

1 INTERNET-DRAