribute indicates whether the printer is currently able to accept jobs, i.e., is  
4080 accepting Print-Job, Print-URI, and Create-Job requests. If the value is 'true', the printer is accepting jobs.  
4081 If the value is 'false', the Printer object is currently rejecting any jobs submitted to it. In this case, the  
4082 Printer object returns the 'server-error-not-accepting-jobs' status code.

4083 This value is independent of the "printer-state" and "printer-state-reasons" attributes because its value does  
4084 not affect the current job; rather it affects future jobs. This attribute, when 'false', causes the Printer to  
4085 reject jobs even when the "printer-state" is 'idle' or, when 'true', causes the Printer object to accept jobs  
4086 even when the "printer-state" is 'stopped'.

#### 4087 **4.4.24 queued-job-count (integer(0:MAX))**

4088 This REQUIRED Printer attribute contains a count of the number of jobs that are either 'pending',  
4089 'processing', 'pending-held', or 'processing-stopped' and is set by the Printer object.

#### 4090 **4.4.25 printer-message-from-operator (text(127))**

4091 This Printer attribute provides a message from an operator, system administrator or "intelligent" process to  
4092 indicate to the end user information or status of the printer, such as why it is unavailable or when it is  
4093 expected to be available.



4094 **4.4.26 color-supported (boolean)**

4095 This Printer attribute identifies whether the device is capable of any type of color printing at all, including  
4096 highlight color. All document instructions having to do with color are embedded within the document PDL  
4097 (none are external IPP attributes in IPP/1.1).

4098 Note: end-users are able to determine the nature and details of the color support by querying the "printer-  
4099 more-info-manufacturer" Printer attribute.

4100 **4.4.27 reference-uri-schemes-supported (1setOf uriScheme)**

4101 This Printer attribute specifies which URI schemes are supported for use in the "document-uri" operation  
4102 attribute of the Print-URI or Send-URI operation. If a Printer object supports these optional operations, it  
4103 MUST support the "reference-uri-schemes-supported" Printer attribute with at least the following schemed  
4104 URI value:

4105 'ftp': The Printer object will use an FTP 'get' operation as defined in RFC 2228 [RFC2228] using FTP  
4106 URLs as defined by [RFC2396] and[RFC2316].  
4107

4108 The Printer object MAY OPTIONALLY support other URI schemes (see section 4.1.6).

4109 **4.4.28 pdl-override-supported (type2 keyword)**

4110 This REQUIRED Printer attribute expresses the ability for a particular Printer implementation to either  
4111 attempt to override document data instructions with IPP attributes or not.

4112 This attribute takes on the following keyword values:

- 4113 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values take  
4114 precedence over embedded instructions in the document data, however there is no guarantee.
- 4115 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP attribute  
4116 values take precedence over embedded instructions in the document data.  
4117

4118 Section 15 contains a full description of how this attribute interacts with and affects other IPP attributes,  
4119 especially the "ipp-attribute-fidelity" attribute.

4120 **4.4.29 printer-up-time (integer(1:MAX))**

4121 This REQUIRED Printer attribute indicates the amount of time (in seconds) that this Printer instance has  
4122 been up and running. The value is a monotonically increasing value starting from 1 when the Printer object  
4123 is started-up (initialized, booted, etc.). This value is used to populate the Event Time Job Description Job  
4124 attributes "time-at-creation", "time-at-processing", and "time-at-completed" (see section 4.3.14).

4125 If the Printer object goes down at some value 'n', and comes back up, the implementation MAY:

- 4126 1. Know how long it has been down, and resume at some value greater than 'n', or

4127 2. Restart from 1.

4128 In other words, if the device or devices that the Printer object is representing are restarted or power cycled,  
4129 the Printer object MAY continue counting this value or MAY reset this value to 1 depending on  
4130 implementation. However, if the Printer object software ceases running, and restarts without knowing the  
4131 last value for "printer-up-time", the implementation MUST reset this value to 1. If this value is reset and  
4132 the Printer has persistent jobs, the Printer MUST reset the "time-at-xxx(integer) Event Time Job  
4133 Description attributes according to Section 4.3.14. An implementation MAY use both implementation  
4134 alternatives, depending on warm versus cold start, respectively.

#### 4135 **4.4.30 printer-current-time (dateTime)**

4136 This Printer attribute indicates the current date and time. This value is used to populate the Event Time Job  
4137 Description attributes: "time-at-creation", "time-at-processing", and "time-at-completed" (see Section  
4138 4.3.14).

4139 The date and time is obtained on a "best efforts basis" and does not have to be that precise in order to work  
4140 in practice. A Printer implementation sets the value of this attribute by obtaining the date and time via  
4141 some implementation-dependent means, such as getting the value from a network time server, initialization  
4142 at time of manufacture, or setting by an administrator. See [IPP-IIG] for examples. If an implementation  
4143 supports this attribute and the implementation knows that it has not yet been set, then the implementation  
4144 MUST return the value of this attribute using the out-of-band 'no-value' meaning not configured. See the  
4145 beginning of section 4.1.

4146 The time zone of this attribute NEED NOT be the time zone used by people located near the Printer object  
4147 or device. The client MUST NOT expect that the time zone of any received 'dateTime' value to be in the  
4148 time zone of the client or in the time zone of the people located near the printer.

4149 The client SHOULD display any dateTime attributes to the user in client local time by converting the  
4150 'dateTime' value returned by the server to the time zone of the client, rather than using the time zone  
4151 returned by the Printer in attributes that use the 'dateTime' attribute syntax.

#### 4152 **4.4.31 multiple-operation-time-out (integer(1:MAX))**

4153 This Printer attributes identifies the minimum time (in seconds) that the Printer object waits for additional  
4154 Send-Document or Send-URI operations to follow a still-open Job object before taking any recovery  
4155 actions, such as the ones indicated in section 3.3.1. If the Printer object supports the Create-Job and Send-  
4156 Document operations (see section 3.2.4 and 3.3.1), it MUST support this attribute.

4157 It is RECOMMENDED that vendors supply a value for this attribute that is between 60 and 240 seconds.  
4158 An implementation MAY allow a system administrator to set this attribute (by means outside this IPP/1.1  
4159 document). If so, the system administrator MAY be able to set values outside this range.

#### 4160 **4.4.32 compression-supported (1setOf type3 keyword)**

4161 This REQUIRED Printer attribute identifies the set of supported compression algorithms for document  
4162 data. Compression only applies to the document data; compression does not apply to the encoding of the  
4163 IPP operation itself. The supported values are used to validate the client supplied "compression" operation  
4164 attributes in Print-Job, Send-Document, and Send-URI requests.

4165 Standard keyword values are :

4166 'none': no compression is used.

4167 'deflate': ZIP public domain inflate/deflate) compression technology [RFC1951]

4168 'gzip' GNU zip compression technology described in RFC 1952 [RFC1952].

4169 'compress': UNIX compression technology [RFC1977]

4170

#### 4171 **4.4.33 job-k-octets-supported (rangeOfInteger(0:MAX))**

4172 This Printer attribute specifies the upper and lower bounds of total sizes of jobs in K octets, i.e., in units of  
4173 1024 octets. The supported values are used to validate the client supplied "job-k-octets" operation attributes  
4174 in create requests. The corresponding job description attribute "job-k-octets" is defined in section 4.3.17.1.

#### 4175 **4.4.34 job-impressions-supported (rangeOfInteger(0:MAX))**

4176 This Printer attribute specifies the upper and lower bounds for the number of impressions per job. The  
4177 supported values are used to validate the client supplied "job-impressions" operation attributes in create  
4178 requests. The corresponding job description attribute "job-impressions" is defined in section 4.3.17.2.

#### 4179 **4.4.35 job-media-sheets-supported (rangeOfInteger(0:MAX))**

4180 This Printer attribute specifies the upper and lower bounds for the number of media sheets per job. The  
4181 supported values are used to validate the client supplied "job-media-sheets" operation attributes in create  
4182 requests. The corresponding Job attribute "job-media-sheets" is defined in section 4.3.17.3.

#### 4183 **4.4.36 pages-per-minute (integer(0:MAX))**

4184 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number which  
4185 may be generated by this printer (e.g., simplex, black-and-white). This attribute is informative, not a  
4186 service guarantee. Generally, it is the value used in the marketing literature to describe the device.

4187 A value of 0 indicates a device that takes more than two minutes to process a page.

#### 4188 **4.4.37 pages-per-minute-color (integer(0:MAX))**

4189 This Printer attributes specifies the nominal number of pages per minute to the nearest whole number which  
4190 may be generated by this printer when printing color (e.g., simplex, color). For purposes of this attribute,  
4191 "color" means the same as for the "color-supported" attribute, namely, the device is capable of any type of

4192 color printing at all, including highlight color. This attribute is informative, not a service guarantee.  
4193 Generally, it is the value used in the marketing literature to describe the color capabilities of this device.

4194 A value of 0 indicates a device that takes more than two minutes to process a page.

4195 If a color device has several color modes, it MAY use the pages-per-minute value for this attribute that  
4196 corresponds to the mode that produces the highest number.

4197 Black and white only printers MUST NOT support this attribute. If this attribute is present, then the "color-  
4198 supported" Printer description attribute MUST be present and have a 'true' value.

4199 The values of these two attributes returned by the Get-Printer-Attributes operation MAY be affected by the  
4200 "document-format" attribute supplied by the client in the Get-Printer-Attributes request. In other words, the  
4201 implementation MAY have different speeds depending on the document format being processed. See  
4202 section 3.2.5.1 Get-Printer-Attributes.

## 4203 **5. Conformance**

4204 This section describes conformance issues and requirements. This document introduces model entities such  
4205 as objects, operations, attributes, attribute syntaxes, and attribute values. These conformance sections  
4206 describe the conformance requirements which apply to these model entities.

### 4207 **5.1 Client Conformance Requirements**

4208 This section describes the conformance requirements for a client (see section 2.1), whether it be:

- 4209 1. contained within software controlled by an end user, e.g. activated by the "Print" menu item in an  
4210 application that sends IPP requests or
- 4211 2. the print server component that sends IPP requests to either an output device or another  
4212 "downstream" print server.

4213 A conforming client MUST support all REQUIRED operations as defined in this document. For each  
4214 attribute included in an operation request, a conforming client MUST supply a value whose type and value  
4215 syntax conforms to the requirements of the Model document as specified in Sections 3 and 4. A  
4216 conforming client MAY supply any IETF standards track extensions and/or vendor extensions in an  
4217 operation request, as long as the extensions meet the requirements in Section 6.

4218 Otherwise, there are no conformance requirements placed on the user interfaces provided by IPP clients or  
4219 their applications. For example, one application might not allow an end user to submit multiple documents  
4220 per job, while another does. One application might first query a Printer object in order to supply a graphical  
4221 user interface (GUI) dialogue box with supported and default values whereas a different implementation  
4222 might not.

4223 When sending a request, an IPP client NEED NOT supply any attributes that are indicated as  
4224 OPTIONALLY supplied by the client.

4225 A client MUST be able to accept any of the attribute syntaxes defined in Section 4.1, including their full  
4226 range, that may be returned to it in a response from a Printer object. In particular for each attribute that the  
4227 client supports whose attribute syntax is 'text', the client MUST accept and process both the  
4228 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that the client supports  
4229 whose attribute syntax is 'name', the client MUST accept and process both the 'nameWithoutLanguage' and  
4230 'nameWithLanguage' forms. For presentation purposes, truncation of long attribute values is not  
4231 recommended. A recommended approach would be for the client implementation to allow the user to scroll  
4232 through long attribute values.

4233 A response MAY contain attribute groups, attributes, attribute syntaxes, values, and status codes that the  
4234 client does not expect. Therefore, a client implementation MUST gracefully handle such responses and not  
4235 refuse to inter-operate with a conforming Printer that is returning IETF standards track extension or vendor  
4236 extensions, including attribute groups, attributes, attribute syntaxes, attribute values, status codes, and out-  
4237 of-band attribute values that conform to Section 6. Clients may choose to ignore any parameters, attributes,  
4238 attribute syntaxes, or values that they do not understand.

4239 While a client is sending data to a printer, it SHOULD do its best to prevent a channel from being closed by  
4240 a lower layer when the channel is blocked (i.e. flow-controlled off) for whatever reason, e.g. 'out of paper'  
4241 or 'job ahead hasn't freed up enough memory'. However, the layer that launched the print submission (e.g.  
4242 an end user) MAY close the channel in order to cancel the job. When a client closes a channel, a Printer  
4243 MAY print all or part of the received portion of the document. See the "Encoding and Transport" document  
4244 [IPP-PRO] for more details.

4245 A client MUST support Client Authentication as defined in the IPP/1.1 Encoding and Transport document  
4246 [IPP-PRO]. A client SHOULD support Operation Privacy and Server Authentication as defined in the  
4247 IPP/1.1 Encoding and Transport document [IPP-PRO]. See also section 8 of this document.

## 4248 **5.2 IPP Object Conformance Requirements**

4249 This section specifies the conformance requirements for conforming implementations of IPP objects (see  
4250 section 2). These requirements apply to an IPP object whether it is:

4251 (1) an (embedded) device component that accepts IPP requests and controls the device or

4252 (2) a component of a print server that accepts IPP requests (where the print server control one or  
4253 more networked devices using IPP or other protocols).

### 4254 **5.2.1 Objects**

4255 Conforming implementations MUST implement all of the model objects as defined in this document in the  
4256 indicated sections:

4257 Section 2.1 - Printer Object

4258 Section 2.2 - Job Object

## 4259 5.2.2 Operations

4260 Conforming IPP object implementations **MUST** implement all of the **REQUIRED** model operations,  
4261 including **REQUIRED** responses, as defined in this document in the indicated sections:

4262 For a Printer object:

4263	Print-Job (section 3.2.1)	<b>REQUIRED</b>
4264	Print-URI (section 3.2.2)	<b>OPTIONAL</b>
4265	Validate-Job (section 3.2.3)	<b>REQUIRED</b>
4266	Create-Job (section 3.2.4)	<b>OPTIONAL</b>
4267	Get-Printer-Attributes (section 3.2.5)	<b>REQUIRED</b>
4268	Get-Jobs (section 3.2.6)	<b>REQUIRED</b>
4269	Pause-Printer (section 3.2.7)	<b>OPTIONAL</b>
4270	Resume-Printer (section 3.2.8)	<b>OPTIONAL</b>
4271	Purge-Jobs (section 3.2.9)	<b>OPTIONAL</b>

4272

4273 For a Job object:

4274	Send-Document (section 3.3.1)	<b>OPTIONAL</b>
4275	Send-URI (section 3.3.2)	<b>OPTIONAL</b>
4276	Cancel-Job (section 3.3.3)	<b>REQUIRED</b>
4277	Get-Job-Attributes (section 3.3.4)	<b>REQUIRED</b>
4278	Hold-Job (section 3.3.5)	<b>OPTIONAL</b>
4279	Release-Job (section 3.3.6)	<b>OPTIONAL</b>
4280	Restart-Job (section 3.3.7)	<b>OPTIONAL</b>

4281

4282 Conforming IPP objects **MUST** support all **REQUIRED** operation attributes and all values of such  
4283 attributes if so indicated in the description. Conforming IPP objects **MUST** ignore all unsupported or  
4284 unknown operation attributes or operation attribute groups received in a request, but **MUST** reject a request  
4285 that contains a supported operation attribute that contains an unsupported value.

4286 Conforming IPP objects **MAY** return operation responses that contain attributes groups, attributes names,  
4287 attribute syntaxes, attribute values, and status codes that are extensions to this standard. The additional  
4288 attribute groups **MAY** occur in any order.

4289 The following section on object attributes specifies the support required for object attributes.

## 4290 5.2.3 IPP Object Attributes

4291 Conforming IPP objects **MUST** support all of the **REQUIRED** object attributes, as defined in this document  
4292 in the indicated sections.

4293 If an object supports an attribute, it MUST support only those values specified in this document or through  
4294 the extension mechanism described in section 5.2.4. It MAY support any non-empty subset of these values.  
4295 That is, it MUST support at least one of the specified values and at most all of them.

#### 4296 **5.2.4 Versions**

4297 IPP/1.1 clients MUST meet the conformance requirements for clients specified in this document and [IPP-  
4298 PRO]. IPP/1.1 clients MUST send requests containing a "version-number" parameter with a '1.1' value.

4299 IPP/1.1 Printer and Job objects MUST meet the conformance requirements for IPP objects specified in this  
4300 document and [IPP-PRO]. IPP/1.1 objects MUST accept requests containing a "version-number"  
4301 parameter with a '1.1' value (or reject the request if the operation is not supported).

4302 It is beyond the scope of this specification to mandate conformance with previous versions. IPP/1.1 was  
4303 deliberately designed, however, to make supporting previous versions easy. It is worth noting that, at the  
4304 time of composing this specification (1999), we would expect IPP/1.1 Printer implementations to:

4305         understand any valid request in the format of IPP/1.0, or 1.1;

4306         respond appropriately with a response containing the same "version-number" parameter value used  
4307         by the client in the request.

4308 And we would expect IPP/1.1 clients to:

4309         understand any valid response in the format of IPP/1.0, or 1.1.

4310 It is recommended that IPP/1.1 clients try supplying alternate version numbers if they receive a 'server-  
4311 error-version-not-supported' error return in a response.

#### 4312 **5.2.5 Extensions**

4313 A conforming IPP object MAY support IETF standards track extensions and vendor extensions, as long as  
4314 the extensions meet the requirements specified in Section 6.

4315 For each attribute included in an operation response, a conforming IPP object MUST return a value whose  
4316 type and value syntax conforms to the requirement of the Model document as specified in Sections 3 and 4.

#### 4317 **5.2.6 Attribute Syntaxes**

4318 An IPP object MUST be able to accept any of the attribute syntaxes defined in Section 4.1, including their  
4319 full range, in any operation in which a client may supply attributes or the system administrator may  
4320 configure attributes (by means outside the scope of this IPP/1.1 document). In particular for each attribute  
4321 that the IPP object supports whose attribute syntax is 'text', the IPP object MUST accept and process both  
4322 the 'textWithoutLanguage' and 'textWithLanguage' forms. Similarly, for each attribute that the IPP object  
4323 supports whose attribute syntax is 'name', the IPP object MUST accept and process both the  
4324 'nameWithoutLanguage' and 'nameWithLanguage' forms. Furthermore, an IPP object MUST return

4325 attributes to the client in operation responses that conform to the syntax specified in Section 4.1, including  
4326 their full range if supplied previously by a client.

### 4327 **5.2.7 Security**

4328 An IPP Printer implementation SHOULD contain support for Client Authentication as defined in the  
4329 IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer implementation MAY allow an  
4330 administrator to configure the Printer so that all, some, or none of the users are authenticated. See also  
4331 section 8 of this document.

4332 An IPP Printer implementation SHOULD contain support for Operation Privacy and Server Authentication  
4333 as defined in the IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer implementation MAY  
4334 allow an administrator to configure the degree of support for Operation Privacy and Server Authentication.  
4335 See also section 8 of this document.

4336 Security MUST NOT be compromised when a client supplies a lower "version-number" parameter in a  
4337 request. For example, if an IPP/1.1 conforming Printer object accepts version '1.0' requests and is  
4338 configured to enforce Digest Authentication, it MUST do the same for a version '1.0' request.

## 4339 **5.3 Charset and Natural Language Requirements**

4340 All clients and IPP objects MUST support the 'utf-8' charset as defined in section 4.1.7.

4341 IPP objects MUST be able to accept any client request which correctly uses the "attributes-natural-  
4342 language" operation attribute or the Natural Language Override mechanism on any individual attribute  
4343 whether or not the natural language is supported by the IPP object. If an IPP object supports a natural  
4344 language, then it MUST be able to translate (perhaps by table lookup) all generated 'text' or 'name' attribute  
4345 values into one of the supported languages (see section 3.1.4). That is, the IPP object that supports a  
4346 natural language NEED NOT be a general purpose translator of any arbitrary 'text' or 'name' value supplied  
4347 by the client into that natural language. However, the object MUST be able to translate (automatically  
4348 generate) any of its own attribute values and messages into that natural language.

## 4349 **6. IANA Considerations**

4350 This section describes the procedures for defining semantics for the following IETF standards track  
4351 extensions and vendor extensions to the IPP/1.1 Model and Semantics document:

- 4352 1. keyword attribute values
- 4353 2. enum attribute values
- 4354 3. attributes
- 4355 4. attribute syntaxes
- 4356 5. operations
- 4357 6. attribute groups



- 4358 7. status codes
- 4359 8. out-of-band attribute values
- 4360

4361 Extensions registered for use with IPP/1.1 are OPTIONAL for client and IPP object conformance to the  
4362 IPP/1.1 "Model and Semantics" document (this document).

4363 These extension procedures are aligned with the guidelines as set forth by the IESG [IANA-CON]. Section  
4364 11 describes how to propose new registrations for consideration. IANA will reject registration proposals  
4365 that leave out required information or do not follow the appropriate format described in Section 11. The  
4366 IPP/1.1 Model and Semantics document may also be extended by an appropriate RFC that specifies any of  
4367 the above extensions.

## 4368 **6.1 Typed 'keyword' and 'enum' Extensions**

4369 IPP allows for 'keyword' and 'enum' extensions (see sections 4.1.2.3 and 4.1.4). This document uses  
4370 prefixes to the 'keyword' and 'enum' basic attribute syntax type in order to communicate extra information  
4371 to the reader through its name. This extra information is not represented in the protocol because it is  
4372 unimportant to a client or Printer object. The list below describes the prefixes and their meaning.

4373 "type1": This IPP specification document must be revised (or another IETF standards track document  
4374 which augments this document) to add a new keyword or a new enum. No vendor defined  
4375 keywords or enums are allowed.

4376  
4377 "type2": Implementers can, at any time, add new keyword or enum values by proposing the complete  
4378 specification to IANA:

4379  
4380 iana@iana.org

4381  
4382 IANA will forward the registration proposal to the IPP Designated Expert who will review the  
4383 proposal with a mailing list that the Designated Expert keeps for this purpose. Initially, that list will  
4384 be the mailing list used by the IPP WG:

4385  
4386 ipp@pwg.org

4387  
4388 even after the IPP WG is disbanded as permitted by [IANA-CON]. The IPP Designated Expert is  
4389 appointed by the IESG Area Director responsible for IPP, according to [IANA-CON].

4390  
4391 When a type2 keyword or enum is approved, the IPP Designated Expert becomes the point of  
4392 contact for any future maintenance that might be required for that registration.

4393  
4394 "type3": Implementers can, at any time, add new keyword and enum values by submitting the complete  
4395 specification to IANA as for type2 who will forward the proposal to the IPP Designated Expert.  
4396 While no additional technical review is required, the IPP Designated Expert may, at his/her

4397 discretion, forward the proposal to the same mailing list as for type2 registrations for advice and  
4398 comment.

4399  
4400 When a type3 keyword or enum is approved by the IPP Designated Expert, the original proposer  
4401 becomes the point of contact for any future maintenance that might be required for that registration.  
4402

4403 For type2 and type3 keywords, the proposer includes the name of the keyword in the registration proposal  
4404 and the name is part of the technical review.

4405 After type2 and type3 enums specifications are approved, the IPP Designated Expert in consultation with  
4406 IANA assigns the next available enum number for each enum value.

4407 IANA will publish approved type2 and type3 keyword and enum attributes value registration specifications  
4408 in:

4409 `ftp.isi.edu/iana/assignments/ipp/attribute-values/xxx/yyy.txt`

4410 where xxx is the attribute name that specifies the initial values and yyy.txt is a descriptive file name that  
4411 contains one or more enums or keywords approved at the same time. For example, if several additional  
4412 enums for stapling are approved for use with the "finishings" attribute (and "finishings-default" and  
4413 "finishings-supported" attributes), IANA will publish the additional values in the file:

4414 `ftp.isi.edu/iana/assignments/ipp/attribute-values/finishings/stapling.txt`

4415 Note: Some attributes are defined to be: 'type3 keywords' | 'name' which allows for attribute values to be  
4416 extended by a site administrator with administrator defined names. Such names are not registered with  
4417 IANA.

4418 By definition, each of the three types above assert some sort of registry or review process in order for  
4419 extensions to be considered valid. Each higher numbered level (1, 2, 3) tends to be decreasingly less  
4420 stringent than the previous level. Therefore, any typeN value MAY be registered using a process for some  
4421 typeM where M is less than N, however such registration is NOT REQUIRED. For example, a type3 value  
4422 MAY be registered in a type 1 manner (by being included in a future version of an IPP specification),  
4423 however, it is NOT REQUIRED.

4424 This document defines keyword and enum values for all of the above types, including type3 keywords.

4425 For vendor keyword extensions, implementers SHOULD use keywords with a suitable distinguishing  
4426 prefix, such as "xxx-" where xxx follows the syntax rules for keywords (see section 4.1.3) and is the  
4427 (lowercase) fully qualified company name registered with IANA for use in domain names [RFC1035]. For  
4428 example, if the company XYZ Corp. had obtained the domain name "XYZ.com", then a vendor keyword  
4429 'abc' would be: 'xyz.com-abc'.

4430 Note: RFC 1035 [RFC1035] indicates that while upper and lower case letters are allowed in domain names,  
4431 no significance is attached to the case. That is, two names with the same spelling but different case are to  
4432 be treated as if identical. Also, the labels in a domain name must follow the rules for ARPANET host

4433 names: They must start with a letter, end with a letter or digit, and have as interior characters only letters,  
4434 digits, and hyphen. Labels must be 63 characters or less. Labels are separated by the "." character.

4435 For vendor enum extensions, implementers **MUST** use values in the reserved integer range which is 2\*\*30  
4436 to 2\*\*31-1.

## 4437 **6.2 Attribute Extensibility**

4438 Attribute names (see section 4.1.3) are type2 keywords. Therefore, new attributes may be registered and  
4439 have the same status as attributes in this document by following the type2 extension rules. For vendor  
4440 attribute extensions, implementers **SHOULD** use keywords with a suitable distinguishing prefix as  
4441 described in Section 6.1.

4442 IANA will publish approved attribute registration specifications as separate files:

4443 `ftp.isi.edu/iana/assignments/ipp/attributes/xxx-yyy.txt`

4444 where "xxx-yyy" is the new attribute name.

4445 If a new Printer object attribute is defined and its values can be affected by a specific document format, its  
4446 specification needs to contain the following sentence:

4447 "The value of this attribute returned in a Get-Printer-Attributes response **MAY** depend on the  
4448 "document-format" attribute supplied (see Section 3.2.5.1)."

4449 If the specification does not, then its value in the Get-Printer-Attributes response **MUST NOT** depend on  
4450 the "document-format" supplied in the request. When a new Job Template attribute is registered, the value  
4451 of the Printer attributes **MAY** vary with "document-format" supplied in the request without the  
4452 specification having to indicate so.

## 4453 **6.3 Attribute Syntax Extensibility**

4454 Attribute syntaxes (see section 4.1) are like type2 enums. Therefore, new attribute syntaxes may be  
4455 registered and have the same status as attribute syntaxes in this document by following the type2 extension  
4456 rules described in Section 6.1. The initial set of value codes that identify each of the attribute syntaxes have  
4457 been assigned in the "Encoding and Transport" document [IPP-PRO], including a designated range for  
4458 vendor extension.

4459 For attribute syntaxes, the IPP Designated Expert in consultation with IANA assigns the next attribute  
4460 syntax code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute  
4461 syntax registration specifications as separate files:

4462 `ftp.isi.edu/iana/assignments/ipp/attribute-syntaxes/xxx-yyy.txt`

4463 where 'xxx-yyy' is the new attribute syntax name.

## 4464 6.4 Operation Extensibility

4465 Operations (see section 3) may also be registered following the type2 procedures described in Section 6.1,  
4466 though major new operations will usually be done by a new standards track RFC that augments this  
4467 document. For vendor operation extensions, implementers MUST use the range for the "operation-id" in  
4468 requests specified in Section 4.4.15 "operations-supported" Printer attribute.

4469 For operations, the IPP Designated Expert in consultation with IANA assigns the next operation-id code as  
4470 specified in Section 4.4.15. IANA will publish approved operation registration specifications as separate  
4471 files:

4472 ftp.isi.edu/iana/assignments/ipp/operations/Xxx-Yyy.txt

4473 where "Xxx-Yyy" is the new operation name.

## 4474 6.5 Attribute Group Extensibility

4475 Attribute groups (see section 3.1.3) passed in requests and responses may be registered following the type2  
4476 procedures described in Section 6.1. The initial set of attribute group tags have been assigned in the  
4477 "Encoding and Transport" document [IPP-PRO], including a designated range for vendor extension.

4478 For attribute groups, the IPP Designated Expert in consultation with IANA assigns the next attribute group  
4479 tag code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute group  
4480 registration specifications as separate files:

4481 ftp.isi.edu/iana/assignments/ipp/attribute-group-tags/xxx-yyy-tag.txt

4482 where 'xxx-yyy-tag' is the new attribute group tag name.

## 4483 6.6 Status Code Extensibility

4484 Operation status codes (see section 3.1.6.1) may also be registered following the type2 procedures described  
4485 in Section 6.1. The values for status codes are allocated in ranges as specified in Section 14 for each status  
4486 code class:

4487 "informational" - Request received, continuing process  
4488 "successful" - The action was successfully received, understood, and accepted  
4489 "redirection" - Further action must be taken in order to complete the request  
4490 "client-error" - The request contains bad syntax or cannot be fulfilled  
4491 "server-error" - The IPP object failed to fulfill an apparently valid request  
4492

4493 For vendor operation status code extensions, implementers MUST use the top of each range as specified in  
4494 Section 13.

4495 For operation status codes, the IPP Designated Expert in consultation with IANA assigns the next status  
4496 code in the appropriate class range as specified in Section 13. IANA will publish approved status code  
4497 registration specifications as separate files:

4498         ftp.isi.edu/iana/assignments/ipp/status-codes/xxx-yyy.txt

4499 where "xxx-yyy" is the new operation status code keyword.

## 4500 **6.7 Out-of-band Attribute Value Extensibility**

4501 Out-of-band attribute values (see the beginning of section 4.1) passed in requests and responses may be  
4502 registered following the type2 procedures described in Section 6.1. The initial set of out-of-band attribute  
4503 value tags have been assigned in the "Encoding and Transport" document [IPP-PRO].

4504 For out-of-band attribute value tags, the IPP Designated Expert in consultation with IANA assigns the next  
4505 out-of-band attribute value tag code in the appropriate range as specified in [IPP-PRO]. IANA will publish  
4506 approved out-of-band attribute value tags registration specifications as separate files:

4507         ftp.isi.edu/iana/assignments/ipp/out-of-band-attribute-value-tags/xxx-yyy-tag.txt

4508 where 'xxx-yyy-tag' is the new out-of-band attribute value tag name.

## 4509 **6.8 Registration of MIME types/sub-types for document-formats**

4510 The "document-format" attribute's syntax is 'mimeMediaType'. This means that valid values are Internet  
4511 Media Types (see Section 4.1.9). RFC 2045 [RFC2045] defines the syntax for valid Internet media types.  
4512 IANA is the registry for all Internet media types.

## 4513 **6.9 Registration of charsets for use in 'charset' attribute values**

4514 The "attributes-charset" attribute's syntax is 'charset'. This means that valid values are charsets names.  
4515 When a charset in the IANA registry has more than one name (alias), the name labeled as "(preferred  
4516 MIME name)", if present, MUST be used (see Section 4.1.7). IANA is the registry for charsets following  
4517 the procedures of [RFC2278].

## 4518 **7. Internationalization Considerations**

4519 Some of the attributes have values that are text strings and names which are intended for human  
4520 understanding rather than machine understanding (see the 'text' and 'name' attribute syntaxes in Sections  
4521 4.1.1 and 4.1.2).

4522 In each operation request, the client

- 4523       - identifies the charset and natural language of the request which affects each supplied 'text' and 'name'  
4524       attribute value, and  
4525       - requests the charset and natural language for attributes returned by the IPP object in operation  
4526       responses (as described in Section 3.1.4.1).  
4527

4528 In addition, the client MAY separately and individually identify the Natural Language Override of a  
4529 supplied 'text' or 'name' attribute using the 'textWithLanguage' and 'nameWithLanguage' technique  
4530 described section 4.1.1.2 and 4.1.2.2 respectively.

4531 All IPP objects MUST support the UTF-8 [RFC2279] charset in all 'text' and 'name' attributes supported. If  
4532 an IPP object supports more than the UTF-8 charset, the object MUST convert between them in order to  
4533 return the requested charset to the client according to Section 3.1.4.2. If an IPP object supports more than  
4534 one natural language, the object SHOULD return 'text' and 'name' values in the natural language requested  
4535 where those values are generated by the Printer (see Section 3.1.4.1).

4536 For Printers that support multiple charsets and/or multiple natural languages in 'text' and 'name' attributes,  
4537 different jobs may have been submitted in differing charsets and/or natural languages. All responses MUST  
4538 be returned in the charset requested by the client. However, the Get-Jobs operation uses the  
4539 'textWithLanguage' and 'nameWithLanguage' mechanism to identify the differing natural languages with  
4540 each job attribute returned.

4541 The Printer object also has configured charset and natural language attributes. The client can query the  
4542 Printer object to determine the list of charsets and natural languages supported by the Printer object and  
4543 what the Printer object's configured values are. See the "charset-configured", "charset-supported", "natural-  
4544 language-configured", and "generated-natural-language-supported" Printer description attributes for more  
4545 details.

4546 The "charset-supported" attributed identifies the supported charsets. If a charset is supported, the IPP  
4547 object MUST be capable of converting to and from that charset into any other supported charset. In many  
4548 cases, an IPP object will support only one charset and it MUST be the UTF-8 charset.

4549 The "charset-configured" attribute identifies the one supported charset which is the native charset given the  
4550 current configuration of the IPP object (administrator defined).

4551 The "generated-natural-language-supported" attribute identifies the set of supported natural languages for  
4552 generated messages; it is not related to the set of natural languages that must be accepted for client supplied  
4553 'text' and 'name' attributes. For client supplied 'text' and 'name' attributes, an IPP object MUST accept ALL  
4554 supplied natural languages. Just because a Printer object is currently configured to support 'en-us' natural  
4555 language does not mean that the Printer object should reject a job if the client supplies a job name that is in  
4556 'fr-ca'.

4557 The "natural-language-configured" attribute identifies the one supported natural language for generated  
4558 messages which is the native natural language given the current configuration of the IPP object  
4559 (administrator defined).

4560 Attributes of type 'text' and 'name' are populated from different sources. These attributes can be categorized  
4561 into following groups (depending on the source of the attribute):

- 4562 1. Some attributes are supplied by the client (e.g., the client supplied "job-name", "document-name",  
4563 and "requesting-user-name" operation attributes along with the corresponding Job object's "job-  
4564 name" and "job-originating-user-name" attributes). The IPP object MUST accept these attributes in  
4565 any natural language no matter what the set of supported languages for generated messages
- 4566 2. Some attributes are supplied by the system administrator (e.g., the Printer object's "printer-name" and  
4567 "printer-location" attributes). These too can be in any natural language. If the natural language for  
4568 these attributes is different than what a client requests, then they must be reported using the Natural  
4569 Language Override mechanism.
- 4570 3. Some attributes are supplied by the device manufacturer (e.g., the Printer object's "printer-make-and-  
4571 model" attribute). These too can be in any natural language. If the natural language for these  
4572 attributes is different than what a client requests, then they must be reported using the Natural  
4573 Language Override mechanism.
- 4574 4. Some attributes are supplied by the operator (e.g., the Job object's "job-message-from-operator"  
4575 attribute). These too can be in any natural language. If the natural language for these attributes is  
4576 different than what a client requests, then they must be reported using the Natural Language  
4577 Override mechanism.
- 4578 5. Some attributes are generated by the IPP object (e.g., the Job object's "job-state-message" attribute,  
4579 the Printer object's "printer-state-message" attribute, and the "status-message" operation attribute).  
4580 These attributes can only be in one of the "generated-natural-language-supported" natural  
4581 languages. If a client requests some natural language for these attributes other than one of the  
4582 supported values, the IPP object SHOULD respond using the value of the "natural-language-  
4583 configured" attribute (using the Natural Language Override mechanism if needed).

4584  
4585 The 'text' and 'name' attributes specified in this version of this document (additional ones will be registered  
4586 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 (1setOf text)	Job or Printer object - see rule 1
job-document-access-errors (1setOf 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

4587 Rule 1 - Neither the Printer nor the client localizes these message attributes, since they are intended for use  
4588 by the system administrator or other experienced technical persons.



4589

4590

## 8. Security Considerations

4591 It is difficult to anticipate the security risks that might exist in any given IPP environment. For example, if  
4592 IPP is used within a given corporation over a private network, the risks of exposing document data may be  
4593 low enough that the corporation will choose not to use encryption on that data. However, if the connection  
4594 between the client and the IPP object is over a public network, the client may wish to protect the content of  
4595 the information during transmission through the network with encryption.

4596 Furthermore, the value of the information being printed may vary from one IPP environment to the next.  
4597 Printing payroll checks, for example, would have a different value than printing public information from a  
4598 file. There is also the possibility of denial-of-service attacks, but denial-of-service attacks against printing  
4599 resources are not well understood and there is no published precedents regarding this scenario.

4600 Once the authenticated identity of the requester has been supplied to the IPP object, the object uses that  
4601 identity to enforce any authorization policy that might be in place. For example, one site's policy might be  
4602 that only the job owner is allowed to cancel a job. The details and mechanisms to set up a particular access  
4603 control policy are not part of IPP/1.1, and must be established via some other type of administrative or  
4604 access control framework. However, there are operation status codes that allow an IPP server to return  
4605 information back to a client about any potential access control violations for an IPP object.

4606 During a create operation, the client's identity is recorded in the Job object in an implementation-defined  
4607 attribute. This information can be used to verify a client's identity for subsequent operations on that Job  
4608 object in order to enforce any access control policy that might be in effect. See section 8.3 below for more  
4609 details.

4610 Since the security levels or the specific threats that an IPP system administrator may be concerned with  
4611 cannot be anticipated, IPP **MUST** be capable of operating with different security mechanisms and security  
4612 policies as required by the individual installation. Security policies might vary from very strong, to very  
4613 weak, to none at all, and corresponding security mechanisms will be required.

4614

### 8.1 Security Scenarios

4615 The following sections describe specific security attacks for IPP environments. Where examples are  
4616 provided they should be considered illustrative of the environment and not an exhaustive set. Not all of  
4617 these environments will necessarily be addressed in initial implementations of IPP.

4618

#### 8.1.1 Client and Server in the Same Security Domain

4619 This environment is typical of internal networks where traditional office workers print the output of  
4620 personal productivity applications on shared work-group printers, or where batch applications print their  
4621 output on large production printers. Although the identity of the user may be trusted in this environment, a

4622 user might want to protect the content of a document against such attacks as eavesdropping, replaying or  
4623 tampering.

### 4624 **8.1.2 Client and Server in Different Security Domains**

4625 Examples of this environment include printing a document created by the client on a publicly available  
4626 printer, such as at a commercial print shop; or printing a document remotely on a business associate's  
4627 printer. This latter operation is functionally equivalent to sending the document to the business associate as  
4628 a facsimile. Printing sensitive information on a Printer in a different security domain requires strong  
4629 security measures. In this environment authentication of the printer is required as well as protection against  
4630 unauthorized use of print resources. Since the document crosses security domains, protection against  
4631 eavesdropping and document tampering are also required. It will also be important in this environment to  
4632 protect Printers against "spamming" and malicious document content.

### 4633 **8.1.3 Print by Reference**

4634 When the document is not stored on the client, printing can be done by reference. That is, the print request  
4635 can contain a reference, or pointer, to the document instead of the actual document itself (see sections 3.2.2  
4636 and 3.3.2). Standard methods currently do not exist for remote entities to "assume" the credentials of a  
4637 client for forwarding requests to a 3rd party. It is anticipated that Print-By-Reference will be used to access  
4638 "public" documents and that sophisticated methods for authenticating "proxies" is not specified in this  
4639 document.

## 4640 **8.2 URIs in Operation, Job, and Printer attributes**

4641 The "printer-uri-supported" attribute contains the Printer object's URI(s). Its companion attribute, "uri-  
4642 security-supported", identifies the security mechanism used for each URI listed in the "printer-uri-  
4643 supported" attribute. For each Printer operation request, a client MUST supply only one URI in the  
4644 "printer-uri" operation attribute. In other words, even though the Printer supports more than one URI, the  
4645 client only interacts with the Printer object using one of its URIs. This duality is not needed for Job objects,  
4646 since the Printer objects is the factory for Job objects, and the Printer object will generate the correct URI  
4647 for new Job objects depending on the Printer object's security configuration.

## 4648 **8.3 URIs for each authentication mechanisms**

4649 Each URI has an authentication mechanism associated with it. If the URI is the i'th element of "printer-uri-  
4650 supported", then authentication mechanism is the "i th" element of "uri-authentication-supported". For a list  
4651 of possible authentication mechanisms, see section 4.4.2.

4652 The Printer object uses an authentication mechanism to determine the name of the user performing an  
4653 operation. This user is called the "authenticated user". The credibility of authentication depends on the  
4654 mechanism that the Printer uses to obtain the user's name. When the authentication mechanism is 'none', all  
4655 authenticated users are "anonymous".

4656 During job creation operations, the Printer initializes the value of the "job-originating-user-name" attribute  
4657 (see section 4.3.6) to be the authenticated user. The authenticated user in this case is called the "job owner".

4658 If an implementation can be configured to support more than one authentication mechanism (see section  
4659 4.4.2), then it **MUST** implement rules for determining equality of authenticated user names which have  
4660 been authenticated via different authentication mechanisms. One possible policy is that identical names  
4661 that are authenticated via different mechanisms are different. For example, a user can cancel his job only if  
4662 he uses the same authentication mechanism for both Cancel-Job and Print-Job. Another policy is that  
4663 identical names that are authenticated via different mechanism are the same if the authentication  
4664 mechanism for the later operation is not less strong than the authentication mechanism for the earlier job  
4665 creation operation. For example, a user can cancel his job only if he uses the same or stronger  
4666 authentication mechanism for Cancel-Job and Print-Job. With this second policy a job submitted via  
4667 'requesting-user-name' authentication could be canceled via 'digest' authentication. With the first policy, the  
4668 job could not be canceled in this way.

4669 A client is able to determine the authentication mechanism used to create a job. It is the *i*'th value of the  
4670 Printer's "uri-authentication-supported" attribute (see section 4.4.2), where *i* is the index of the element of  
4671 the Printer's "printer-uri-supported" attribute (see section 4.4.1) equal to the job's "job-printer-uri" attribute  
4672 (see section 4.3.3).

## 4673 **8.4 Restricted Queries**

4674 In many IPP operations, a client supplies a list of attributes to be returned in the response. For security  
4675 reasons, an IPP object may be configured not to return all attributes (or all values) that a client requests.  
4676 The job attributes returned **MAY** depend on whether the requesting user is the same as the user that  
4677 submitted the job. The IPP object **MAY** even return none of the requested attributes. In such cases, the  
4678 status returned is the same as if the object had returned all requested attributes. The client cannot tell by  
4679 such a response whether the requested attribute was present or absent on the object.

## 4680 **8.5 Operations performed by operators and system administrators**

4681 For the three printer operations Pause-Printer, Resume-Printer, and Purge-Jobs (see sections 3.2.7, 3.2.8 and  
4682 3.2.9), the requesting user is intended to be an operator or administrator of the Printer object (see section 1).  
4683 Otherwise, the IPP Printer **MUST** reject the operation and return: 'client-error-forbidden', 'client-error-not-  
4684 authenticated', or 'client-error-not-authorized' as appropriate. For operations on jobs, the requesting user is  
4685 intended to be the job owner or may be an operator or administrator of the Printer object. The means for  
4686 authorizing an operator or administrator of the Printer object are not specified in this document.

## 4687 **8.6 Queries on jobs submitted using non-IPP protocols**

4688 If the device that an IPP Printer is representing is able to accept jobs using other job submission protocols  
4689 in addition to IPP, it is **RECOMMENDED** that such an implementation at least allow such "foreign" jobs to  
4690 be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an implementation **NEED**  
4691 **NOT** support all of the same IPP job attributes as for IPP jobs. The IPP object returns the 'unknown' out-of-

4692 band value for any requested attribute of a foreign job that is supported for IPP jobs, but not for foreign  
4693 jobs.

4694 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such "foreign  
4695 jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-Attributes and  
4696 Cancel-Job. Such an implementation also needs to deal with the problem of authentication of such foreign  
4697 jobs. One approach would be to treat all such foreign jobs as belonging to users other than the user of the  
4698 IPP client. Another approach would be for the foreign job to belong to 'anonymous'. Only if the IPP client  
4699 has been authenticated as an operator or administrator of the IPP Printer object, could the foreign jobs be  
4700 queried by an IPP request. Alternatively, if the security policy is to allow users to query other users' jobs,  
4701 then the foreign jobs would also be visible to an end-user IPP client using Get-Jobs and Get-Job-Attributes.

## 4702 9. References

4703 [ASME-Y14.1M]

4704 Metric Drawing Sheet Size and Format, ASME Y14.1M-1995. This standard defines metric sheet  
4705 sizes and formats for engineering drawings.

4706 [ASCII]

4707 Coded Character Set - 7-bit American Standard Code for Information Interchange (ASCII), ANSI  
4708 X3.4-1986. This standard is the specification of the US-ASCII charset.

4709 [BCP-11]

4710 [Bradner S.](#), [Hovey R.](#), "The Organizations Involved in the IETF Standards Process", 1996/10/29  
4711 (RFC 2028)

4712 [HTPP]

4713 J. Barnett, K. Carter, R. DeBry, "Initial Draft - Hypertext Printing Protocol - HTPP/1.0", October  
4714 1996, <ftp://ftp.pwg.org/pub/pwg/ipp/historic/http/overview.ps.gz>

4715 [IANA-CON]

4716 Narte, T. and Alvestrand, H.T.: Guidelines for Writing an IANA Considerations Section in RFCs,  
4717 Work in Progress, draft-iesg-iana-considerations-04.txt, May 21, 1998.

4718 [IANA-CS]

4719 IANA Registry of Coded Character Sets: <ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>

4720 [IANA-MT]

4721 IANA Registry of Media Types: <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/>

4722 [IPP-IIG]

4723 Hastings, T., Manros, C., "Internet Printing Protocol/1.1: draft-ietf-ipp-implementers-guide-v11-  
4724 00.txt, work in progress, September 27, 1999.

- 4725 [IPP-PRO]  
4726 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and  
4727 Transport", draft-ietf-ipp-protocol-v11-05.txt, March 1, 2000.
- 4728 [ISO10646-1]  
4729 ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-Octet Coded Character Set  
4730 (UCS) - Part 1: Architecture and Basic Multilingual Plane, JTC1/SC2."
- 4731 [ISO8859-1]  
4732 ISO/IEC 8859-1:1987, "Information technology -- 8-bit One-Byte Coded Character Set - Part 1:  
4733 Latin Alphabet Nr 1", 1987, JTC1/SC2.
- 4734 [ISO10175]  
4735 ISO/IEC 10175 Document Printing Application (DPA), June 1996.
- 4736 [LDPA]  
4737 T. Hastings, S. Isaacson, M. MacKay, C. Manros, D. Taylor, P. Zehler, "LDPA - Lightweight  
4738 Document Printing Application", October 1996,  
4739 <ftp://ftp.pwg.org/pub/pwg/ipp/historic/ldpa/ldpa8.pdf.gz>
- 4740 [P1387.4]  
4741 Kirk, M. (editor), POSIX System Administration - Part 4: Printing Interfaces, POSIX 1387.4 D8,  
4742 1994.
- 4743 [PSIS] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August 1995.
- 4744 [PWG]  
4745 Printer Working Group, <http://www.pwg.org>.
- 4746 [RFC1035]  
4747 P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", RFC 1035,  
4748 November 1987.
- 4749 [RFC1179]  
4750 McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.
- 4751 [RFC1759]  
4752 Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759, March  
4753 1995.
- 4754 [RFC1766]  
4755 H. Alvestrand, "Tags for the Identification of Languages", RFC 1766, March 1995.
- 4756 [RFC1951]  
4757 P. Deutsch, "DEFLATE Compressed Data Format Specification version 1.3 ", RFC 1951, May  
4758 1996.

- 4759 [RFC1952]  
4760 P. Deutsch, "GZIP file format specification version 4.3", RFC 1952, May 1996.
- 4761 [RFC1977]  
4762 V. Schryver, "PPP BSD Compression Protocol", RFC 1977, August 1996.
- 4763 [RFC2026]  
4764 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 4765 [RFC2045]  
4766 N. Fried, N. Borenstein, ", Multipurpose Internet Mail Extensions (MIME) Part One: Format of  
4767 Internet Message Bodies " RFC 2045, November 1996.
- 4768 [RFC2046]  
4769 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed & N. Borenstein.  
4770 November 1996. (Obsoletes RFC1521, RFC1522, RFC1590), RFC 2046.
- 4771 [RFC2048]  
4772 N. Freed, J. Klensin & J. Postel, "Multipurpose Internet Mail Extension (MIME) Part Four:  
4773 Registration Procedures". RFC 2048, November 1996.
- 4774 [RFC2119]  
4775 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March 1997
- 4776 [RFC2228]  
4777 M. Horowitz, S. Lunt, "FTP Security Extensions", RFC 2228, October 1997.
- 4778 [RFC2246]  
4779 T. Dierks, C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.
- 4780 [RFC2277]  
4781 H. Alvestrand, "IETF Policy on Character Sets and Languages" RFC 2277, January 1998.
- 4782 [RFC2278]  
4783 N. Freed, J. Postel: "IANA CharSet Registration Procedures", RFC 2278, January 1998.
- 4784 [RFC2279]  
4785 F. Yergeau , "UTF-8, a transformation format of ISO 10646", RFC 2279. January 1998.
- 4786 [RFC2316]  
4787 S. Bellovin , "Report of the IAB Security Architecture Workshop", RFC 2316, April 1998.
- 4788 [RFC2396]  
4789 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax",  
4790 RFC 2396, August 1998.

- 4791 [RFC2565]  
4792 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and  
4793 Transport", RFC 2565, April 1999.
- 4794 [RFC2566]  
4795 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and  
4796 Semantics", RFC 2566, April 1999.
- 4797 [RFC2567]  
4798 Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.
- 4799 [RFC2568]  
4800 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",  
4801 RFC 2568, April 1999.
- 4802 [RFC2569]  
4803 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC  
4804 2569, April 1999.
- 4805 [RFC2579]  
4806 K. McCloghrie, D. Perkins, J. Schoenwaelder, "Textual Conventions for SMIV2" RFC 2579 (Also  
4807 STD0058), April 1999.
- 4808 [RFC2616]  
4809 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext  
4810 Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
- 4811 [RFC2617]  
4812 J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, "HTTP  
4813 Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.
- 4814 [SSL]  
4815 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.
- 4816 [SWP]  
4817 P. Moore, B. Jahromi, S. Butler, "Simple Web Printing SWP/1.0", May 7, 1997,  
4818 [ftp://ftp.pwg.org/pub/pwg/ipp/new\\_PRO/swp9705.pdf](ftp://ftp.pwg.org/pub/pwg/ipp/new_PRO/swp9705.pdf)

## 4819 **10. Author's Address**

4820 Scott A. Isaacson (Editor)  
4821 Novell, Inc.  
4822 122 E 1700 S  
4823 Provo, UT 84606

4824  
4825  
4826  
4827  
4828  
4829  
4830  
4831  
4832  
4833  
4834  
4835  
4836  
4837  
4838  
4839  
4840  
4841  
4842  
4843  
4844  
4845  
4846  
4847  
4848  
4849  
4850  
4851  
4852  
4853  
4854  
4855  
4856  
4857  
4858  
4859  
4860  
4861  
4862  
4863  
4864  
4865  
4866

Phone: 801-861-7366  
Fax: 801-861-2517  
e-mail: [sisaacson@novell.com](mailto:sisaacson@novell.com)

Tom Hastings  
Xerox Corporation  
737 Hawaii St. ESAE 231  
El Segundo, CA 90245

Phone: 310-333-6413  
Fax: 310-333-5514  
e-mail: [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)

Robert Herriot  
Xerox Corp.  
3400 Hill View Ave, Building 1  
Palo Alto, CA 94304

Phone: 650-813-7696  
Fax: 650-813-6860  
e-mail: [robert.herriot@pahv.xerox.com](mailto:robert.herriot@pahv.xerox.com)

Roger deBry  
Utah Valley State College  
Orem, UT 84058

Phone: (801) 222-8000  
EMail: [debryro@uvsc.edu](mailto:debryro@uvsc.edu)

Patrick Powell  
Astart Technologies  
9475 Chesapeake Dr., Suite D  
San Diego, CA 95123

Phone: (619) 874-6543  
Fax: (619) 279-8424  
e-mail: [papowell@astart.com](mailto:papowell@astart.com)

IPP Mailing List: [ipp@pwg.org](mailto:ipp@pwg.org)  
IPP Mailing List Subscription: [ipp-request@pwg.org](mailto:ipp-request@pwg.org)  
IPP Web Page: <http://www.pwg.org/ipp/>



4867 Implementers of this specification document are encouraged to join IPP Mailing List in order to participate  
4868 in any discussions of clarification issues and review of registration proposals for additional attributes and  
4869 values.

4870

4871 Other Participants:

Chuck Adams - Tektronix	Shivaun Albright - HP
Stefan Andersson - Axis	Jeff Barnett - IBM
Ron Bergman - Hitachi Koki Imaging Systems	Dennis Carney - IBM
Keith Carter - IBM	Angelo Caruso - Xerox
Rajesh Chawla - TR Computing Solutions	Nancy Chen - Okidata
Josh Cohen - Microsoft	Jeff Copeland - QMS
Andy Davidson - Tektronix	Roger deBry - IBM
Maulik Desai - Auco	Mabry Dozier - QMS
Lee Farrell - Canon Information Systems	Satoshi Fujitami - Ricoh
Steve Gebert - IBM	Sue Gleeson - Digital
Charles Gordon - Osicom	Brian Grimshaw - Apple
Jerry Hadsell - IBM	Richard Hart - Digital
Tom Hastings - Xerox	Henrik Holst - I-data
Stephen Holmstead	Zhi-Hong Huang - Zenographics
Scott Isaacson - Novell	Babek Jahromi - Microsoft
Swen Johnson - Xerox	David Kellerman - Northlake Software
Robert Kline - TrueSpectra	Charles Kong - Panasonic
Carl Kugler - IBM	Dave Kuntz - Hewlett-Packard
Takami Kurono - Brother	Rick Landau - Digital
Scott Lawrence - Agranot Systems	Greg LeClair - Epson
Dwight Lewis - Lexmark	Harry Lewis - IBM
Tony Liao - Vivid Image	Roy Lomicka - Digital
Pete Loya - HP	Ray Lutz - Cognisys
Mike MacKay - Novell, Inc.	David Manchala - Xerox
Carl-Uno Manros - Xerox	Jay Martin - Underscore
Stan McConnell - Xerox	Larry Masinter - Xerox
Sandra Matts - Hewlett Packard	Peter Michalek - Shinesoft
Ira McDonald - High North Inc.	Mike Moldovan - G3 Nova
Tetsuya Morita - Ricoh	Yuichi Niwa - Ricoh
Pat Nogay - IBM	Ron Norton - Printronics
Hugo Parra, Novell	Bob Pentecost - Hewlett-Packard
Patrick Powell - Astart Technologies	Jeff Rackowitz - Intermec
Eric Random - Peerless	Rob Rhoads - Intel
Xavier Riley - Xerox	Gary Roberts - Ricoh
David Roach - Unisys	Stuart Rowley - Kyocera
Yuji Sasaki - Japan Computer Industry	Richard Schneider - Epson
Kris Schoff - HP	Katsuaki Sekiguchi - Canon Information Systems
Bob Setterbo - Adobe	Gail Songer - Peerless
Hideki Tanaka - Cannon Information Systems	Devon Taylor - Novell, Inc.
Mike Timperman - Lexmark	Atsushi Uchino - Epson
Shigeru Ueda - Canon	Bob Von Anandel - Allegro Software
William Wagner - NetSilicon/DPI	Jim Walker - DAZEL
Chris Wellens - Interworking Labs	Trevor Wells - Hewlett Packard
Craig Whittle - Sharp Labs	Rob Whittle - Novell, Inc.
Jasper Wong - Xionics	Don Wright - Lexmark
Michael Wu - Heidelberg Digital	Rick Yardumian - Xerox
Michael Yeung - Canon Information Systems	Lloyd Young - Lexmark
Atsushi Yuki - Kyocera	Peter Zehler - Xerox
William Zhang- Canon Information Systems	Frank Zhao - Panasonic
Steve Zilles - Adobe	Rob Zirnstein - Canon Information Systems

4872

## 4873 **11. Formats for IPP Registration Proposals**

4874 In order to propose an IPP extension for registration, the proposer must submit an application to IANA by  
4875 email to "iana@iana.org" or by filling out the appropriate form on the IANA web pages  
4876 (<http://www.iana.org>). This section specifies the required information and the formats for proposing  
4877 registrations of extensions to IPP as provided in Section 6 for:

- 4878
- 4879 1. type2 'keyword' attribute values
- 4880 2. type3 'keyword' attribute values
- 4881 3. type2 'enum' attribute values
- 4882 4. type3 'enum' attribute values
- 4883 5. attributes
- 4884 6. attribute syntaxes
- 4885 7. operations
- 4886 8. status codes
- 4887 9. out-of-band attribute values

### 4888 **11.1 Type2 keyword attribute values registration**

4889 Type of registration: type2 keyword attribute value  
4890 Name of attribute to which this keyword specification is to be added:  
4891 Proposed keyword name of this keyword value:  
4892 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):  
4893 Name of proposer:  
4894 Address of proposer:  
4895 Email address of proposer:  
4896  
4897 Note: For type2 keywords, the Designated Expert will be the point of contact for the approved registration  
4898 specification, if any maintenance of the registration specification is needed.

### 4899 **11.2 Type3 keyword attribute values registration**

4900 Type of registration: type3 keyword attribute value  
4901 Name of attribute to which this keyword specification is to be added:  
4902 Proposed keyword name of this keyword value:  
4903 Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):  
4904 Name of proposer:  
4905 Address of proposer:  
4906 Email address of proposer:  
4907  
4908 Note: For type3 keywords, the proposer will be the point of contact for the approved registration  
4909 specification, if any maintenance of the registration specification is needed.

### 4910 **11.3 Type2 enum attribute values registration**

4911 Type of registration: type2 enum attribute value

4912 Name of attribute to which this enum specification is to be added:

4913 Keyword symbolic name of this enum value:

4914 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4915 Specification of this enum value (follow the style of IPP Model Section 4.1.4):

4916 Name of proposer:

4917 Address of proposer:

4918 Email address of proposer:

4919

4920 Note: For type2 enums, the Designated Expert will be the point of contact for the approved registration  
4921 specification, if any maintenance of the registration specification is needed.

### 4922 **11.4 Type3 enum attribute values registration**

4923 Type of registration: type3 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 type3 enums, the proposer will be the point of contact for the approved registration specification,  
4933 if any maintenance of the registration specification is needed.

### 4934 **11.5 Attribute registration**

4935 Type of registration: attribute

4936 Proposed keyword name of this attribute:

4937 Types of attribute (Operation, Job Template, Job Description, Printer Description):

4938 Operations to be used with if the attribute is an operation attribute:

4939 Object (Job, Printer, etc. if bound to an object):

4940 Attribute syntax(es) (include 1setOf and range as in Section 4.2):

4941 If attribute syntax is 'keyword' or 'enum', is it type2 or type3:

4942 If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2):

4943 If this is a Job Template attribute, how does its specification depend on the value of the "multiple-  
4944 document-handling" attribute:

4945 Specification of this attribute (follow the style of IPP Model Section 4.2):

4946 Name of proposer:

4947 Address of proposer:

4948 Email address of proposer:

4949

4950 Note: For attributes, the IPP Designated Expert will be the point of contact for the approved registration  
4951 specification, if any maintenance of the registration specification is needed.

## 4952 **11.6 Attribute Syntax registration**

4953 Type of registration: attribute syntax

4954 Proposed name of this attribute syntax:

4955 Type of attribute syntax (integer, octetString, character-string, see [IPP-PRO]):

4956 Numeric tag according to [IPP-PRO] (to be assigned by the IPP Designated Expert in consultation with  
4957 IANA):

4958 Specification of this attribute (follow the style of IPP Model Section 4.1):

4959 Name of proposer:

4960 Address of proposer:

4961 Email address of proposer:

4962

4963 Note: For attribute syntaxes, the IPP Designated Expert will be the point of contact for the approved  
4964 registration specification, if any maintenance of the registration specification is needed.

## 4965 **11.7 Operation registration**

4966 Type of registration: operation

4967 Proposed name of this operation:

4968 Numeric operation-id value according to section 4.4.15 (to be assigned by the IPP Designated Expert in  
4969 consultation with IANA):

4970 Object Target (Job, Printer, etc. that operation is upon):

4971 Specification of this operation (follow the style of IPP Model Section 3):

4972 Name of proposer:

4973 Address of proposer:

4974 Email address of proposer:

4975

4976 Note: For operations, the IPP Designated Expert will be the point of contact for the approved registration  
4977 specification, if any maintenance of the registration specification is needed.

## 4978 **11.8 Attribute Group registration**

4979 Type of registration: attribute group

4980 Proposed name of this attribute group:

4981 Numeric tag according to [IPP-PRO] (to be assigned by the IPP Designated Expert in consultation with  
4982 IANA):

4983 Operation requests and group number for each operation in which the attribute group occurs:

4984 Operation responses and group number for each operation in which the attribute group occurs:

4985 Specification of this attribute group (follow the style of IPP Model Section 3):

4986 Name of proposer:

4987 Address of proposer:

4988 Email address of proposer:

4989

4990 Note: For attribute groups, the IPP Designated Expert will be the point of contact for the approved  
4991 registration specification, if any maintenance of the registration specification is needed.

## 4992 **11.9 Status code registration**

4993 Type of registration: status code

4994 Keyword symbolic name of this status code value:

4995 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4996 Operations that this status code may be used with:

4997 Specification of this status code (follow the style of IPP Model Section 13 APPENDIX B: Status Codes  
4998 and Suggested Status Code Messages):

4999 Name of proposer:

5000 Address of proposer:

5001 Email address of proposer:

5002

5003 Note: For status codes, the Designated Expert will be the point of contact for the approved registration  
5004 specification, if any maintenance of the registration specification is needed.

## 5005 **11.10 Out-of-band Attribute Value registration**

5006 Type of registration: out-of-band attribute value

5007 Proposed name of this out-of-band attribute value:

5008 Numeric tag according to [IPP-PRO] (to be assigned by the IPP Designated Expert in consultation with  
5009 IANA):

5010 Operations that this out-of-band attribute value may be used with:

5011 Attributes that this out-of-band attribute value may be used with:

5012 Specification of this out-of-band attribute value (follow the style of the beginning of IPP Model Section  
5013 4.1):

5014 Name of proposer:

5015 Address of proposer:

5016 Email address of proposer:

5017

5018 Note: For out-of-band attribute values, the IPP Designated Expert will be the point of contact for the  
5019 approved registration specification, if any maintenance of the registration specification is needed.

## 5020 **12. APPENDIX A: Terminology**

5021 This specification document uses the terminology defined in this section.

## 5022 **12.1 Conformance Terminology**

5023 The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT",  
5024 "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in  
5025 RFC 2119 [RFC2119].

### 5026 **12.1.1 NEED NOT**

5027 This term is not included in RFC 2119. The verb "NEED NOT" indicates an action that the subject of the  
5028 sentence does not have to implement in order to claim conformance to the standard. The verb "NEED  
5029 NOT" is used instead of "MAY NOT" since "MAY NOT" sounds like a prohibition.

## 5030 **12.2 Model Terminology**

### 5031 **12.2.1 Keyword**

5032 Keywords are used within this document as identifiers of semantic entities within the abstract model (see  
5033 section 4.1.2.3). Attribute names, some attribute values, attribute syntaxes, and attribute group names are  
5034 represented as keywords.

### 5035 **12.2.2 Attributes**

5036 An attribute is an item of information that is associated with an instance of an IPP object. An attribute  
5037 consists of an attribute name and one or more attribute values. Each attribute has a specific attribute syntax.  
5038 All object attributes are defined in section 4 and all operation attributes are defined in section 3.

5039 Job Template Attributes are described in section 4.2. The client optionally supplies Job Template attributes  
5040 in a create request (operation requests that create Job objects). The Printer object has associated attributes  
5041 which define supported and default values for the Printer.

#### 5042 **12.2.2.1 Attribute Name**

5043 Each attribute is uniquely identified in this document by its attribute name. An attribute name is a keyword.  
5044 The keyword attribute name is given in the section header describing that attribute. In running text in this  
5045 document, attribute names are indicated inside double quotation marks (") where the quotation marks are  
5046 not part of the keyword itself.

#### 5047 **12.2.2.2 Attribute Group Name**

5048 Related attributes are grouped into named groups. The name of the group is a keyword. The group name  
5049 may be used in place of naming all the attributes in the group explicitly. Attribute groups are defined in  
5050 section 3.

### 5051 12.2.2.3 Attribute Value

5052 Each attribute has one or more values. Attribute values are represented in the syntax type specified for that  
5053 attribute. In running text in this document, attribute values are indicated inside single quotation marks ('),  
5054 whether their attribute syntax is keyword, integer, text, etc. where the quotation marks are not part of the  
5055 value itself.

### 5056 12.2.2.4 Attribute Syntax

5057 Each attribute is defined using an explicit syntax type. In this document, each syntax type is defined as a  
5058 keyword with specific meaning. The "Encoding and Transport" document [IPP-PRO] indicates the actual  
5059 "on-the-wire" encoding rules for each syntax type. Attribute syntax types are defined in section 4.1.

### 5060 12.2.3 Supports

5061 By definition, a Printer object supports an attribute only if that Printer object responds with the  
5062 corresponding attribute populated with some value(s) in a response to a query for that attribute. A Printer  
5063 object supports an attribute value if the value is one of the Printer object's "supported values" attributes.  
5064 The device behind a Printer object may exhibit a behavior that corresponds to some IPP attribute, but if the  
5065 Printer object, when queried for that attribute, doesn't respond with the attribute, then as far as IPP is  
5066 concerned, that implementation does not support that feature. If the Printer object's "xxx-supported"  
5067 attribute is not populated with a particular value (even if that value is a legal value for that attribute), then  
5068 that Printer object does not support that particular value.

5069 A conforming implementation MUST support all REQUIRED attributes. However, even for REQUIRED  
5070 attributes, conformance to IPP does not mandate that all implementations support all possible values  
5071 representing all possible job processing behaviors and features. For example, if a given instance of a  
5072 Printer supports only certain document formats, then that Printer responds with the "document-format-  
5073 supported" attribute populated with a set of values, possibly only one, taken from the entire set of possible  
5074 values defined for that attribute. This limited set of values represents the Printer's set of supported  
5075 document formats. Supporting an attribute and some set of values for that attribute enables IPP end users to  
5076 be aware of and make use of those features associated with that attribute and those values. If an  
5077 implementation chooses to not support an attribute or some specific value, then IPP end users would have  
5078 no ability to make use of that feature within the context of IPP itself. However, due to existing practice and  
5079 legacy systems which are not IPP aware, there might be some other mechanism outside the scope of IPP to  
5080 control or request the "unsupported" feature (such as embedded instructions within the document data  
5081 itself).

5082 For example, consider the "finishings-supported" attribute.

- 5083 1) If a Printer object is not physically capable of stapling, the "finishings-supported" attribute MUST  
5084 NOT be populated with the value of 'staple'.
- 5085 2) A Printer object is physically capable of stapling, however an implementation chooses not to support  
5086 stapling in the IPP "finishings" attribute. In this case, 'staple' MUST NOT be a value in the  
5087 "finishings-supported" Printer object attribute. Without support for the value 'staple', an IPP end  
5088 user would have no means within the protocol itself to request that a Job be stapled. However, an



5089 existing document data formatter might be able to request that the document be stapled directly with  
5090 an embedded instruction within the document data. In this case, the IPP implementation does not  
5091 "support" stapling, however the end user is still able to have some control over the stapling of the  
5092 completed job.

5093 3) A Printer object is physically capable of stapling, and an implementation chooses to support stapling  
5094 in the IPP "finishings" attribute. In this case, 'staple' MUST be a value in the "finishings-supported"  
5095 Printer object attribute. Doing so, would enable end users to be aware of and make use of the  
5096 stapling feature using IPP attributes.

5097

5098 Even though support for Job Template attributes by a Printer object is OPTIONAL, it is RECOMMENDED  
5099 that if the device behind a Printer object is capable of realizing any feature or function that corresponds to  
5100 an IPP attribute and some associated value, then that implementation SHOULD support that IPP attribute  
5101 and value.

5102 The set of values in any of the supported value attributes is set (populated) by some administrative process  
5103 or automatic sensing mechanism that is outside the scope of this IPP/1.1 document. For administrative  
5104 policy and control reasons, an administrator may choose to make only a subset of possible values visible to  
5105 the end user. In this case, the real output device behind the IPP Printer abstraction may be capable of a  
5106 certain feature, however an administrator is specifying that access to that feature not be exposed to the end  
5107 user through the IPP protocol. Also, since a Printer object may represent a logical print device (not just a  
5108 physical device) the actual process for supporting a value is undefined and left up to the implementation.  
5109 However, if a Printer object supports a value, some manual human action may be needed to realize the  
5110 semantic action associated with the value, but no end user action is required.

5111 For example, if one of the values in the "finishings-supported" attribute is 'staple', the actual process might  
5112 be an automatic staple action by a physical device controlled by some command sent to the device. Or, the  
5113 actual process of stapling might be a manual action by an operator at an operator attended Printer object.

5114 For another example of how supported attributes function, consider a system administrator who desires to  
5115 control all print jobs so that no job sheets are printed in order to conserve paper. To force no job sheets, the  
5116 system administrator sets the only supported value for the "job-sheets-supported" attribute to 'none'. In this  
5117 case, if a client requests anything except 'none', the create request is rejected or the "job-sheets" value is  
5118 ignored (depending on the value of "ipp-attribute-fidelity"). To force the use of job start/end sheets on all  
5119 jobs, the administrator does not include the value 'none' in the "job-sheets-supported" attribute. In this case,  
5120 if a client requests 'none', the create request is rejected or the "job-sheets" value is ignored (again depending  
5121 on the value of "ipp-attribute-fidelity").

#### 5122 **12.2.4 print-stream page**

5123 A "print-stream page" is a page according to the definition of pages in the language used to express the  
5124 document data.

5125 **12.2.5 impression**

5126 An "impression" is the image (possibly many print-stream pages in different configurations) imposed onto a  
5127 single media page.

5128 **13. APPENDIX B: Status Codes and Suggested Status Code Messages**

5129 This section defines status code enum keywords and values that are used to provide semantic information  
5130 on the results of an operation request. Each operation response **MUST** include a status code. The response  
5131 **MAY** also contain a status message that provides a short textual description of the status. The status code  
5132 is intended for use by automata, and the status message is intended for the human end user. Since the status  
5133 message is an **OPTIONAL** component of the operation response, an IPP application (i.e., a browser, GUI,  
5134 print driver or gateway) is **NOT REQUIRED** to examine or display the status message, since it **MAY** not be  
5135 returned to the application.

5136 The prefix of the status keyword defines the class of response as follows:

- 5137 "informational" - Request received, continuing process
- 5138 "successful" - The action was successfully received, understood, and accepted
- 5139 "redirection" - Further action must be taken in order to complete the request
- 5140 "client-error" - The request contains bad syntax or cannot be fulfilled
- 5141 "server-error" - The IPP object failed to fulfill an apparently valid request
- 5142

5143 As with type2 enums, IPP status codes are extensible. IPP clients are **NOT REQUIRED** to understand the  
5144 meaning of all registered status codes, though such understanding is obviously desirable. However, IPP  
5145 clients **MUST** understand the class of any status code, as indicated by the prefix, and treat any unrecognized  
5146 response as being equivalent to the first status code of that class, with the exception that an unrecognized  
5147 response **MUST NOT** be cached. For example, if an unrecognized status code of "client-error-xxx-yyy" is  
5148 received by the client, it can safely assume that there was something wrong with its request and treat the  
5149 response as if it had received a "client-error-bad-request" status code. In such cases, IPP applications  
5150 **SHOULD** present the **OPTIONAL** message (if present) to the end user since the message is likely to  
5151 contain human readable information which will help to explain the unusual status. The name of the enum  
5152 is the suggested status message for US English.

5153 The status code values range from 0x0000 to 0x7FFF. The value ranges for each status code class are as  
5154 follows:

- 5155 "successful" - 0x0000 to 0x00FF
- 5156 "informational" - 0x0100 to 0x01FF
- 5157 "redirection" - 0x0200 to 0x02FF
- 5158 "client-error" - 0x0400 to 0x04FF
- 5159 "server-error" - 0x0500 to 0x05FF
- 5160

5161 The top half (128 values) of each range (0x0n40 to 0x0nFF, for n = 0 to 5) is reserved for vendor use within  
5162 each status code class. Values 0x0600 to 0x7FFF are reserved for future assignment by IETF standards  
5163 track documents and MUST NOT be used.

## 5164 **13.1 Status Codes**

5165 Each status code is described below. Section 13.1.5.9 contains a table that indicates which status codes  
5166 apply to which operations. The Implementer's Guide [IPP-IIG] describe the suggested steps for processing  
5167 IPP attributes for all operations, including returning status codes.

### 5168 **13.1.1 Informational**

5169 This class of status code indicates a provisional response and is to be used for informational purposes only.

5170 There are no status codes defined in IPP/1.1 for this class of status code.

### 5171 **13.1.2 Successful Status Codes**

5172 This class of status code indicates that the client's request was successfully received, understood, and  
5173 accepted.

#### 5174 **13.1.2.1 successful-ok (0x0000)**

5175 The request has succeeded and no request attributes were substituted or ignored. In the case of a response  
5176 to a create request, the 'successful-ok' status code indicates that the request was successfully received and  
5177 validated, and that the Job object has been created; it does not indicate that the job has been processed. The  
5178 transition of the Job object into the 'completed' state is the only indicator that the job has been printed.

#### 5179 **13.1.2.2 successful-ok-ignored-or-substituted-attributes (0x0001)**

5180 The request has succeeded, but some supplied (1) attributes were ignored or (2) unsupported values were  
5181 substituted with supported values or were ignored in order to perform the operation without rejecting it.  
5182 Unsupported attributes, attribute syntaxes, or values MUST be returned in the Unsupported Attributes  
5183 group of the response for all operations. There is an exception to this rule for the query operations: Get-  
5184 Printer-Attributes, Get-Jobs, and Get-Job-Attributes for the "requested-attributes" operation attribute only.  
5185 When the supplied values of the "requested-attributes" operation attribute are requesting attributes that are  
5186 not supported, the IPP object MAY, but is NOT REQUIRED to, return the "requested-attributes" attribute  
5187 in the Unsupported Attribute response group (with the unsupported values only). See sections 3.1.7 and  
5188 3.2.1.2.

#### 5189 **13.1.2.3 successful-ok-conflicting-attributes (0x0002)**

5190 The request has succeeded, but some supplied attribute values conflicted with the values of other supplied  
5191 attributes. These conflicting values were either (1) substituted with (supported) values or (2) the attributes

5192 were removed in order to process the job without rejecting it. Attributes or values which conflict with other  
5193 attributes and have been substituted or ignored MUST be returned in the Unsupported Attributes group of  
5194 the response for all operations as supplied by the client. See sections 3.1.7 and 3.2.1.2.

### 5195 **13.1.3 Redirection Status Codes**

5196 This class of status code indicates that further action needs to be taken to fulfill the request.

5197 There are no status codes defined in IPP/1.1 for this class of status code.

### 5198 **13.1.4 Client Error Status Codes**

5199 This class of status code is intended for cases in which the client seems to have erred. The IPP object  
5200 SHOULD return a message containing an explanation of the error situation and whether it is a temporary or  
5201 permanent condition.

#### 5202 **13.1.4.1 client-error-bad-request (0x0400)**

5203 The request could not be understood by the IPP object due to malformed syntax (such as the value of a  
5204 fixed length attribute whose length does not match the prescribed length for that attribute - see the  
5205 Implementer's Guide [IPP-IIG] ). The IPP application SHOULD NOT repeat the request without  
5206 modifications.

#### 5207 **13.1.4.2 client-error-forbidden (0x0401)**

5208 The IPP object understood the request, but is refusing to fulfill it. Additional authentication information or  
5209 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is  
5210 commonly used when the IPP object does not wish to reveal exactly why the request has been refused or  
5211 when no other response is applicable.

#### 5212 **13.1.4.3 client-error-not-authenticated (0x0402)**

5213 The request requires user authentication. The IPP client may repeat the request with suitable authentication  
5214 information. If the request already included authentication information, then this status code indicates that  
5215 authorization has been refused for those credentials. If this response contains the same challenge as the  
5216 prior response, and the user agent has already attempted authentication at least once, then the response  
5217 message may contain relevant diagnostic information. This status codes reveals more information than  
5218 "client-error-forbidden".

#### 5219 **13.1.4.4 client-error-not-authorized (0x0403)**

5220 The requester is not authorized to perform the request. Additional authentication information or  
5221 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is used  
5222 when the IPP object wishes to reveal that the authentication information is understandable, however, the

5223 requester is explicitly not authorized to perform the request. This status codes reveals more information  
5224 than "client-error-forbidden" and "client-error-not-authenticated".

#### 5225 **13.1.4.5 client-error-not-possible (0x0404)**

5226 This status code is used when the request is for something that can not happen. For example, there might  
5227 be a request to cancel a job that has already been canceled or aborted by the system. The IPP client  
5228 SHOULD NOT repeat the request.

#### 5229 **13.1.4.6 client-error-timeout (0x0405)**

5230 The client did not produce a request within the time that the IPP object was prepared to wait. For example,  
5231 a client issued a Create-Job operation and then, after a long period of time, issued a Send-Document  
5232 operation and this error status code was returned in response to the Send-Document request (see section  
5233 3.3.1). The IPP object might have been forced to clean up resources that had been held for the waiting  
5234 additional Documents. The IPP object was forced to close the Job since the client took too long. The client  
5235 SHOULD NOT repeat the request without modifications.

#### 5236 **13.1.4.7 client-error-not-found (0x0406)**

5237 The IPP object has not found anything matching the request URI. No indication is given of whether the  
5238 condition is temporary or permanent. For example, a client with an old reference to a Job (a URI) tries to  
5239 cancel the Job, however in the mean time the Job might have been completed and all record of it at the  
5240 Printer has been deleted. This status code, 'client-error-not-found' is returned indicating that the referenced  
5241 Job can not be found. This error status code is also used when a client supplies a URI as a reference to the  
5242 document data in either a Print-URI or Send-URI operation, but the document can not be found.

5243 In practice, an IPP application should avoid a not found situation by first querying and presenting a list of  
5244 valid Printer URIs and Job URIs to the end-user.

#### 5245 **13.1.4.8 client-error-gone (0x0407)**

5246 The requested object is no longer available and no forwarding address is known. This condition should be  
5247 considered permanent. Clients with link editing capabilities should delete references to the request URI  
5248 after user approval. If the IPP object does not know or has no facility to determine, whether or not the  
5249 condition is permanent, the status code "client-error-not-found" should be used instead.

5250 This response is primarily intended to assist the task of maintenance by notifying the recipient that the  
5251 resource is intentionally unavailable and that the IPP object administrator desires that remote links to that  
5252 resource be removed. It is not necessary to mark all permanently unavailable resources as "gone" or to keep  
5253 the mark for any length of time -- that is left to the discretion of the IPP object administrator and/or Printer  
5254 implementation.

**5255 13.1.4.9 client-error-request-entity-too-large (0x0408)**

5256 The IPP object is refusing to process a request because the request entity is larger than the IPP object is  
5257 willing or able to process. An IPP Printer returns this status code when it limits the size of print jobs and it  
5258 receives a print job that exceeds that limit or when the attributes are so many that their encoding causes the  
5259 request entity to exceed IPP object capacity.

**5260 13.1.4.10 client-error-request-value-too-long (0x0409)**

5261 The IPP object is refusing to service the request because one or more of the client-supplied attributes has a  
5262 variable length value that is longer than the maximum length specified for that attribute. The IPP object  
5263 might not have sufficient resources (memory, buffers, etc.) to process (even temporarily), interpret, and/or  
5264 ignore a value larger than the maximum length. Another use of this error code is when the IPP object  
5265 supports the processing of a large value that is less than the maximum length, but during the processing of  
5266 the request as a whole, the object may pass the value onto some other system component which is not able  
5267 to accept the large value. For more details, see the Implementer's Guide [IPP-IIG] .

5268 Note: For attribute values that are URIs, this rare condition is only likely to occur when a client has  
5269 improperly submitted a request with long query information (e.g. an IPP application allows an end-user to  
5270 enter an invalid URI), when the client has descended into a URI "black hole" of redirection (e.g., a  
5271 redirected URI prefix that points to a suffix of itself), or when the IPP object is under attack by a client  
5272 attempting to exploit security holes present in some IPP objects using fixed-length buffers for reading or  
5273 manipulating the Request-URI.

**5274 13.1.4.11 client-error-document-format-not-supported (0x040A)**

5275 The IPP object is refusing to service the request because the document data is in a format, as specified in  
5276 the "document-format" operation attribute, that is not supported by the Printer object. This error is returned  
5277 independent of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return this status code,  
5278 even if there are other Job Template attributes that are not supported as well, since this error is a bigger  
5279 problem than with Job Template attributes. See sections 3.1.6.1, 3.1.7, and 3.2.1.1.

**5280 13.1.4.12 client-error-attributes-or-values-not-supported (0x040B)**

5281 In a create request, if the Printer object does not support one or more attributes, attribute syntaxes, or  
5282 attribute values supplied in the request and the client supplied the "ipp-attribute-fidelity" operation attribute  
5283 with the 'true' value, the Printer object MUST return this status code. The Printer object MUST also return  
5284 in the Unsupported Attributes Group all the attributes and/or values supplied by the client that are not  
5285 supported. See section 3.1.7. For example, if the request indicates 'iso-a4' media, but that media type is not  
5286 supported by the Printer object. Or, if the client supplies a Job Template attribute and the attribute itself is  
5287 not even supported by the Printer. If the "ipp-attribute-fidelity" attribute is 'false', the Printer MUST ignore  
5288 or substitute values for unsupported Job Template attributes and values rather than reject the request and  
5289 return this status code.

5290 For any operation where a client requests attributes (such as a Get-Jobs, Get-Printer-Attributes, or Get-Job-  
5291 Attributes operation), if the IPP object does not support one or more of the requested attributes, the IPP  
5292 object simply ignores the unsupported requested attributes and processes the request as if they had not been  
5293 supplied, rather than returning this status code. In this case, the IPP object MUST return the 'successful-ok-  
5294 ignored-or-substituted-attributes' status code and MAY return the unsupported attributes as values of the  
5295 "requested-attributes" in the Unsupported Attributes Group (see section 13.1.2.2).

#### 5296 **13.1.4.13 client-error-uri-scheme-not-supported (0x040C)**

5297 The scheme of the client-supplied URI in a Print-URI or a Send-URI operation is not supported. See  
5298 sections 3.1.6.1 and 3.1.7.

#### 5299 **13.1.4.14 client-error-charset-not-supported (0x040D)**

5300 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-  
5301 charset" operation attribute, the Printer MUST reject the operation and return this status and any 'text' or  
5302 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1). See sections 3.1.6.1 and 3.1.7.

#### 5303 **13.1.4.15 client-error-conflicting-attributes (0x040E)**

5304 The request is rejected because some attribute values conflicted with the values of other attributes which  
5305 this document does not permit to be substituted or ignored. The Printer object MUST also return in the  
5306 Unsupported Attributes Group the conflicting attributes supplied by the client. See sections 3.1.7 and  
5307 3.2.1.2.

#### 5308 **13.1.4.16 client-error-compression-not-supported (0x040F)**

5309 The IPP object is refusing to service the request because the document data, as specified in the  
5310 "compression" operation attribute, is compressed in a way that is not supported by the Printer object. This  
5311 error is returned independent of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return  
5312 this status code, even if there are other Job Template attributes that are not supported as well, since this  
5313 error is a bigger problem than with Job Template attributes. See sections 3.1.6.1, 3.1.7, and 3.2.1.1.

#### 5314 **13.1.4.17 client-error-compression-error (0x0410)**

5315 The IPP object is refusing to service the request because the document data cannot be decompressed when  
5316 using the algorithm specified by the "compression" operation attribute. This error is returned independent  
5317 of the client-supplied "ipp-attribute-fidelity". The Printer object MUST return this status code, even if there  
5318 are Job Template attributes that are not supported as well, since this error is a bigger problem than with Job  
5319 Template attributes. See sections 3.1.7 and 3.2.1.1.

**5320 13.1.4.18 client-error-document-format-error (0x0411)**

5321 The IPP object is refusing to service the request because Printer encountered an error in the document data  
5322 while interpreting it. This error is returned independent of the client-supplied "ipp-attribute-fidelity". The  
5323 Printer object **MUST** return this status code, even if there are Job Template attributes that are not supported  
5324 as well, since this error is a bigger problem than with Job Template attributes. See sections 3.1.7 and  
5325 3.2.1.1.

**5326 13.1.4.19 client-error-document-access-error (0x0412)**

5327 The IPP object is refusing to service the Print-URI or Send-URI request because Printer encountered an  
5328 access error while attempting to validate the accessibility or access the document data specified in the  
5329 "document-uri" operation attribute. The Printer **MAY** also return a specific document access error code  
5330 using the "document-access-error" operation attribute (see section 3.1.6.4). This error is returned  
5331 independent of the client-supplied "ipp-attribute-fidelity". The Printer object **MUST** return this status code,  
5332 even if there are Job Template attributes that are not supported as well, since this error is a bigger problem  
5333 than with Job Template attributes. See sections 3.1.6.1 and 3.1.7.

**5334 13.1.5 Server Error Status Codes**

5335 This class of status codes indicates cases in which the IPP object is aware that it has erred or is incapable of  
5336 performing the request. The IPP object **SHOULD** include a message containing an explanation of the error  
5337 situation, and whether it is a temporary or permanent condition.

**5338 13.1.5.1 server-error-internal-error (0x0500)**

5339 The IPP object encountered an unexpected condition that prevented it from fulfilling the request. This error  
5340 status code differs from "server-error-temporary-error" in that it implies a more permanent type of internal  
5341 error. It also differs from "server-error-device-error" in that it implies an unexpected condition (unlike a  
5342 paper-jam or out-of-toner problem which is undesirable but expected). This error status code indicates that  
5343 probably some knowledgeable human intervention is required.

**5344 13.1.5.2 server-error-operation-not-supported (0x0501)**

5345 The IPP object does not support the functionality required to fulfill the request. This is the appropriate  
5346 response when the IPP object does not recognize an operation or is not capable of supporting it. See  
5347 sections 3.1.6.1 and 3.1.7.

**5348 13.1.5.3 server-error-service-unavailable (0x0502)**

5349 The IPP object is currently unable to handle the request due to a temporary overloading or maintenance of  
5350 the IPP object. The implication is that this is a temporary condition which will be alleviated after some  
5351 delay. If known, the length of the delay may be indicated in the message. If no delay is given, the IPP  
5352 application should handle the response as it would for a "server-error-temporary-error" response. If the



5353 condition is more permanent, the error status codes "client-error-gone" or "client-error-not-found" could be  
5354 used.

#### 5355 **13.1.5.4 server-error-version-not-supported (0x0503)**

5356 The IPP object does not support, or refuses to support, the IPP protocol version that was supplied as the  
5357 value of the "version-number" operation parameter in the request. The IPP object is indicating that it is  
5358 unable or unwilling to complete the request using the same major and minor version number as supplied in  
5359 the request other than with this error message. The error response SHOULD contain a "status-message"  
5360 attribute (see section 3.1.6.2) describing why that version is not supported and what other versions are  
5361 supported by that IPP object. See sections 3.1.6.1, 3.1.7, and 3.1.8.

5362 The error response MUST identify in the "version-number" operation parameter the closest version number  
5363 that the IPP object does support. For example, if a client supplies version '1.0' and an IPP/1.1 object  
5364 supports version '1.0', then it responds with version '1.0' in all responses to such a request. If the IPP/1.1  
5365 object does not support version '1.0', then it should accept the request and respond with version '1.1' or may  
5366 reject the request and respond with this error code and version '1.1'. If a client supplies a version '1.2', the  
5367 IPP/1.1 object should accept the request and return version '1.1' or may reject the request and respond with  
5368 this error code and version '1.1'. See sections 3.1.8 and 4.4.14.

#### 5369 **13.1.5.5 server-error-device-error (0x0504)**

5370 A printer error, such as a paper jam, occurs while the IPP object processes a Print or Send operation. The  
5371 response contains the true Job Status (the values of the "job-state" and "job-state-reasons" attributes).  
5372 Additional information can be returned in the OPTIONAL "job-state-message" attribute value or in the  
5373 OPTIONAL status message that describes the error in more detail. This error status code is only returned in  
5374 situations where the Printer is unable to accept the create request because of such a device error. For  
5375 example, if the Printer is unable to spool, and can only accept one job at a time, the reason it might reject a  
5376 create request is that the printer currently has a paper jam. In many cases however, where the Printer object  
5377 can accept the request even though the Printer has some error condition, the 'successful-ok' status code will  
5378 be returned. In such a case, the client would look at the returned Job Object Attributes or later query the  
5379 Printer to determine its state and state reasons.

#### 5380 **13.1.5.6 server-error-temporary-error (0x0505)**

5381 A temporary error such as a buffer full write error, a memory overflow (i.e. the document data exceeds the  
5382 memory of the Printer), or a disk full condition, occurs while the IPP Printer processes an operation. The  
5383 client MAY try the unmodified request again at some later point in time with an expectation that the  
5384 temporary internal error condition may have been cleared. Alternatively, as an implementation option, a  
5385 Printer object MAY delay the response until the temporary condition is cleared so that no error is returned.

5386 **13.1.5.7 server-error-not-accepting-jobs (0x0506)**

5387 A temporary error indicating that the Printer is not currently accepting jobs, because the administrator has  
5388 set the value of the Printer's "printer-is-accepting-jobs" attribute to 'false' (by means outside the scope of  
5389 this IPP/1.1 document).

5390 **13.1.5.8 server-error-busy (0x0507)**

5391 A temporary error indicating that the Printer is too busy processing jobs and/or other requests. The client  
5392 SHOULD try the unmodified request again at some later point in time with an expectation that the  
5393 temporary busy condition will have been cleared.

5394 **13.1.5.9 server-error-job-canceled (0x0508)**

5395 An error indicating that the job has been canceled by an operator or the system while the client was  
5396 transmitting the data to the IPP Printer. If a job-id and job-uri had been created, then they are returned in  
5397 the Print-Job, Send-Document, or Send-URI response as usual; otherwise, no job-id and job-uri are returned  
5398 in the response.

5399 **13.1.5.10 server-error-multiple-document-jobs-not-supported (0x0509)**

5400 The IPP object does not support multiple documents per job and a client attempted to supply document data  
5401 with a second Send-Document or Send-URI operation.

5402 **13.2 Status Codes for IPP Operations**

5403 PJ = Print-Job, PU = Print-URI, CJ = Create-Job, SD = Send-Document  
 5404 SU = Send-URI, V = Validate-Job, GA = Get-Job-Attributes and  
 5405 Get-Printer-Attributes, GJ = Get-Jobs, C = Cancel-Job

5406

5407

5408	IPP Status Keyword	PJ	PU	CJ	SD	SU	V	GA	GJ	C
5409	-----	--	--	--	--	--	-	--	--	-
5410	successful-ok	x	x	x	x	x	x	x	x	x
5411	successful-ok-ignored-or-substituted-	x	x	x	x	x	x	x	x	x
5412	attributes									
5413	successful-ok-conflicting-attributes	x	x	x	x	x	x	x	x	x
5414	client-error-bad-request	x	x	x	x	x	x	x	x	x
5415	client-error-forbidden	x	x	x	x	x	x	x	x	x
5416	client-error-not-authenticated	x	x	x	x	x	x	x	x	x
5417	client-error-not-authorized	x	x	x	x	x	x	x	x	x
5418	client-error-not-possible	x	x	x	x	x	x	x	x	x
5419	client-error-timeout				x	x				
5420	client-error-not-found	x	x	x	x	x	x	x	x	x
5421	client-error-gone	x	x	x	x	x	x	x	x	x
5422	client-error-request-entity-too-large	x	x	x	x	x	x	x	x	x
5423	client-error-request-value-too-long	x	x	x	x	x	x	x	x	x
5424	client-error-document-format-not-	x	x		x	x	x	x		
5425	supported									
5426	client-error-attributes-or-values-not-	x	x	x	x	x	x	x	x	x
5427	supported									
5428	client-error-uri-scheme-not-supported		x			x				
5429	client-error-charset-not-supported	x	x	x	x	x	x	x	x	x
5430	client-error-conflicting-attributes	x	x	x	x	x	x	x	x	x
5431	client-error-compression-not-supported	x	x		x	x	x			
5432	client-error-compression-error	x	x		x	x				
5433	client-error-document-format-error	x	x		x	x				
5434	client-error-document-access-error		x			x				
5435	server-error-internal-error	x	x	x	x	x	x	x	x	x
5436	server-error-operation-not-supported		x	x	x	x				
5437	server-error-service-unavailable	x	x	x	x	x	x	x	x	x
5438	server-error-version-not-supported	x	x	x	x	x	x	x	x	x
5439	server-error-device-error	x	x	x	x	x				
5440	server-error-temporary-error	x	x	x	x	x				
5441	server-error-not-accepting-jobs	x	x	x			x			
5442	server-error-busy	x	x	x	x	x	x	x	x	x
5443	server-error-job-canceled	x			x	x				
5444	server-error-multiple-document-jobs-				x	x				
5445	not-supported									

5446 HJ = Hold-Job, RJ = Release-Job, RS = Restart-Job  
 5447 PP = Pause-Printer, RP = Resume-Printer, PJ = Purge-Jobs

5448		IPP Operations (cont.)					
5449	IPP Status Keyword	HJ	RJ	RS	PP	RP	PJ
5450	-----	--	--	--	--	--	--
5451	successful-ok	x	x	x	x	x	x
5452	successful-ok-ignored-or-substituted-	x	x	x	x	x	x
5453	attributes						
5454	successful-ok-conflicting-attributes	x	x	x	x	x	x
5455	client-error-bad-request	x	x	x	x	x	x
5456	client-error-forbidden	x	x	x	x	x	x
5457	client-error-not-authenticated	x	x	x	x	x	x
5458	client-error-not-authorized	x	x	x	x	x	x
5459	client-error-not-possible	x	x	x	x	x	x
5460	client-error-timeout						
5461	client-error-not-found	x	x	x	x	x	x
5462	client-error-gone	x	x	x	x	x	x
5463	client-error-request-entity-too-large	x	x	x	x	x	x
5464	client-error-request-value-too-long	x	x	x	x	x	x
5465	client-error-document-format-not-						
5466	supported						
5467	client-error-attributes-or-values-not-	x	x	x	x	x	x
5468	supported						
5469	client-error-uri-scheme-not-supported						
5470	client-error-charset-not-supported	x	x	x	x	x	x
5471	client-error-conflicting-attributes	x	x	x	x	x	x
5472	client-error-compression-not-supported						
5473	client-error-compression-error						
5474	client-error-document-format-error						
5475	client-error-document-access-error						
5476	server-error-internal-error	x	x	x	x	x	x
5477	server-error-operation-not-supported	x	x	x	x	x	x
5478	server-error-service-unavailable	x	x	x	x	x	x
5479	server-error-version-not-supported	x	x	x	x	x	x
5480	server-error-device-error						
5481	server-error-temporary-error	x	x	x	x	x	x
5482	server-error-not-accepting-jobs						
5483	server-error-busy	x	x	x	x	x	x
5484	server-error-job-canceled						
5485	server-error-multiple-document-jobs-						
5486	not-supported						
5487							
5488							

5489

5490 **14. APPENDIX C: "media" keyword values**5491 **14. APPENDIX C: "media" keyword values**

5492 Standard keyword values are taken from several sources.

5493 Standard values are defined (taken from DPA[ISO10175] and the Printer MIB[RFC1759]):

5494 'default': The default medium for the output device  
5495 'iso-a4-white': Specifies the ISO A4 white medium: 210 mm x 297 mm  
5496 'iso-a4-colored': Specifies the ISO A4 colored medium: 210 mm x 297 mm  
5497 'iso-a4-transparent' Specifies the ISO A4 transparent medium: 210 mm x 297 mm  
5498 'iso-a3-white': Specifies the ISO A3 white medium: 297 mm x 420 mm  
5499 'iso-a3-colored': Specifies the ISO A3 colored medium: 297 mm x 420 mm  
5500 'iso-a5-white': Specifies the ISO A5 white medium: 148 mm x 210 mm  
5501 'iso-a5-colored': Specifies the ISO A5 colored medium: 148 mm x 210 mm  
5502 'iso-b4-white': Specifies the ISO B4 white medium: 250 mm x 353 mm  
5503 'iso-b4-colored': Specifies the ISO B4 colored medium: 250 mm x 353 mm  
5504 'iso-b5-white': Specifies the ISO B5 white medium: 176 mm x 250 mm  
5505 'iso-b5-colored': Specifies the ISO B5 colored medium: 176 mm x 250 mm  
5506 'jis-b4-white': Specifies the JIS B4 white medium: 257 mm x 364 mm  
5507 'jis-b4-colored': Specifies the JIS B4 colored medium: 257 mm x 364 mm  
5508 'jis-b5-white': Specifies the JIS B5 white medium: 182 mm x 257 mm  
5509 'jis-b5-colored': Specifies the JIS B5 colored medium: 182 mm x 257 mm  
5510

5511 The following standard values are defined for North American media:

5512 'na-letter-white': Specifies the North American letter white medium  
5513 'na-letter-colored': Specifies the North American letter colored medium  
5514 'na-letter-transparent': Specifies the North American letter transparent medium  
5515 'na-legal-white': Specifies the North American legal white medium  
5516 'na-legal-colored': Specifies the North American legal colored medium  
5517

5518 The following standard values are defined for envelopes:

5519 'iso-b4-envelope': Specifies the ISO B4 envelope medium  
5520 'iso-b5-envelope': Specifies the ISO B5 envelope medium  
5521 'iso-c3-envelope': Specifies the ISO C3 envelope medium  
5522 'iso-c4-envelope': Specifies the ISO C4 envelope medium  
5523 'iso-c5-envelope': Specifies the ISO C5 envelope medium  
5524 'iso-c6-envelope': Specifies the ISO C6 envelope medium  
5525 'iso-designated-long-envelope': Specifies the ISO Designated Long envelope medium  
5526 'na-10x13-envelope': Specifies the North American 10x13 envelope medium

5527 'na-9x12-envelope': Specifies the North American 9x12 envelope medium  
5528 'monarch-envelope': Specifies the Monarch envelope  
5529 'na-number-10-envelope': Specifies the North American number 10 business envelope medium  
5530 'na-7x9-envelope': Specifies the North American 7x9 inch envelope  
5531 'na-9x11-envelope': Specifies the North American 9x11 inch envelope  
5532 'na-10x14-envelope': Specifies the North American 10x14 inch envelope  
5533 'na-number-9-envelope': Specifies the North American number 9 business envelope  
5534 'na-6x9-envelope': Specifies the North American 6x9 inch envelope  
5535 'na-10x15-envelope': Specifies the North American 10x15 inch envelope  
5536

5537 The following standard values are defined for the less commonly used media-:

5538 'executive-white': Specifies the white executive medium  
5539 'folio-white': Specifies the folio white medium  
5540 'invoice-white': Specifies the white invoice medium  
5541 'ledger-white': Specifies the white ledger medium  
5542 'quarto-white': Specified the white quarto medium  
5543 'iso-a0-white': Specifies the ISO A0 white medium: 841 mm x 1189 mm  
5544 'iso-a0-transparent': Specifies the ISO A0 transparent medium: 841 mm x 1189 mm  
5545 'iso-a0-translucent': Specifies the ISO A0 translucent medium: 841 mm x 1189 mm  
5546 'iso-a1-white': Specifies the ISO A1 white medium: 594 mm x 841 mm  
5547 'iso-a1-transparent': Specifies the ISO A1 transparent medium: 594 mm x 841 mm  
5548 'iso-a1-translucent': Specifies the ISO A1 translucent medium: 594 mm x 841 mm  
5549 'iso-a2-white': Specifies the ISO A2 white medium: 420 mm x 594 mm  
5550 'iso-a2-transparent': Specifies the ISO A2 transparent medium: 420 mm x 594 mm  
5551 'iso-a2-translucent': Specifies the ISO A2 translucent medium: 420 mm x 594 mm  
5552 'iso-a3-transparent': Specifies the ISO A3 transparent medium: 297 mm x 420 mm  
5553 'iso-a3-translucent': Specifies the ISO A3 translucent medium: 297 mm x 420 mm  
5554 'iso-a4-translucent': Specifies the ISO A4 translucent medium: 210 mm x 297 mm  
5555 'iso-a5-transparent': Specifies the ISO A5 transparent medium: 148 mm x 210 mm  
5556 'iso-a5-translucent': Specifies the ISO A5 translucent medium: 148 mm x 210 mm  
5557 'iso-a6-white': Specifies the ISO A6 white medium: 105 mm x 148 mm  
5558 'iso-a7-white': Specifies the ISO A7 white medium: 74 mm x 105 mm  
5559 'iso-a8-white': Specifies the ISO A8 white medium: 52 mm x 74 mm  
5560 'iso-a9-white': Specifies the ISO A9 white medium: 37 mm x 52 mm  
5561 'iso-10-white': Specifies the ISO A10 white medium: 26 mm x 37 mm  
5562 'iso-b0-white': Specifies the ISO B0 white medium: 1000 mm x 1414 mm  
5563 'iso-b1-white': Specifies the ISO B1 white medium: 707 mm x 1000 mm  
5564 'iso-b2-white': Specifies the ISO B2 white medium: 500 mm x 707 mm  
5565 'iso-b3-white': Specifies the ISO B3 white medium: 353 mm x 500 mm  
5566 'iso-b6-white': Specifies the ISO B6 white medium: 125 mm x 176 mm  
5567 'iso-b7-white': Specifies the ISO B7 white medium: 88 mm x 125 mm  
5568 'iso-b8-white': Specifies the ISO B8 white medium: 62 mm x 88 mm  
5569 'iso-b9-white': Specifies the ISO B9 white medium: 44 mm x 62 mm  
5570 'iso-b10-white': Specifies the ISO B10 white medium: 31 mm x 44 mm

5571 'jis-b0-white': Specifies the JIS B0 white medium: 1030 mm x 1456 mm  
5572 'jis-b0-transparent': Specifies the JIS B0 transparent medium: 1030 mm x 1456 mm  
5573 'jis-b0-translucent': Specifies the JIS B0 translucent medium: 1030 mm x 1456 mm  
5574 'jis-b1-white': Specifies the JIS B1 white medium: 728 mm x 1030 mm  
5575 'jis-b1-transparent': Specifies the JIS B1 transparent medium: 728 mm x 1030 mm  
5576 'jis-b1-translucent': Specifies the JIS B1 translucent medium: 728 mm x 1030 mm  
5577 'jis-b2-white': Specifies the JIS B2 white medium: 515 mm x 728 mm  
5578 'jis-b2-transparent': Specifies the JIS B2 transparent medium: 515 mm x 728 mm  
5579 'jis-b2-translucent': Specifies the JIS B2 translucent medium: 515 mm x 728 mm  
5580 'jis-b3-white': Specifies the JIS B3 white medium: 364 mm x 515 mm  
5581 'jis-b3-transparent': Specifies the JIS B3 transparent medium: 364 mm x 515 mm  
5582 'jis-b3-translucent': Specifies the JIS B3 translucent medium: 364 mm x 515 mm  
5583 'jis-b4-transparent': Specifies the JIS B4 transparent medium: 257 mm x 364 mm  
5584 'jis-b4-translucent': Specifies the JIS B4 translucent medium: 257 mm x 364 mm  
5585 'jis-b5-transparent': Specifies the JIS B5 transparent medium: 182 mm x 257 mm  
5586 'jis-b5-translucent': Specifies the JIS B5 translucent medium: 182 mm x 257 mm  
5587 'jis-b6-white': Specifies the JIS B6 white medium: 128 mm x 182 mm  
5588 'jis-b7-white': Specifies the JIS B7 white medium: 91 mm x 128 mm  
5589 'jis-b8-white': Specifies the JIS B8 white medium: 64 mm x 91 mm  
5590 'jis-b9-white': Specifies the JIS B9 white medium: 45 mm x 64 mm  
5591 'jis-b10-white': Specifies the JIS B10 white medium: 32 mm x 45 mm  
5592

5593 The following standard values are defined for American Standard (i.e. ANSI) engineering media:

5594 'a-white': Specifies the engineering ANSI A size white medium: 8.5 inches x 11 inches  
5595 'a-transparent': Specifies the engineering ANSI A size transparent medium: 8.5 inches x 11 inches  
5596 'a-translucent': Specifies the engineering ANSI A size translucent medium: 8.5 inches x 11 inches  
5597 'b-white': Specifies the engineering ANSI B size white medium: 11 inches x 17 inches  
5598 'b-transparent': Specifies the engineering ANSI B size transparent medium: 11 inches x 17 inches  
5599 'b-translucent': Specifies the engineering ANSI B size translucent medium: 11 inches x 17 inches  
5600 'c-white': Specifies the engineering ANSI C size white medium: 17 inches x 22 inches  
5601 'c-transparent': Specifies the engineering ANSI C size transparent medium: 17 inches x 22 inches  
5602 'c-translucent': Specifies the engineering ANSI C size translucent medium: 17 inches x 22 inches  
5603 'd-white': Specifies the engineering ANSI D size white medium: 22 inches x 34 inches  
5604 'd-transparent': Specifies the engineering ANSI D size transparent medium: 22 inches x 34 inches  
5605 'd-translucent': Specifies the engineering ANSI D size translucent medium: 22 inches x 34 inches  
5606 'e-white': Specifies the engineering ANSI E size white medium: 34 inches x 44 inches  
5607 'e-transparent': Specifies the engineering ANSI E size transparent medium: 34 inches x 44 inches  
5608 'e-translucent': Specifies the engineering ANSI E size translucent medium: 34 inches x 44 inches  
5609

5610 The following standard values are defined for American Standard (i.e. ANSI) engineering media for devices  
5611 that provide the "synchro-cut" feature (see section 14.1):

5612 'axsynchro-white': Specifies the roll paper having the width of the longer edge (11 inches) of the  
5613 engineering ANSI A size white medium and cuts synchronizing with data.

5614 'axsynchro-transparent': Specifies the roll paper having the width of the longer edge (11 inches) of the  
5615 engineering ANSI A size transparent medium and cuts synchronizing with data.  
5616 'axsynchro-translucent': Specifies the roll paper having the width of the longer edge (11 inches) of the  
5617 engineering ANSI A size translucent medium and cuts synchronizing with data.  
5618 'bxsynchro-white': Specifies the roll paper having the width of the longer edge (17 inches) of the  
5619 engineering ANSI B size white medium and cuts synchronizing with data.  
5620 'bxsynchro-transparent': Specifies the roll paper having the width of the longer edge (17 inches) of the  
5621 engineering ANSI B size transparent medium and cuts synchronizing with data.  
5622 'bxsynchro-translucent': Specifies the roll paper having the width of the longer edge (17 inches) of the  
5623 engineering ANSI B size translucent medium and cuts synchronizing with data.  
5624 'cxsynchro-white': Specifies the roll paper having the width of the longer edge (22 inches) of the  
5625 engineering ANSI C size white medium and cuts synchronizing with data.  
5626 'cxsynchro-transparent': Specifies the roll paper having the width of the longer edge (22 inches) of the  
5627 engineering ANSI C size transparent medium and cuts synchronizing with data.  
5628 'cxsynchro-translucent': Specifies the roll paper having the width of the longer edge (22 inches) of the  
5629 engineering ANSI C size translucent medium and cuts synchronizing with data.  
5630 'dxsynchro-white': Specifies the roll paper having the width of the longer edge (34 inches) of the  
5631 engineering ANSI D size white medium and cuts synchronizing with data.  
5632 'dxsynchro-transparent': Specifies the roll paper having the width of the longer edge (34 inches) of the  
5633 engineering ANSI D size transparent medium and cuts synchronizing with data.  
5634 'dxsynchro-translucent': Specifies the roll paper having the width of the longer edge (34 inches) of the  
5635 engineering ANSI D size translucent medium and cuts synchronizing with data.  
5636 'exsynchro-white': Specifies the roll paper having the width of the longer edge (44 inches) of the  
5637 engineering ANSI E size white medium and cuts synchronizing with data.  
5638 'exsynchro-transparent': Specifies the roll paper having the width of the longer edge (44 inches) of the  
5639 engineering ANSI E size transparent medium and cuts synchronizing with data.  
5640 'exsynchro-translucent': Specifies the roll paper having the width of the longer edge (44 inches) of the  
5641 engineering ANSI E size translucent medium and cuts synchronizing with data.  
5642

5643 The following standard values are defined for American Architectural engineering media:

5644 'arch-a-white': Specifies the Architectural A size white medium: 9 inches x 12 inches  
5645 'arch-a-transparent': Specifies the Architectural A size transparent medium: 9 inches x 12 inches  
5646 'arch-a-translucent': Specifies the Architectural A size translucent medium: 9 inches x 12 inches  
5647 'arch-b-white': Specifies the Architectural B size white medium: 12 inches x 18 inches  
5648 'arch-b-transparent': Specifies the Architectural B size transparent medium: 12 inches x 18 inches  
5649 'arch-b-translucent': Specifies the Architectural B size translucent medium: 12 inches x 18 inches  
5650 'arch-c-white': Specifies the Architectural C size white medium: 18 inches x 24 inches  
5651 'arch-c-transparent': Specifies the Architectural C size transparent medium: 18 inches x 24 inches  
5652 'arch-c-translucent': Specifies the Architectural C size translucent medium: 18 inches x 24 inches  
5653 'arch-d-white': Specifies the Architectural D size white medium: 24 inches x 36 inches  
5654 'arch-d-transparent': Specifies the Architectural D size transparent medium: 24 inches x 36 inches  
5655 'arch-d-translucent': Specifies the Architectural D size translucent medium: 24 inches x 36 inches  
5656 'arch-e-white': Specifies the Architectural E size white medium: 36 inches x 48 inches  
5657 'arch-e-transparent': Specifies the Architectural E size transparent medium: 36 inches x 48 inches



5658 'arch-e-translucent': Specifies the Architectural E size translucent medium: 36 inches x 48 inches  
5659

5660 The following standard values are defined for American Architectural engineering media for devices that  
5661 provide the "synchro-cut" feature (see section 14.1):

5662 'arch-axsynchro-white': Specifies the roll paper having the width of the longer edge (12 inches) of the  
5663 Architectural A size white medium and cuts synchronizing with data.

5664 'arch-axsynchro-transparent': Specifies the roll paper having the width of the longer edge (12 inches) of  
5665 the Architectural A size transparent medium and cuts synchronizing with data.

5666 'arch-axsynchro-translucent': Specifies the roll paper having the width of the longer edge (12 inches) of  
5667 the Architectural A size translucent medium and cuts synchronizing with data.

5668 'arch-bxsynchro-white': Specifies the roll paper having the width of the longer edge (18 inches) of the  
5669 Architectural B size white medium and cuts synchronizing with data.

5670 'arch-bxsynchro-transparent': Specifies the roll paper having the width of the longer edge (18 inches) of  
5671 the Architectural B size transparent medium and cuts synchronizing with data.

5672 'arch-bxsynchro-translucent': Specifies the roll paper having the width of the longer edge (18 inches) of  
5673 the Architectural B size translucent medium and cuts synchronizing with data.

5674 'arch-cxsynchro-white': Specifies the roll paper having the width of the longer edge (24 inches) of the  
5675 Architectural C size white medium and cuts synchronizing with data.

5676 'arch-cxsynchro-transparent': Specifies the roll paper having the width of the longer edge (24 inches) of  
5677 the Architectural C size transparent medium and cuts synchronizing with data.

5678 'arch-cxsynchro-translucent': Specifies the roll paper having the width of the longer edge (24 inches) of  
5679 the Architectural C size translucent medium and cuts synchronizing with data.

5680 'arch-dxsynchro-white': Specifies the roll paper having the width of the longer edge (36 inches) of the  
5681 Architectural D size white medium and cuts synchronizing with data.

5682 'arch-dxsynchro-transparent': Specifies the roll paper having the width of the longer edge (36 inches) of  
5683 the Architectural D size transparent medium and cuts synchronizing with data.

5684 'arch-dxsynchro-translucent': Specifies the roll paper having the width of the longer edge (36 inches) of  
5685 the Architectural D size translucent medium and cuts synchronizing with data.

5686 'arch-exsynchro-white': Specifies the roll paper having the width of the longer edge (48 inches) of the  
5687 Architectural E size white medium and cuts synchronizing with data.

5688 'arch-exsynchro-transparent': Specifies the roll paper having the width of the longer edge (48 inches) of  
5689 the Architectural E size transparent medium and cuts synchronizing with data.

5690 'arch-exsynchro-translucent': Specifies the roll paper having the width of the longer edge (48 inches) of  
5691 the Architectural E size translucent medium and cuts synchronizing with data.  
5692

5693 The following standard values are defined for Japanese and European Standard (i.e. ISO) engineering  
5694 media, which are of a long fixed size [ASME-Y14.1M]:

5695 'iso-a1x3-white': Specifies the ISO A1X3 white medium having the width of the longer edge (841 mm)  
5696 of the ISO A1 medium

5697 'iso-a1x3-transparent': Specifies the ISO A1X3 transparent medium having the width of the longer edge  
5698 (841 mm) of the ISO A1 medium

5699 'iso-a1x3-translucent': Specifies the ISO A1X3 translucent medium having the width of the longer edge  
5700 (841 mm) of the ISO A1 medium

5701 'iso-a1x4-white': Specifies the ISO A1X4 white medium having the width of the longer edge (841 mm)  
5702 of the ISO A1 medium

5703 'iso-a1x4-transparent': Specifies the ISO A1X4 transparent medium having the width of the longer edge  
5704 (841 mm) of the ISO A1 medium

5705 'iso-a1x4-translucent': Specifies the ISO A1X4 translucent medium having the width of the longer  
5706 edge (841 mm) of the ISO A1 medium

5707 'iso-a2x3-white': Specifies the ISO A2X3 white medium having the width of the longer edge (594 mm)  
5708 of the ISO A2 medium

5709 'iso-a2x3-transparent': Specifies the ISO A2X3 transparent medium having the width of the longer edge  
5710 (594 mm) of the ISO A2 medium

5711 'iso-a2x3-translucent': Specifies the ISO A2X3 translucent medium having the width of the longer edge  
5712 (594 mm) of the ISO A2 medium

5713 'iso-a2x4-white': Specifies the ISO A2X4 white medium having the width of the longer edge (594 mm)  
5714 of the ISO A2 medium

5715 'iso-a2x4-transparent': Specifies the ISO A2X4 transparent medium having the width of the longer edge  
5716 (594 mm) of the ISO A2 medium

5717 'iso-a2x4-translucent': Specifies the ISO A2X4 translucent medium having the width of the longer edge  
5718 (594 mm) of the ISO A2 medium

5719 'iso-a2x5-white': Specifies the ISO A2X5 white medium having the width of the longer edge (594 mm)  
5720 of the ISO A2 medium

5721 'iso-a2x5-transparent': Specifies the ISO A2X5 transparent medium having the width of the longer edge  
5722 (594 mm) of the ISO A2 medium

5723 'iso-a2x5-translucent': Specifies the ISO A2X5 translucent medium having the width of the longer edge  
5724 (594 mm) of the ISO A2 medium

5725 'iso-a3x3-white': Specifies the ISO A3X3 white medium having the width of the longer edge (420 mm)  
5726 of the ISO A3 medium

5727 'iso-a3x3-transparent': Specifies the ISO A3X3 transparent medium having the width of the longer edge  
5728 (420 mm) of the ISO A3 medium

5729 'iso-a3x3-translucent': Specifies the ISO A3X3 translucent medium having the width of the longer edge  
5730 (420 mm) of the ISO A3 medium

5731 'iso-a3x4-white': Specifies the ISO A3X4 white medium having the width of the longer edge (420 mm)  
5732 of the ISO A3 medium

5733 'iso-a3x4-transparent': Specifies the ISO A3X4 transparent medium having the width of the longer edge  
5734 (420 mm) of the ISO A3 medium

5735 'iso-a3x4-translucent': Specifies the ISO A3X4 translucent medium having the width of the longer edge  
5736 (420 mm) of the ISO A3 medium

5737 'iso-a3x5-white': Specifies the ISO A3X5 white medium having the width of the longer edge (420 mm)  
5738 of the ISO A3 medium

5739 'iso-a3x5-transparent': Specifies the ISO A3X5 transparent medium having the width of the longer edge  
5740 (420 mm) of the ISO A3 medium

5741 'iso-a3x5-translucent': Specifies the ISO A3X5 translucent medium having the width of the longer edge  
5742 (420 mm) of the ISO A3 medium

5743 'iso-a3x6-white': Specifies the ISO A3X6 white medium having the width of the longer edge (420 mm)  
5744 of the ISO A3 medium

5745 'iso-a3x6-transparent': Specifies the ISO A3X6 transparent medium having the width of the longer edge  
5746 (420 mm) of the ISO A3 medium

5747 'iso-a3x6-translucent': Specifies the ISO A3X6 translucent medium having the width of the longer edge  
5748 (420 mm) of the ISO A3 medium

5749 'iso-a3x7-white': Specifies the ISO A3X7 white medium having the width of the longer edge (420 mm)  
5750 of the ISO A3 medium

5751 'iso-a3x7-transparent': Specifies the ISO A3X7 transparent medium having the width of the longer edge  
5752 (420 mm) of the ISO A3 medium

5753 'iso-a3x7-translucent': Specifies the ISO A3X7 translucent' medium having the width of the longer  
5754 edge (420 mm) of the ISO A3 medium

5755 'iso-a4x3-white': Specifies the ISO A4X3 white medium having the width of the longer edge (297 mm)  
5756 of the ISO A4 medium

5757 'iso-a4x3-transparent': Specifies the ISO A4X3 transparent medium having the width of the longer edge  
5758 (297 mm) of the ISO A4 medium

5759 'iso-a4x3-translucent': Specifies the ISO A4X3 translucent' medium having the width of the longer  
5760 edge (297 mm) of the ISO A4 medium

5761 'iso-a4x4-white': Specifies the ISO A4X4 white medium having the width of the longer edge (297 mm)  
5762 of the ISO A4 medium

5763 'iso-a4x4-transparent': Specifies the ISO A4X4 transparent medium having the width of the longer edge  
5764 (297 mm) of the ISO A4 medium

5765 'iso-a4x4-translucent': Specifies the ISO A4X4 translucent medium having the width of the longer edge  
5766 (297 mm) of the ISO A4 medium

5767 'iso-a4x5-white': Specifies the ISO A4X5 white medium having the width of the longer edge (297 mm)  
5768 of the ISO A4 medium

5769 'iso-a4x5-transparent': Specifies the ISO A4X5 transparent medium having the width of the longer edge  
5770 (297 mm) of the ISO A4 medium

5771 'iso-a4x5-translucent': Specifies the ISO A4X5 translucent medium having the width of the longer edge  
5772 (297 mm) of the ISO A4 medium

5773 'iso-a4x6-white': Specifies the ISO A4X6 white medium having the width of the longer edge (297 mm)  
5774 of the ISO A4 medium

5775 'iso-a4x6-transparent': Specifies the ISO A4X6 transparent medium having the width of the longer edge  
5776 (297 mm) of the ISO A4 medium

5777 'iso-a4x6-translucent': Specifies the ISO A4X6 translucent medium having the width of the longer edge  
5778 (297 mm) of the ISO A4 medium

5779 'iso-a4x7-white': Specifies the ISO A4X7 white medium having the width of the longer edge (297 mm)  
5780 of the ISO A4 medium

5781 'iso-a4x7-transparent': Specifies the ISO A4X7 transparent medium having the width of the longer edge  
5782 (297 mm) of the ISO A4 medium

5783 'iso-a4x7-translucent': Specifies the ISO A4X7 translucent medium having the width of the longer edge  
5784 (297 mm) of the ISO A4 medium

5785 'iso-a4x8-white': Specifies the ISO A4X8 white medium having the width of the longer edge (297 mm)  
5786 of the ISO A4 medium

5787 'iso-a4x8-transparent': Specifies the ISO A4X8 transparent medium having the width of the longer edge  
5788 (297 mm) of the ISO A4 medium

5789 'iso-a4x8-translucent': Specifies the ISO A4X8 translucent medium having the width of the longer edge  
5790 (297 mm) of the ISO A4 medium  
5791 'iso-a4x9-white': Specifies the ISO A4X9 white medium having the width of the longer edge (297 mm)  
5792 of the ISO A4 medium  
5793 'iso-a4x9-transparent': Specifies the ISO A4X9 transparent medium having the width of the longer edge  
5794 (297 mm) of the ISO A4 medium  
5795 'iso-a4x9-translucent': Specifies the ISO A4X9 translucent medium having the width of the longer edge  
5796 (297 mm) of the ISO A4 medium  
5797

5798 The following standard values are defined for Japanese and European Standard (i.e. ISO) engineering  
5799 media, which are either a long fixed size [ASME-Y14.1M] or roll feed, for devices that provide the  
5800 "synchro-cut" feature (see section 14.1):

5801 'iso-a0xsynchro-white': Specifies the paper having the width of the longer edge (1189 mm) of the ISO  
5802 A0 white medium and cuts synchronizing with data.  
5803 'iso-a0xsynchro-transparent': Specifies the paper having the width of the longer edge (1189 mm) of the  
5804 ISO A0 transparent medium and cuts synchronizing with data.  
5805 'iso-a0xsynchro-translucent': Specifies the paper having the width of the longer edge (1189 mm) of the  
5806 ISO A0 translucent medium and cuts synchronizing with data.  
5807 'iso-a1xsynchro-white': Specifies the paper having the width of the longer edge (841 mm) of the ISO  
5808 A1 white medium and cuts synchronizing with data.  
5809 'iso-a1xsynchro-transparent': Specifies the paper having the width of the longer edge (841 mm) of the  
5810 ISO A1 transparent medium and cuts synchronizing with data.  
5811 'iso-a1xsynchro-translucent': Specifies the paper having the width of the longer edge (841 mm) of the  
5812 ISO A1 translucent medium and cuts synchronizing with data.  
5813 'iso-a2xsynchro-white': Specifies the paper having the width of the longer edge (594 mm) of the ISO  
5814 A2 white medium and cuts synchronizing with data.  
5815 'iso-a2xsynchro-transparent': Specifies the paper having the width of the longer edge (594 mm) of the  
5816 ISO A2 transparent medium and cuts synchronizing with data.  
5817 'iso-a2xsynchro-translucent': Specifies the paper having the width of the longer edge (594 mm) of the  
5818 ISO A2 translucent medium and cuts synchronizing with data.  
5819 'iso-a3xsynchro-white': Specifies the paper having the width of the longer edge (420 mm) of the ISO  
5820 A3 white medium and cuts synchronizing with data.  
5821 'iso-a3xsynchro-transparent': Specifies the paper having the width of the longer edge (420 mm) of the  
5822 ISO A3 transparent medium and cuts synchronizing with data.  
5823 'iso-a3xsynchro-translucent': Specifies the paper having the width of the longer edge (420 mm) of the  
5824 ISO A3 translucent medium and cuts synchronizing with data.  
5825 'iso-a4xsynchro-white': Specifies the paper having the width of the longer edge (297 mm) of the ISO  
5826 A4 white medium and cuts synchronizing with data.  
5827 'iso-a4xsynchro-transparent': Specifies the paper having the width of the longer edge (297 mm) of the  
5828 ISO A4 transparent medium and cuts synchronizing with data.  
5829 'iso-a4xsynchro-translucent': Specifies the paper having the width of the longer edge (297 mm) of the  
5830 ISO A4 transparent medium and cuts synchronizing with data.  
5831

5832 The following standard values are defined for American Standard (i.e. ANSI) engineering media, American  
5833 Architectural engineering media, and Japanese and European Standard (i.e. ISO) engineering media, which  
5834 are either a long fixed size [ASME-Y14.1M] or roll feed, for devices that provide the "synchro-cut" feature  
5835 and/or the "auto-select" feature (see section 14.1):

5836 'auto-white': Specifies that the printer selects the white medium with the appropriate fixed size (e.g. a1,  
5837 a2, etc.) or data-synchro size, and the selection is implementation-defined.

5838 'auto-transparent': Specifies that the printer selects the transparent medium with the appropriate fixed  
5839 size (e.g. a1, a2, etc.) or data-synchro size, and the selection is implementation-defined.

5840 'auto-translucent': Specifies that the printer selects the translucent medium with the appropriate fixed  
5841 size (e.g. a1, a2, etc.) or data-synchro size, and the selection is implementation-defined.

5842 'auto-fixed-size-white': Specifies that the printer selects the white medium with the appropriate fixed  
5843 size (e.g. a1, a2, etc.) or the appropriate long fixed size listed above.

5844 'auto-fixed-size-transparent': Specifies that the printer selects the transparent medium with the  
5845 appropriate fixed size (e.g. a1, a2, etc.) or the appropriate long fixed size listed above.

5846 'auto-fixed-size-translucent': Specifies that the printer selects the translucent medium with the  
5847 appropriate fixed size (e.g. a1, a2, etc.) or the appropriate long fixed size listed above.

5848 'auto-synchro-white': Specifies that the printer selects the white paper with the appropriate width and  
5849 cuts it synchronizing with data.

5850 'auto-synchro-transparent': Specifies that the printer selects the transparent paper with the appropriate  
5851 width and cuts it synchronizing with data.

5852 'auto-synchro-translucent': Specifies that the printer selects the translucent paper with the appropriate  
5853 width and cuts it synchronizing with data.  
5854

5855 The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

5856 'top': The top input tray in the printer.

5857 'middle': The middle input tray in the printer.

5858 'bottom': The bottom input tray in the printer.

5859 'envelope': The envelope input tray in the printer.

5860 'manual': The manual feed input tray in the printer.

5861 'large-capacity': The large capacity input tray in the printer.

5862 'main': The main input tray

5863 'side': The side input tray  
5864

5865 The following standard values are defined for media sizes (from ISO DPA):

5866 'iso-a0': Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216

5867 'iso-a1': Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216

5868 'iso-a2': Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216

5869 'iso-a3': Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216

5870 'iso-a4': Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216

5871 'iso-a5': Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216

5872 'iso-a6': Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216

5873 'iso-a7': Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216

5874 'iso-a8': Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216  
5875 'iso-a9': Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216  
5876 'iso-a10': Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216  
5877 'iso-b0': Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216  
5878 'iso-b1': Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216  
5879 'iso-b2': Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216  
5880 'iso-b3': Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216  
5881 'iso-b4': Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216  
5882 'iso-b5': Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216  
5883 'iso-b6': Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216  
5884 'iso-b7': Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216  
5885 'iso-b8': Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216  
5886 'iso-b9': Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216  
5887 'iso-b10': Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216  
5888 'na-letter': Specifies the North American letter size: 8.5 inches by 11 inches  
5889 'na-legal': Specifies the North American legal size: 8.5 inches by 14 inches  
5890 'na-8x10': Specifies the North American 8 inches by 10 inches  
5891 'na-5x7': Specifies the North American 5 inches by 7 inches  
5892 'executive': Specifies the executive size (7.25 X 10.5 in)  
5893 'folio': Specifies the folio size (8.5 X 13 in)  
5894 'invoice': Specifies the invoice size (5.5 X 8.5 in)  
5895 'ledger': Specifies the ledger size (11 X 17 in)  
5896 'quarto': Specifies the quarto size (8.5 X 10.83 in)  
5897 'iso-c3': Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269  
5898 'iso-c4': Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269  
5899 'iso-c5': Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269  
5900 'iso-c6': Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269  
5901 'iso-designated-long': Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO  
5902 269  
5903 'na-10x13-envelope': Specifies the North American 10x13 size: 10 inches by 13 inches  
5904 'na-9x12-envelope': Specifies the North American 9x12 size: 9 inches by 12 inches  
5905 'na-number-10-envelope': Specifies the North American number 10 business envelope size: 4.125  
5906 inches by 9.5 inches  
5907 'na-7x9-envelope': Specifies the North American 7x9 inch envelope size  
5908 'na-9x11-envelope': Specifies the North American 9x11 inch envelope size  
5909 'na-10x14-envelope': Specifies the North American 10x14 inch envelope size  
5910 'na-number-9-envelope': Specifies the North American number 9 business envelope size  
5911 'na-6x9-envelope': Specifies the North American 6x9 envelope size  
5912 'na-10x15-envelope': Specifies the North American 10x15 envelope size  
5913 'monarch-envelope': Specifies the Monarch envelope size (3.87 x 7.5 in)  
5914 'jis-b0': Specifies the JIS B0 size: 1030mm x 1456mm  
5915 'jis-b1': Specifies the JIS B1 size: 728mm x 1030mm  
5916 'jis-b2': Specifies the JIS B2 size: 515mm x 728mm  
5917 'jis-b3': Specifies the JIS B3 size: 364mm x 515mm  
5918 'jis-b4': Specifies the JIS B4 size: 257mm x 364mm

5919 'jis-b5': Specifies the JIS B5 size: 182mm x 257mm

5920 'jis-b6': Specifies the JIS B6 size: 128mm x 182mm

5921 'jis-b7': Specifies the JIS B7 size: 91mm x 128mm

5922 'jis-b8': Specifies the JIS B8 size: 64mm x 91mm

5923 'jis-b9': Specifies the JIS B9 size: 45mm x 64mm

5924 'jis-b10': Specifies the JIS B10 size: 32mm x 45mm

5925 The following standard values are defined for American Standard (i.e. ANSI) engineering media sizes:

5926 'a': Specifies the engineering ANSI A size medium: 8.5 inches x 11 inches

5927 'b': Specifies the engineering ANSI B size medium: 11 inches x 17 inches

5928 'c': Specifies the engineering ANSI C size medium: 17 inches x 22 inches

5929 'd': Specifies the engineering ANSI D size medium: 22 inches x 34 inches

5930 'e': Specifies the engineering ANSI E size medium: 34 inches x 44 inches

5931

5932 The following standard values are defined for American Architectural engineering media sizes:

5933 'arch-a': Specifies the Architectural A size medium: 9 inches x 12 inches

5934 'arch-b': Specifies the Architectural B size medium: 12 inches x 18 inches

5935 'arch-c': Specifies the Architectural C size medium: 18 inches x 24 inches

5936 'arch-d': Specifies the Architectural D size medium: 24 inches x 36 inches

5937 'arch-e': Specifies the Architectural E size medium: 36 inches x 48 inches

5938

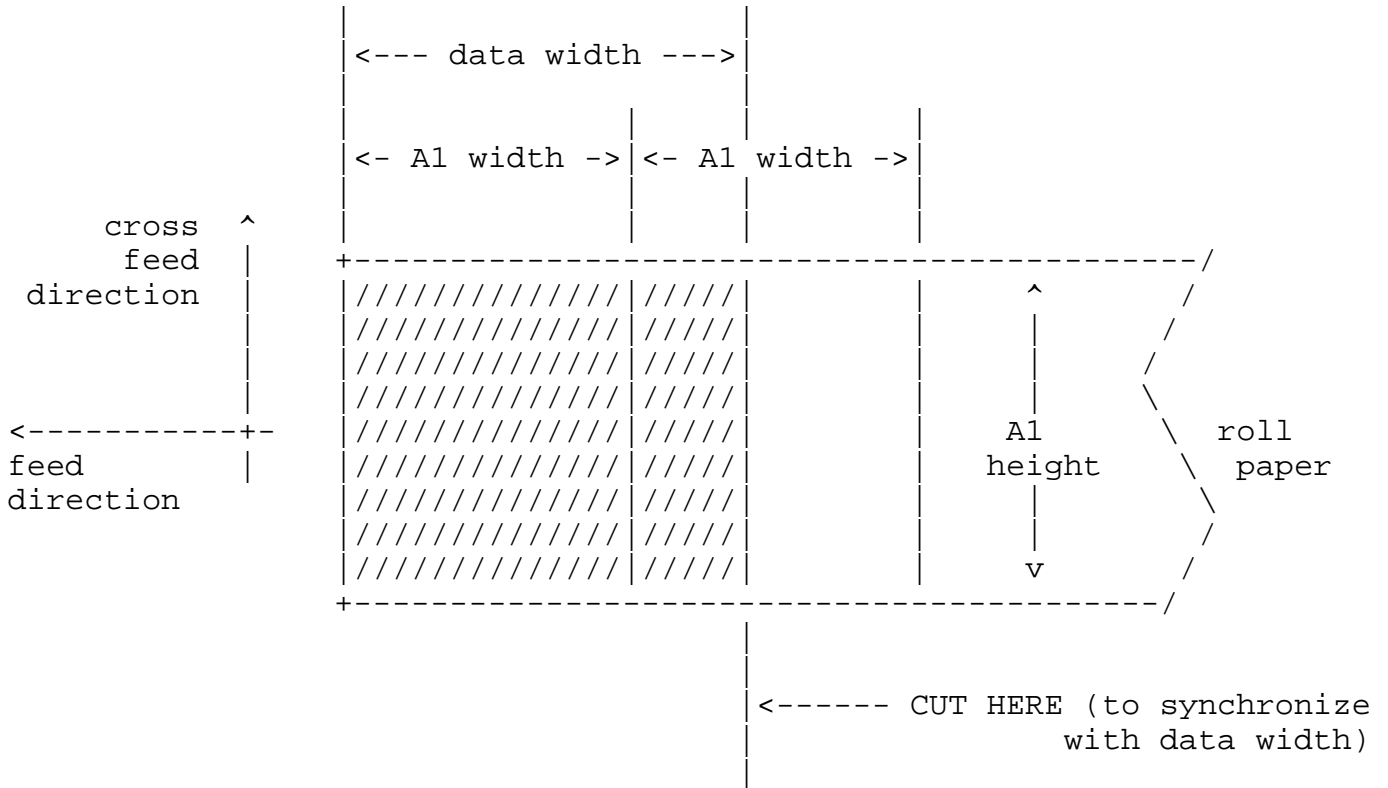
5939  
5940  
5941  
5942  
5943  
5944  
5945  
5946  
5947  
5948  
5949  
5950  
5951  
5952  
5953  
5954  
5955  
5956  
5957  
5958  
5959  
5960  
5961  
5962  
5963  
5964  
5965  
5966  
5967  
5968  
5969  
5970

**14.1. Examples**

Below are examples to supplement the engineering media value definitions.

Example 1: "Synchro-Cut", a device cutting the roll paper in synchronization with the data

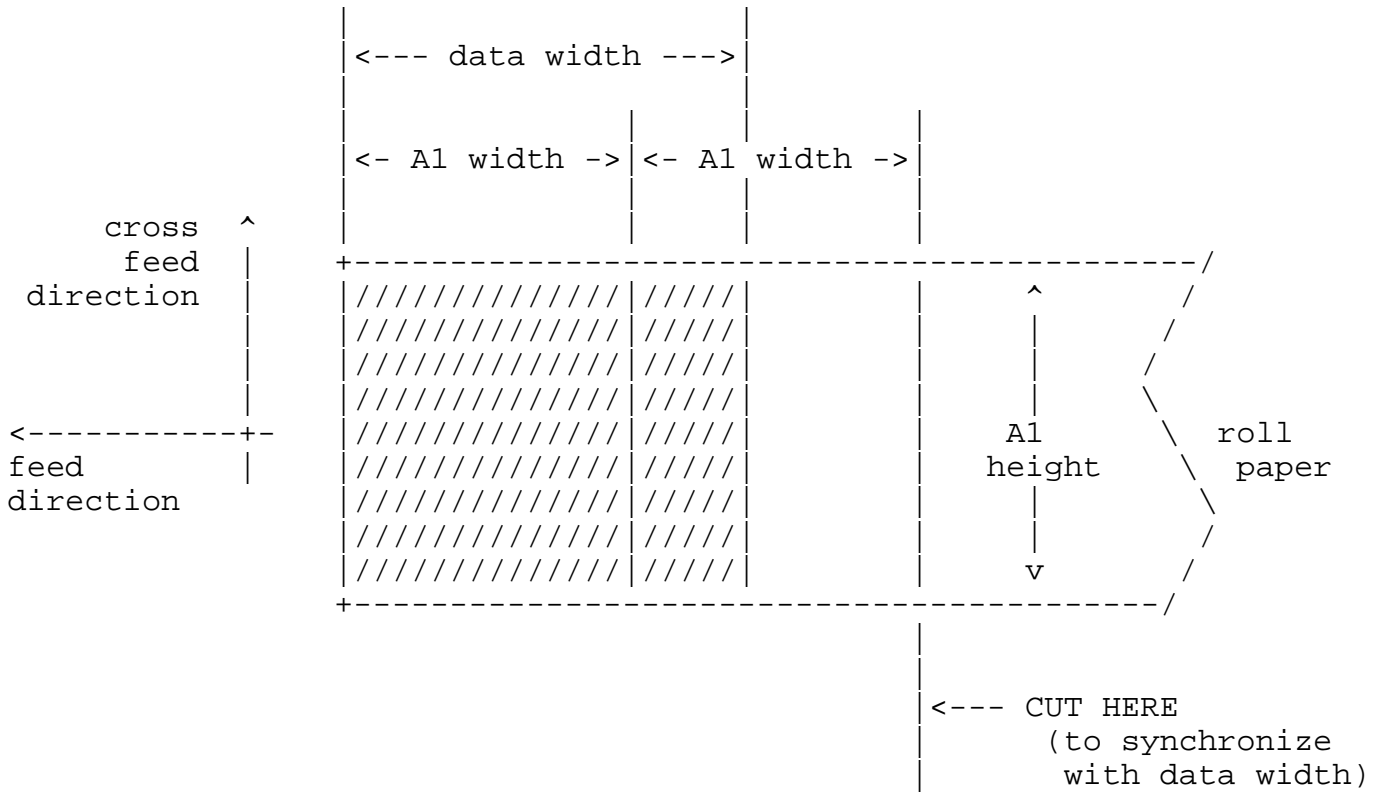
data height: A1 height  
data width (shaded): A1 width < data width < (A1 width) x 2  
specified value: 'iso-alxsynchro-white'





5971  
5972  
5973  
5974  
5975  
5976  
5977  
5978  
5979  
5980  
5981  
5982  
5983  
5984  
5985  
5986  
5987  
5988  
5989  
5990  
5991  
5992  
5993  
5994  
5995  
5996  
5997  
5998  
5999  
6000  
6001  
6002  
6003

Example 2: "Auto-Cut", a device cutting the roll paper at multiples of fixed-size media width  
data height: A1 height  
data width (shaded): A1 width < data width < (A1 width) x 2  
specified value: 'auto-fixed-size-white'



6004

6005

6006 Example 3: the 'iso-a4x4-white' fixed size paper

```

6007     paper height:           A4 height
6008     paper width:           (A4 width) x 4
6009     specified value:       'iso-a4x4-white'

```

6010

6011

```

6012 | <- A4 width -> | <- A4 width -> | <- A4 width -> | <- A4 width -> |
6013 |               |               |               |               |
6014 |               |               |               |               |

```

6013

6014

6015

6016

6017

6018

6019

6020

6021

6022

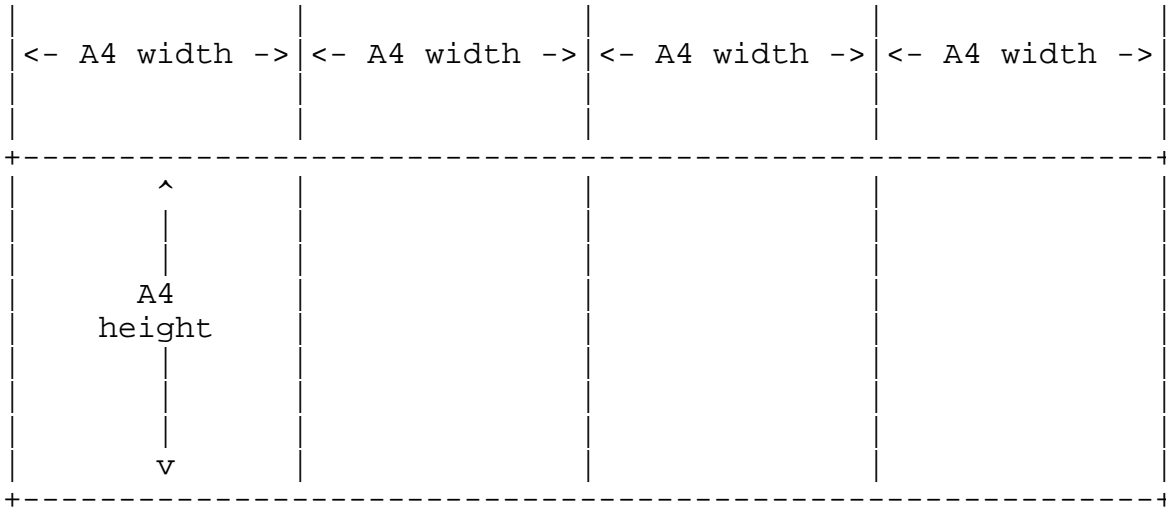
6023

6024

6025

6026

6027



6028

6029

6030 Example 4: "Synchro-Cut", a device cutting the fixed size paper in synchronization with the data

6031 data height: A4 height

6032 data width (shaded): (A4 width) x 2 < data width < (A4 width) x 3

6033 specified value: 'iso-a4synchro-white'

6034

6035

6036

6037

6038

6039

6040

6041

6042

6043

6044

6045

6046

6047

6048

6049

6050

6051

6052

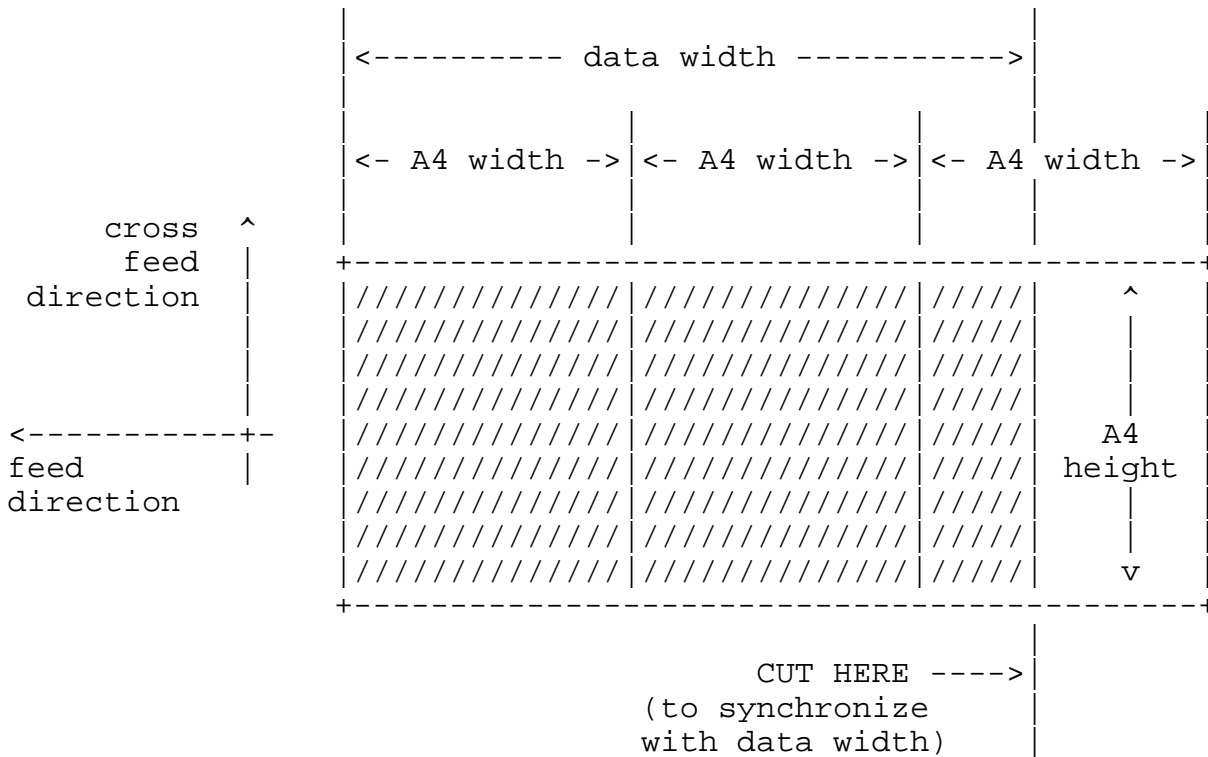
6053

6054

6055

6056

6057



6058

6059 Standard keyword values are taken from several sources.

6060 Standard values are defined (taken from DPA[ISO10175] and the Printer MIB[RFC1759]):

6061 'default': The default medium for the output device  
6062 'iso-a4-white': Specifies the ISO A4 white medium  
6063 'iso-a4-colored': Specifies the ISO A4 colored medium  
6064 'iso-a4-transparent' Specifies the ISO A4 transparent medium  
6065 'iso-a3-white': Specifies the ISO A3 white medium  
6066 'iso-a3-colored': Specifies the ISO A3 colored medium  
6067 'iso-a5-white': Specifies the ISO A5 white medium  
6068 'iso-a5-colored': Specifies the ISO A5 colored medium  
6069 'iso-b4-white': Specifies the ISO B4 white medium  
6070 'iso-b4-colored': Specifies the ISO B4 colored medium  
6071 'iso-b5-white': Specifies the ISO B5 white medium  
6072 'iso-b5-colored': Specifies the ISO B5 colored medium  
6073 'jis-b4-white': Specifies the JIS B4 white medium  
6074 'jis-b4-colored': Specifies the JIS B4 colored medium  
6075 'jis-b5-white': Specifies the JIS B5 white medium  
6076 'jis-b5-colored': Specifies the JIS B5 colored medium  
6077

6078 The following standard values are defined for North American media:

6079 'na-letter-white': Specifies the North American letter white medium  
6080 'na-letter-colored': Specifies the North American letter colored medium  
6081 'na-letter-transparent': Specifies the North American letter transparent medium  
6082 'na-legal-white': Specifies the North American legal white medium  
6083 'na-legal-colored': Specifies the North American legal colored medium  
6084

6085 The following standard values are defined for envelopes:

6086 'iso-b4-envelope': Specifies the ISO B4 envelope medium  
6087 'iso-b5-envelope': Specifies the ISO B5 envelope medium  
6088 'iso-c3-envelope': Specifies the ISO C3 envelope medium  
6089 'iso-c4-envelope': Specifies the ISO C4 envelope medium  
6090 'iso-c5-envelope': Specifies the ISO C5 envelope medium  
6091 'iso-c6-envelope': Specifies the ISO C6 envelope medium  
6092 'iso-designated-long-envelope': Specifies the ISO Designated Long envelope medium  
6093 'na-10x13-envelope': Specifies the North American 10x13 envelope medium  
6094 'na-9x12-envelope': Specifies the North American 9x12 envelope medium  
6095 'monarch-envelope': Specifies the Monarch envelope  
6096 'na-number-10-envelope': Specifies the North American number 10 business envelope medium  
6097 'na-7x9-envelope': Specifies the North American 7x9 inch envelope  
6098 'na-9x11-envelope': Specifies the North American 9x11 inch envelope

6099 'na-10x14-envelope': Specifies the North American 10x14 inch envelope  
6100 'na-number-9-envelope': Specifies the North American number 9 business envelope  
6101 'na-6x9-envelope': Specifies the North American 6x9 inch envelope  
6102 'na-10x15-envelope': Specifies the North American 10x15 inch envelope  
6103

6104 The following standard values are defined for the less commonly used media (white-only):

6105 'executive-white': Specifies the white executive medium  
6106 'folio-white': Specifies the folio white medium  
6107 'invoice-white': Specifies the white invoice medium  
6108 'ledger-white': Specifies the white ledger medium  
6109 'quarto-white': Specified the white quarto medium  
6110 'iso-a0-white': Specifies the ISO A0 white medium  
6111 'iso-a1-white': Specifies the ISO A1 white medium  
6112 'iso-a2-white': Specifies the ISO A2 white medium  
6113 'iso-a6-white': Specifies the ISO A6 white medium  
6114 'iso-a7-white': Specifies the ISO A7 white medium  
6115 'iso-a8-white': Specifies the ISO A8 white medium  
6116 'iso-a9-white': Specifies the ISO A9 white medium  
6117 'iso-10-white': Specifies the ISO A10 white medium  
6118 'iso-b0-white': Specifies the ISO B0 white medium  
6119 'iso-b1-white': Specifies the ISO B1 white medium  
6120 'iso-b2-white': Specifies the ISO B2 white medium  
6121 'iso-b3-white': Specifies the ISO B3 white medium  
6122 'iso-b6-white': Specifies the ISO B6 white medium  
6123 'iso-b7-white': Specifies the ISO B7 white medium  
6124 'iso-b8-white': Specifies the ISO B8 white medium  
6125 'iso-b9-white': Specifies the ISO B9 white medium  
6126 'iso-b10-white': Specifies the ISO B10 white medium  
6127 'jis-b0-white': Specifies the JIS B0 white medium  
6128 'jis-b1-white': Specifies the JIS B1 white medium  
6129 'jis-b2-white': Specifies the JIS B2 white medium  
6130 'jis-b3-white': Specifies the JIS B3 white medium  
6131 'jis-b6-white': Specifies the JIS B6 white medium  
6132 'jis-b7-white': Specifies the JIS B7 white medium  
6133 'jis-b8-white': Specifies the JIS B8 white medium  
6134 'jis-b9-white': Specifies the JIS B9 white medium  
6135 'jis-b10-white': Specifies the JIS B10 white medium  
6136

6137 The following standard values are defined for engineering media (white only):

6138 'a-white': Specifies the engineering A size medium  
6139 'b-white': Specifies the engineering B size medium  
6140 'c-white': Specifies the engineering C size medium

6141 'd-white': Specifies the engineering D size medium  
6142 'e-white': Specifies the engineering E size medium  
6143

6144 The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

6145 'top': The top input tray in the printer.  
6146 'middle': The middle input tray in the printer.  
6147 'bottom': The bottom input tray in the printer.  
6148 'envelope': The envelope input tray in the printer.  
6149 'manual': The manual feed input tray in the printer.  
6150 'large-capacity': The large capacity input tray in the printer.  
6151 'main': The main input tray  
6152 'side': The side input tray  
6153

6154 The following standard values are defined for media sizes (from ISO DPA):

6155 'iso-a0': Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216  
6156 'iso-a1': Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216  
6157 'iso-a2': Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216  
6158 'iso-a3': Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216  
6159 'iso-a4': Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216  
6160 'iso-a5': Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216  
6161 'iso-a6': Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216  
6162 'iso-a7': Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216  
6163 'iso-a8': Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216  
6164 'iso-a9': Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216  
6165 'iso-a10': Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216  
6166 'iso-b0': Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216  
6167 'iso-b1': Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216  
6168 'iso-b2': Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216  
6169 'iso-b3': Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216  
6170 'iso-b4': Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216  
6171 'iso-b5': Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216  
6172 'iso-b6': Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216  
6173 'iso-b7': Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216  
6174 'iso-b8': Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216  
6175 'iso-b9': Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216  
6176 'iso-b10': Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216  
6177 'na-letter': Specifies the North American letter size: 8.5 inches by 11 inches  
6178 'na-legal': Specifies the North American legal size: 8.5 inches by 14 inches  
6179 'executive': Specifies the executive size (7.25 X 10.5 in)  
6180 'folio': Specifies the folio size (8.5 X 13 in)  
6181 'invoice': Specifies the invoice size (5.5 X 8.5 in)  
6182 'ledger': Specifies the ledger size (11 X 17 in)

6183 'quarto': Specifies the quarto size (8.5 X 10.83 in)  
 6184 'iso-c3': Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269  
 6185 'iso-c4': Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269  
 6186 'iso-c5': Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269  
 6187 'iso-c6': Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269  
 6188 'iso-designated-long': Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO  
 6189 269  
 6190 'na-10x13-envelope': Specifies the North American 10x13 size: 10 inches by 13 inches  
 6191 'na-9x12-envelope': Specifies the North American 9x12 size: 9 inches by 12 inches  
 6192 'na-number-10-envelope': Specifies the North American number 10 business envelope size: 4.125  
 6193 inches by 9.5 inches  
 6194 'na-7x9-envelope': Specifies the North American 7x9 inch envelope size  
 6195 'na-9x11-envelope': Specifies the North American 9x11 inch envelope size  
 6196 'na-10x14-envelope': Specifies the North American 10x14 inch envelope size  
 6197 'na-number-9-envelope': Specifies the North American number 9 business envelope size  
 6198 'na-6x9-envelope': Specifies the North American 6x9 envelope size  
 6199 'na-10x15-envelope': Specifies the North American 10x15 envelope size  
 6200 'monarch-envelope': Specifies the Monarch envelope size (3.87 x 7.5 in)  
 6201 'jis-b0': Specifies the JIS B0 size: 1030mm x 1456mm  
 6202 'jis-b1': Specifies the JIS B1 size: 728mm x 1030mm  
 6203 'jis-b2': Specifies the JIS B2 size: 515mm x 728mm  
 6204 'jis-b3': Specifies the JIS B3 size: 364mm x 515mm  
 6205 'jis-b4': Specifies the JIS B4 size: 257mm x 364mm  
 6206 'jis-b5': Specifies the JIS B5 size: 182mm x 257mm  
 6207 'jis-b6': Specifies the JIS B6 size: 128mm x 182mm  
 6208 'jis-b7': Specifies the JIS B7 size: 91mm x 128mm  
 6209 'jis-b8': Specifies the JIS B8 size: 64mm x 91mm  
 6210 'jis-b9': Specifies the JIS B9 size: 45mm x 64mm  
 6211 'jis-b10': Specifies the JIS B10 size: 32mm x 45mm

6212 The following standard values are defined for engineering media sizes:

6213 'a': Specifies the engineering A size: 8.5 inches x 11 inches  
 6214 'b': Specifies the engineering B size: 11 inches x 17 inches  
 6215 'c': Specifies the engineering C size: 17 inches x 22 inches  
 6216 'd': Specifies the engineering D size: 22 inches x 34 inches  
 6217 'e': Specifies the engineering E size: 34 inches x 44 inches  
 6218

## 6219 **15. APPENDIX D: Processing IPP Attributes**

6220 When submitting a print job to a Printer object, the IPP model allows a client to supply operation and Job  
 6221 Template attributes along with the document data. These Job Template attributes in the create request  
 6222 affect the rendering, production and finishing of the documents in the job. Similar types of instructions  
 6223 may also be contained in the document to be printed, that is, embedded within the print data itself. In

6224 addition, the Printer has a set of attributes that describe what rendering and finishing options which are  
6225 supported by that Printer. This model, which allows for flexibility and power, also introduces the potential  
6226 that at job submission time, these client-supplied attributes may conflict with either:

- 6227 - what the implementation is capable of realizing (i.e., what the Printer supports), as well as
  - 6228 - the instructions embedded within the print data itself.
- 6229

6230 The following sections describe how these two types of conflicts are handled in the IPP model.

## 6231 **15.1 Fidelity**

6232 If there is a conflict between what the client requests and what a Printer object supports, the client may  
6233 request one of two possible conflict handling mechanisms:

- 6234 1) either reject the job since the job can not be processed exactly as specified, or
  - 6235 2) allow the Printer to make any changes necessary to proceed with processing the Job the best it can.
- 6236

6237 In the first case the client is indicating to the Printer object: "Print the job exactly as specified with no  
6238 exceptions, and if that can't be done, don't even bother printing the job at all." In the second case, the client  
6239 is indicating to the Printer object: "It is more important to make sure the job is printed rather than be  
6240 processed exactly as specified; just make sure the job is printed even if client supplied attributes need to be  
6241 changed or ignored."

6242 The IPP model accounts for this situation by introducing an "ipp-attribute-fidelity" attribute.

6243 In a create request, "ipp-attribute-fidelity" is a boolean operation attribute that is **OPTIONALLY** supplied  
6244 by the client. The value 'true' indicates that total fidelity to client supplied Job Template attributes and  
6245 values is required. The client is requesting that the Job be printed exactly as specified, and if that is not  
6246 possible then the job **MUST** be rejected rather than processed incorrectly. The value 'false' indicates that a  
6247 reasonable attempt to print the Job is acceptable. If a Printer does not support some of the client supplied  
6248 Job Template attributes or values, the Printer **MUST** ignore them or substitute any supported value for  
6249 unsupported values, respectively. The Printer may choose to substitute the default value associated with  
6250 that attribute, or use some other supported value that is similar to the unsupported requested value. For  
6251 example, if a client supplies a "media" value of 'na-letter', the Printer may choose to substitute 'iso-a4' rather  
6252 than a default value of 'envelope'. If the client does not supply the "ipp-attribute-fidelity" attribute, the  
6253 Printer assumes a value of 'false'.

6254 Each Printer implementation **MUST** support both types of "fidelity" printing (that is whether the client  
6255 supplies a value of 'true' or 'false'):

- 6256 - If the client supplies 'false' or does not supply the attribute, the Printer object **MUST** always accept the  
6257 request by ignoring unsupported Job Template attributes and by substituting unsupported values of  
6258 supported Job Template attributes with supported values.
- 6259 - If the client supplies 'true', the Printer object **MUST** reject the request if the client supplies  
6260 unsupported Job Template attributes.



6261

6262 Since a client can always query a Printer to find out exactly what is and is not supported, "ipp-attribute-  
6263 fidelity" set to 'false' is useful when:

- 6264 1) The End-User uses a command line interface to request attributes that might not be supported.
- 6265 2) In a GUI context, if the End User expects the job might be moved to another printer and prefers a  
6266 sub-optimal result to nothing at all.
- 6267 3) The End User just wants something reasonable in lieu of nothing at all.

6268

## 6269 15.2 Page Description Language (PDL) Override

6270 If there is a conflict between the value of an IPP Job Template attribute and a corresponding instruction in  
6271 the document data, the value of the IPP attribute SHOULD take precedence over the document instruction.  
6272 Consider the case where a previously formatted file of document data is sent to an IPP Printer. In this case,  
6273 if the client supplies any attributes at job submission time, the client desires that those attributes override  
6274 the embedded instructions. Consider the case were a previously formatted document has embedded in it  
6275 commands to load 'iso-a4' media. However, the document is passed to an end user that only has access to a  
6276 printer with 'na-letter' media loaded. That end user most likely wants to submit that document to an IPP  
6277 Printer with the "media" Job Template attribute set to 'na-letter'. The job submission attribute should take  
6278 precedence over the embedded PDL instruction. However, until companies that supply document data  
6279 interpreters allow a way for external IPP attributes to take precedence over embedded job production  
6280 instructions, a Printer might not be able to support the semantics that IPP attributes override the embedded  
6281 instructions.

6282 The IPP model accounts for this situation by introducing a "pdl-override-supported" attribute that describes  
6283 the Printer objects capabilities to override instructions embedded in the PDL data stream. The value of the  
6284 "pdl-override-supported" attribute is configured by means outside the scope of this IPP/1.1 document.

6285 This REQUIRED Printer attribute takes on the following values:

- 6286 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values take  
6287 precedence over embedded instructions in the document data, however there is no guarantee.
- 6288 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP attribute  
6289 values take precedence over embedded instructions in the document data.

6290

6291 At job processing time, an implementation that supports the value of 'attempted' might do one of several  
6292 different actions:

- 6293 1) Generate an output device specific command sequence to realize the feature represented by the IPP  
6294 attribute value.
- 6295 2) Parse the document data itself and replace the conflicting embedded instruction with a new  
6296 embedded instruction that matches the intent of the IPP attribute value.
- 6297 3) Indicate to the Printer that external supplied attributes take precedence over embedded instructions  
6298 and then pass the external IPP attribute values to the document data interpreter.

6299 4) Anything else that allows for the semantics that IPP attributes override embedded document data  
6300 instructions.  
6301

6302 Since 'attempted' does not offer any type of guarantee, even though a given Printer object might not do a  
6303 very "good" job of attempting to ensure that IPP attributes take a higher precedence over instructions  
6304 embedded in the document data, it would still be a conforming implementation.

6305 At job processing time, an implementation that supports the value of 'not-attempted' might do one of the  
6306 following actions:

- 6307 1) Simply pre-pend the document data with the PDL instruction that corresponds to the client-supplied  
6308 PDL attribute, such that if the document data also has the same PDL instruction, it will override  
6309 what the Printer object pre-pended. In other words, this implementation is using the same  
6310 implementation semantics for the client-supplied IPP attributes as for the Printer object defaults.
- 6311 2) Parse the document data and replace the conflicting embedded instruction with a new embedded  
6312 instruction that approximates, but does not match, the semantic intent of the IPP attribute value.  
6313

6314 Note: The "ipp-attribute-fidelity" attribute applies to the Printer's ability to either accept or reject other  
6315 unsupported Job Template attributes. In other words, if "ipp-attribute-fidelity" is set to 'true', a Job is  
6316 accepted if and only if the client supplied Job Template attributes and values are supported by the Printer.  
6317 Whether these attributes actually affect the processing of the Job when the document data contains  
6318 embedded instructions depends on the ability of the Printer to override the instructions embedded in the  
6319 document data with the semantics of the IPP attributes. If the document data attributes can be overridden  
6320 ("pdl-override-supported" set to 'attempted'), the Printer makes an attempt to use the IPP attributes when  
6321 processing the Job. If the document data attributes can not be overridden ("pdl-override-supported" set to  
6322 'not-attempted'), the Printer makes no attempt to override the embedded document data instructions with the  
6323 IPP attributes when processing the Job, and hence, the IPP attributes may fail to affect the Job processing  
6324 and output when the corresponding instruction is embedded in the document data.

### 6325 **15.3 Using Job Template Attributes During Document Processing.**

6326 The Printer object uses some of the Job object's Job Template attributes during the processing of the  
6327 document data associated with that job. These include, but are not limited to, "orientation-requested",  
6328 "number-up", "sides", "media", and "copies". The processing of each document in a Job Object MUST  
6329 follow the steps below. These steps are intended only to identify when and how attributes are to be used in  
6330 processing document data and any alternative steps that accomplishes the same effect can be used to  
6331 implement this specification document.

- 6332 1. Using the client supplied "document-format" attribute or some form of document format detection  
6333 algorithm (if the value of "document-format" is not specific enough), determine whether or not the  
6334 document data has already been formatted for printing. If the document data has been formatted,  
6335 then go to step 2. Otherwise, the document data MUST be formatted. The formatting detection  
6336 algorithm is implementation defined and is not specified by this document. The formatting of the  
6337 document data uses the "orientation-requested" attribute to determine how the formatted print data  
6338 should be placed on a print-stream page, see section 4.2.10 for the details.

6339

6340

2. The document data is in the form of a print-stream in a known media type. The "page-ranges" attribute is used to select, as specified in section 4.2.7, a sub-sequence of the pages in the print-stream that are to be processed and images.

6342

6343

6344

3. The input to this step is a sequence of print-stream pages. This step is controlled by the "number-up" attribute. If the value of "number-up" is N, then during the processing of the print-stream pages, each N print-stream pages are positioned, as specified in section 4.2.9, to create a single impression. If a given document does not have N more print-stream pages, then the completion of the impression is controlled by the "multiple-document-handling" attribute as described in section 4.2.4; when the value of this attribute is 'single-document' or 'single-document-new-sheet', the print-stream pages of document data from subsequent documents is used to complete the impression.

6345

6346

6347

6348

6349

6350

6351

The size(scaling), position(translation) and rotation of the print-stream pages on the impression is implementation defined. Note that during this process the print-stream pages may be rendered to a form suitable for placing on the impression; this rendering is controlled by the values of the "printer-resolution" and "print-quality" attributes as described in sections 4.2.12 and 4.2.13. In the case N=1, the impression is nearly the same as the print-stream page; the differences would only be in the size, position and rotation of the print-stream page and/or any decoration, such as a frame to the page, that is added by the implementation.

6352

6353

6354

6355

6356

6357

6358

6359

4. The collection of impressions is placed, in sequence, onto sides of the media sheets. This placement is controlled by the "sides" attribute and the orientation of the print-stream page, as described in section 4.2.8. The orientation of the print-stream pages affects the orientation of the impression; for example, if "number-up" equals 2, then, typically, two portrait print-stream pages become one landscape impression. Note that the placement of impressions onto media sheets is also controlled by the "multiple-document-handling" attribute as described in section 4.2.4.

6360

6361

6362

6363

6364

6365

6366

5. The "copies" and "multiple-document-handling" attributes are used to determine how many copies of each media instance are created and in what order. See sections 4.2.5 and 4.2.4 for the details.

6367

6368

6369

6. When the correct number of copies are created, the media instances are finished according to the values of the "finishings" attribute as described in 4.2.6. Note that sometimes finishing operations may require manual intervention to perform the finishing operations on the copies, especially uncollated copies. This document allows any or all of the processing steps to be performed automatically or manually at the discretion of the Printer object.

6370

6371

6372

6373

6374

6375

## 16. APPENDIX E: Generic Directory Schema

6376

6377

6378

6379

6380

This section defines a generic schema for an entry in a directory service. A directory service is a means by which service users can locate service providers. In IPP environments, this means that IPP Printers can be registered (either automatically or with the help of an administrator) as entries of type printer in the directory using an implementation specific mechanism such as entry attributes, entry type fields, specific branches, etc. Directory clients can search or browse for entries of type printer. Clients use the directory

6381 service to find entries based on naming, organizational contexts, or filtered searches on attribute values of  
 6382 entries. For example, a client can find all printers in the "Local Department" context. Authentication and  
 6383 authorization are also often part of a directory service so that an administrator can place limits on end users  
 6384 so that they are only allowed to find entries to which they have certain access rights. IPP itself does not  
 6385 require any specific directory service protocol or provider.

6386 Note: Some directory implementations allow for the notion of "aliasing". That is, one directory entry object  
 6387 can appear as multiple directory entry object with different names for each object. In each case, each alias  
 6388 refers to the same directory entry object which refers to a single IPP Printer object.

6389 The generic schema is a subset of IPP Printer Job Template and Printer Description attributes (sections 4.2  
 6390 and 4.4). These attributes are identified as either RECOMMENDED or OPTIONAL for the directory entry  
 6391 itself. This conformance labeling is NOT the same conformance labeling applied to the attributes of IPP  
 6392 Printers objects. The conformance labeling in this Appendix is intended to apply to directory templates and  
 6393 to IPP Printer implementations that subscribe by adding one or more entries to a directory.  
 6394 RECOMMENDED attributes SHOULD be associated with each directory entry. OPTIONAL attributes  
 6395 MAY be associated with the directory entry (if known or supported). In addition, all directory entry  
 6396 attributes SHOULD reflect the current attribute values for the corresponding Printer object.

6397 The names of attributes in directory schema and entries SHOULD be the same as the IPP Printer attribute  
 6398 names as shown, as much as possible.

6399 In order to bridge between the directory service and the IPP Printer object, one of the RECOMMENDED  
 6400 directory entry attributes is the Printer object's "printer-uri-supported" attribute. The directory client queries  
 6401 the "printer-uri-supported" attribute (or its equivalent) in the directory entry and then the IPP client  
 6402 addresses the IPP Printer object using one of its URIs. The "uri-security-supported" attribute identifies the  
 6403 protocol (if any) used to secure a channel.

6404 The following attributes define the generic schema for directory entries of type PRINTER:

6405	printer-uri-supported	RECOMMENDED	Section 4.4.1
6406	uri-authentication-supported	RECOMMENDED	Section 4.4.2
6407	uri-security-supported	RECOMMENDED	Section 4.4.3
6408	printer-name	RECOMMENDED	Section 4.4.4
6409	printer-location	RECOMMENDED	Section 4.4.5
6410	printer-info	OPTIONAL	Section 4.4.6
6411	printer-more-info	OPTIONAL	Section 4.4.7
6412	printer-make-and-model	RECOMMENDED	Section 4.4.9
6413	ipp-versions-supported	RECOMMENDED	Section 4.4.14
6414	multiple-document-jobs-supported	OPTIONAL	Section 4.4.16
6415	charset-supported	OPTIONAL	Section 4.4.18
6416	generated-natural-language-		
6417	supported	OPTIONAL	Section 4.4.20
6418	document-format-supported	RECOMMENDED	Section 4.4.22
6419	color-supported	RECOMMENDED	Section 4.4.26
6420	compression-supported	RECOMMENDED	Section 4.4.32

6421	pages-per-minute	OPTIONAL	Section 4.4.36
6422	pages-per-minute-color	OPTIONAL	Section 4.4.37
6423			
6424	finishings-supported	OPTIONAL	Section 4.2.6
6425	number-up-supported	OPTIONAL	Section 4.2.7
6426	sides-supported	RECOMMENDED	Section 4.2.8
6427	media-supported	RECOMMENDED	Section 4.2.11
6428	printer-resolution-supported	OPTIONAL	Section 4.2.12
6429	print-quality-supported	OPTIONAL	Section 4.2.13

6430

6431

6432

## 17. APPENDIX F: Differences between the IPP/1.0 and IPP/1.1 "Model and Semantics" Documents

6433

6434

6435

6436

This Appendix is divided into two lists that summarize the differences between IPP/1.1 (this document) and IPP/1.0 [RFC2566]. The section numbers refer to the numbers in this document which in some cases have changed from RFC 2566. When a change affects multiple sections, the item is listed once in the order of the first section affected and the remaining affected section numbers are indicated.

6437

6438

6439

The first list contains extensions and clarifications and the second list contains changes in semantics or conformance. However, client and IPP object implementations of IPP/1.0 MAY implement any of the extensions and clarifications in this document.

6440

The following extensions and clarifications have been incorporated into this document:

6441

6442

6443

6444

6445

6446

6447

6448

6449

6450

6451

6452

6453

6454

6455

6456

6457

6458

6459

6460

6461

6462

6463

6464

6465

6466

6467

6468

1. Section 2.1 - clarified that the term "client" can be either contained in software controlled by an end user or a part of a print server that controls devices.
2. Section 2 - clarified that the term "IPP object" and "Printer object" can either be embedded in a device object or part of a print server that accepts IPP requests.
3. Section 2.4 - added the description of the new "uri-authentication-supported" Printer Description attribute.
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 have their own status code.
5. Section 3.1.3 - clarified that multiple occurrences of the same attribute in an attribute group is malformed. An IPP Printer MAY reject the request or choose one of the attributes.
6. Section 3.1.6 - reorganized this section into sub-sections to separately describe "status-code", "status-message", "detailed-status-message", and "document-access-error" attributes.
7. Section 3.1.6.1 - clarified the error status codes and their relationship to operation attributes.
8. Section 3.1.6.3 - Added the OPTIONAL "detailed-status-message (text(MAX))" operation attribute to provide additional more detailed information about a response.
9. Section 3.1.6.4 and 3.2.2 - Added the OPTIONAL "document-access-error (text(MAX))" operation attribute for use with Print-URI and Send-URI responses.
10. Sections 3.1.7 - Added this new section to clarify returning Unsupported Attributes for all operations, including only returning attributes that were in the request. Moved the text from section 3.2.1.2 Unsupported Attributes to this section.
11. Sections 3.1.7 and 4.1 - clarified the encoding of the "out-of-band" 'unsupported' and 'unknown' values.
12. Section 3.1.8 - clarified that only the version number parameter will be carried forward into future major or minor versions of the protocol.
13. Section 3.1.8 - relaxed the requirements to increment the major version number in future versions of the Model and Semantics document.
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 a Create-Job operation.

- 6469 15. Section 3.1.9 - clarified that a non-spooling Printer MAY accept zero or more subsequent jobs while  
6470 processing a job and flow control them down. Subsequent create requests are rejected with the  
6471 'server-error-busy' error status.
- 6472 16. Section 3.2.1.1 - clarified the validation of the "compression" operation attribute and its relationship  
6473 to the validation of the "document-format" attribute and returning Unsupported Attributes.
- 6474 17. Sections 3.2.1.1, 4.3.8, 13.1.4.16, and 13.1.4.17 - added the 'client-error-compression-not-  
6475 supported', 'client-error-compression-error' status codes and the 'unsupported-compression' and  
6476 'compression-error' job-state-reasons.
- 6477 18. Sections 3.2.1.1 and 4.3.8 - added 'unsupported-document-format' and 'document-format-error' job-  
6478 state-reasons.
- 6479 19. Sections 3.2.2, 4.3.8 and 13.1.4.19 - added 'client-error-document-access-error' status code and  
6480 'document-access-error' job state reason.
- 6481 20. Section 3.2.5.2 and 3.2.6.2 - clarified that the Unsupported Attributes group MUST NOT include  
6482 attributes not requested in the Get-Printer-Attributes request.
- 6483 21. Section 3.2.6 - clarified that "limit" takes precedence over "which-jobs" and "my-jobs".
- 6484 22. Section 3.2.6.2 - clarified that Get-Jobs returns 'successful-ok' when no jobs to return.
- 6485 23. Sections 3.2.7, 3.2.8, and 3.2.9 - added the OPTIONAL Pause-Printer, Resume-Printer, and Purge-  
6486 Jobs operations
- 6487 24. Section 3.3.1 - clarified that the authorization required for a Send-Document request MUST be the  
6488 same user as the Create-Job or an operator.
- 6489 25. Section 3.3.1.1 - clarified that a Create-Job Send-Document with "last-document" = 'true' and no  
6490 data is not an error; its a job with no documents.
- 6491 26. Sections 3.3.5, 3.3.6, and 3.3.7 - added the OPTIONAL Hold-Job, Release-Job, and Restart-Job  
6492 operations. Clarified the Restart-Job operation so that the Printer MUST re-fetch any documents  
6493 passed by-reference (Print-URI or Send-URI).
- 6494 27. Section 4.1 - clarified that the encoding of the out-of-band values are specified in the Encoding and  
6495 Transport" document.
- 6496 28. Section 4.1 - Clarified that the requirement that clients MUST NOT send "out-of-band" values in  
6497 requests applies only to operations defined in this document. Other operations are allowed to define  
6498 "out-of-band" values that clients can supply.
- 6499 29. Sections 4.1.1 and 4.1.2 - clarified that the maximum 'text' and 'name' values of 1023 and 255 are  
6500 for the 'textWithoutLanguage' portion of the 'textWithLanguage' form, so that the maximum number  
6501 of octets for the actual text and name data is the same for the without and with language forms; the  
6502 'naturalLanguage' part is in addition.
- 6503 30. Section 4.1.9 - clarified that 'mimeType' values can include any parameters from the IANA  
6504 Registry, not just charset parameters.
- 6505 31. Section 4.1.9.1 - clarified that 'application/octet-stream' auto-sensing can happen at create request  
6506 time and/or job/document processing time.
- 6507 32. Section 4.1.9.1 - clarified that auto-sensing involves the Printer examining some number of octets of  
6508 document data using an implementation-dependent method.
- 6509 33. Section 4.1.14 - clarified that the localization of dateTime by the client includes the time zone.
- 6510 34. Section 4.2 - clarified that xxx-supported have multiple keywords and/or names by adding  
6511 parentheses to the table to give: (1setOf (type3 keyword | name))
- 6512 35. Section 4.2.2 - added the 'indefinite' keyword value to the "job-hold-until" attribute for use with the  
6513 create operations and Hold-Job and Restart-Job operations.

- 6514 36. Section 4.2.6 - added more enum values to the "finishings" Job Template attribute.  
6515 37. Section 4.2.6 - clarified that the landscape definition is a rotation of the image with respect to the  
6516 medium.  
6517 38. Section 4.3.7 - added that a forwarding server that cannot get any job state MAY return the job's  
6518 state as 'completed', provided that it also return the new 'queued-in-device' job state reason.  
6519 39. Section 4.3.7.2 - added the Partitioning of Job States section to clarify the concepts of Job  
6520 Retention, Job History, and Job Removal.  
6521 40. Section 4.3.8 - added 'job-data-insufficient' job state reason to indicate whether sufficient data has  
6522 arrived for the document to start to be processed.  
6523 41. Section 4.3.8 - added 'document-access-error' job state reason to indicate an access error of any kind.  
6524 42. Section 4.3.8 - added 'job-queued-for-marker' job state reason to indicate whether the job has  
6525 completed some processing and is waiting for the marker.  
6526 43. Section 4.3.8 - added 'unsupported-compression' and 'compression-error' job state reasons to  
6527 indicate compression not supported or compression processing error after the create has been  
6528 accepted.  
6529 44. Section 4.3.8 - added 'unsupported-document-format' and 'document-format-error' job state reasons  
6530 to indicate document not supported or document format processing error after the create has been  
6531 accepted.  
6532 45. Section 4.3.8 - added 'queued-in-device' job state reason to indicate that a job as been forwarded to a  
6533 print system or device that does not provide any job status.  
6534 46. Section 4.3.10 - added "job-detailed-status-messages (1setOf text(MAX)) for returning detailed  
6535 error messages.  
6536 47. Section 4.3.11 - added the "job-document-access-errors (1setOf text(MAX))  
6537 48. Section 4.3.14.2 - clarified that the time recorded is the first time processing since the create  
6538 operation or the Restart-Job operation.  
6539 49. Section 4.3.14.2 and 4.3.14.3 - clarified that the out-of-band value 'no-value' is returned if the job  
6540 has not started processing or has not completed, respectively.  
6541 50. Section 4.3.14 - Added the OPTIONAL "date-time-at-creation", "date-time-at-processing", and  
6542 "date-time-at-completed" Event Time Job Description attributes  
6543 51. Section 4.4.3 - added the 'tls' value to "uri-security-supported" attribute.  
6544 52. Section 4.4.3 - clarified "uri-security-supported" is orthogonal to Client Authentication so that 'none'  
6545 does not exclude Client Authentication.  
6546 53. Section 4.4.11 - simplified the "printer-state" descriptions while generalizing to allow high end  
6547 devices that interpret one or more jobs while marking another. Indicated that 'spool-area-full' and  
6548 'stopped-partly' "printer-state-reasons" may be used to provide further state information.  
6549 54. Section 4.4.12 - added the 'moving-to-paused' keyword value to the "printer-state-reasons" attribute  
6550 for use with the Pause-Printer operation.  
6551 55. Section 4.4.12 - replaced the duplicate 'marker-supply-low' keyword with the missing 'toner-empty'  
6552 keyword for the "printer-state-reasons" attribute. (This correction was also made before RFC 2566  
6553 was published).  
6554 56. Section 4.4.12 - clarified 'spool-area-full' "printer-state-reasons" to include non-spooling printers to  
6555 indicate when it can and cannot accept another job.  
6556 57. Section 4.4.15 - added the enum values to the "operations-supported" attribute for the new  
6557 operations. Clarified that the values of this attribute are encoded as any enum, namely 32-bit values.



- 6558 58. Section 4.4.30 - clarified that the dateTime value of "printer-current-time" is on a "best efforts  
6559 basis". If a proper date-time cannot be obtained, the implementation returns the 'no-value' out-of-  
6560 band value. Also clarified that the time zone NEED NOT be the time zone that the people near the  
6561 device use and that the client SHOULD display the dateTime attributes in the user's local time.
- 6562 59. Sections 4.4.36 and 4.4.37 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-  
6563 color" Printer Description attributes.
- 6564 60. Section 5.1 - clarified that the client conformance requirements apply to clients controlled by an end  
6565 user and clients in servers.
- 6566 61. Section 5.1 - clarified that any response MAY contain additional attribute groups, attributes,  
6567 attribute syntaxes, or attribute values.
- 6568 62. Section 5.1 - clarified that a client SHOULD do its best to prevent a channel from being closed by a  
6569 lower layer when the channel is flow controlled off by the IPP Printer.
- 6570 63. Section 5.2 - clarified that the IPP object requirements apply to objects embedded in devices or that  
6571 are parts of servers.
- 6572 64. Section 5.2.2 - clarified that IPP objects MAY return operation responses that contain attribute  
6573 groups, attribute names, attribute syntaxes, attribute values, and status codes that are extensions to  
6574 this standard.
- 6575 65. Section 6 - changed the terminology of "private extensions" to "vendor extensions" and indicated  
6576 that they are registered with IANA along with IETF standards track extensions.
- 6577 66. Section 6.7 - inserted this section on registering out-of-band attribute values with IANA as  
6578 extensions.
- 6579 67. Section 8.3 - clarified the use of URIs for each Client Authentication mechanism.
- 6580 68. Section 8.5 - added the security discussion around the new operator/administrator operations.
- 6581 69. Section 13.1.4.16 - added client-error-compression-not-supported (0x040F)
- 6582 70. Section 13.1.4.17 - added client-error-compression-error (0x0410)
- 6583 71. Section 13.1.4.18 - added client-error-document-format-error (0x0411)
- 6584 72. Section 13.1.4.19 - added client-error-document-access-error (0x0412)
- 6585 73. Section 13.1.5.10 - added server-error-multiple-document-jobs-not-supported (0x0509)
- 6586 74. Section 14 - added 'a-white', 'b-white', 'c-white', 'd-white', and 'e-white' and clarified that the existing  
6587 'a', 'b', 'c', 'd', and 'e' values are size values. Added American, Japanese, and European Engineering  
6588 sizes, filled out -transparent and -translucent media names and drawings for the synchro cut sizes.
- 6589 75. Section 16 - softened the RECOMMENDATION for IPP Printer attributes in a Directory schema so  
6590 that they can have equivalents.
- 6591 76. Section 16 - added the OPTIONAL "pages-per-minute" and "pages-per-minute-color" Printer  
6592 attributes to the Directory schema.
- 6593 77. Section 16 - added OPTIONAL "multiple-document-jobs-supported" to the Directory schema.
- 6594 78. 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 3.1.6.3 - allowed a Printer to localize the "detailed-status-message" operation response  
6598 attribute, but indicated that such localization might obscure the technical meaning of such  
6599 messages.
- 6600 2. Section 3.1.8, 5.2.4, and 13.1.5.4 - Clients and IPP objects MUST support version 1.1  
6601 conformance requirements. It is recommended that they interoperate with 1.0. Also clarified

- 6602 that IPP Printers MUST accept '1.1' requests. It is recommended that they also accept '1.x'  
6603 requests.
- 6604 3. Section 3.2.1.1 and section 4.4.32 - changed the "compression" operation and the "compression-  
6605 supported" Printer Description attribute from OPTIONAL to REQUIRED.
  - 6606 4. Sections 3.2.1.2 and 4.3.8 - changed "job-state-reasons" from RECOMMENDED to REQUIRED,  
6607 so that "job-state-reasons" MUST be returned in create operation responses.
  - 6608 5. Sections 3.2.4, 3.3.1, 4.4.16, and 16 - changed Create-Job/Send-Document so that they MAY be  
6609 implemented while only supporting one document jobs. Added the "multiple-document-jobs-  
6610 supported" boolean Printer Description attribute to indicate whether Create-Job/Send-  
6611 Document support multiple document jobs or not. Added to the Directory schema.
  - 6612 6. Section 4.1.9 - deleted 'text/plain; charset=iso-10646-ucs-2', since binary is not legal with the 'text'  
6613 type.
  - 6614 7. Section 4.1.9.1 - added the RECOMMENDATION that a Printer indicate by printing on the job's  
6615 job-start-sheet that auto-sensing has occurred and what document format was auto-sensed.
  - 6616 8. Section 4.2.4 - indicated that the "multiple-document-handling" Job Template attribute MUST be  
6617 supported with at least one value if the Printer supports multiple documents per job
  - 6618 9. Section 4.3.7.2 - indicated that the 'job-restartable' job state reason SHOULD be supported if the  
6619 Restart-Job operation is supported.
  - 6620 10. Section 4.3.8 - changed "job-state-reasons" from RECOMMENDED to REQUIRED.
  - 6621 11. Section 4.3.8 - clarified the conformance of the values of the "job-state-reasons" attribute by  
6622 copying conformance requirements from other sections of the document so that it is clear from  
6623 reading the definition of "job-state-reasons" which values MUST or SHOULD be supported.  
6624 The 'none', 'unsupported-compression', and 'unsupported-document-format' values MUST be  
6625 supported. The 'job-hold-until-specified' SHOULD be specified if the "job-hold-until" Job  
6626 Template is supported. The following values SHOULD be supported: 'job-canceled-by-user',  
6627 'aborted-by-system', and 'job-completed-successfully'. The 'job-canceled-by-operator' SHOULD  
6628 be supported if the implementation permits canceling by other than the job owner. The 'job-  
6629 canceled-at-device' SHOULD be supported if the device supports canceling jobs at the console.  
6630 The 'job-completed-with-warnings' SHOULD be supported, if the implementation detects  
6631 warnings. The 'job-completed-with-errors' SHOULD be supported if the implementation  
6632 detects errors. The 'job-restartable' SHOULD be supported if the Restart-Job operation is  
6633 supported.
  - 6634 12. Section 4.3.10 - allowed a Printer to localize the "job-detailed-status-message" Job Description  
6635 attribute, but indicated that such localization might obscure the technical meaning of such  
6636 messages.
  - 6637 13. Section 4.3.14 - changed the "time-at-creation", "time-at-processing", and "time-at-completed"  
6638 Event Time Job Description attributes from OPTIONAL to REQUIRED.
  - 6639 14. Section 4.3.14.4 - added the REQUIRED "job-printer-up-time (integer(1:MAX))" Job Description  
6640 attribute as an alias for "printer-up-time" to reduce number of operations to get job times.
  - 6641 15. Section 4.4.2 - added the REQUIRED "uri-authentication-supported (1setOf type2 keyword)"  
6642 Printer Description attribute to describe the Client Authentication used by each Printer URI.
  - 6643 16. Section 4.4.12 - changed "printer-state-reasons" Printer Description attribute from OPTIONAL to  
6644 REQUIRED.
  - 6645 17. Section 4.4.12 - changed 'paused' value of "printer-state-reasons" to MUST if Pause-Printer  
6646 operation is supported.

- 6647 18. Section 4.4.14 - added the REQUIRED "ipp-versions-supported (1setOf keyword)" Printer  
6648 Description attribute, since IPP/1.1 Printers do not have to support version '1.0' conformance  
6649 requirements. Section 4.4.16 - added the "multiple-document-jobs-supported (boolean)" Printer  
6650 Description attribute so that a client can tell whether a Printer that supports Create-Job/Send-  
6651 Document supports multiple document jobs or not. This attribute is REQUIRED if the Create-  
6652 Job operation is supported.
- 6653 19. Section 4.4.24 - changed the "queued-job-count" Printer Description attribute from  
6654 RECOMMENDED to REQUIRED.
- 6655 20. Section 4.4.32 - changed "compression-supported (1setOf type3 keyword)" Printer Description  
6656 attribute from OPTIONAL to REQUIRED.
- 6657 21. Section 5.1 - changed the client security requirements from RECOMMENDED non-standards  
6658 track SSL3 to MUST support Client Authentication as defined in the IPP/1.1 Encoding and  
6659 Transport document [IPP-PRO]. A client SHOULD support Operation Privacy and Server  
6660 Authentication as defined in the IPP/1.1 Encoding and Transport document [IPP-PRO].
- 6661 22. Section 5.2.7 - changed the IPP object security requirements from OPTIONAL non-standards track  
6662 SSL3 to SHOULD contain support for Client Authentication as defined in the IPP/1.1 Encoding  
6663 and Transport document [IPP-PRO]. A Printer implementation MAY allow an administrator to  
6664 configure the Printer so that all, some, or none of the users are authenticated. An IPP Printer  
6665 implementation SHOULD contain support for Operation Privacy and Server Authentication as  
6666 defined in the IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer implementation  
6667 MAY allow an administrator to configure the degree of support for Operation Privacy and  
6668 Server Authentication. Security MUST NOT be compromised when the client supplies a lower  
6669 version-number in a request.

6670 See also the "IPP/1.1 Encoding and Transport" [IPP-PRO] document for differences between IPP/1.0  
6671 [RFC2565] and IPP/1.1 [IPP-PRO].

## 6672 18. Full Copyright Statement

6673 Copyright (C) The Internet Society (2000). All Rights Reserved.

6674 This document and translations of it may be copied and furnished to others, and derivative works that  
6675 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and  
6676 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and  
6677 this paragraph are included on all such copies and derivative works. However, this document itself may not  
6678 be modified in any way, such as by removing the copyright notice or references to the Internet Society or  
6679 other Internet organizations, except as needed for the purpose of developing Internet standards in which  
6680 case the procedures for copyrights defined in the Internet Standards process must be followed, or as  
6681 required to translate it into languages other than English.

6682 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its  
6683 successors or assigns.

6684 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET  
6685 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,  
6686 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE  
6687 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED  
6688 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

6689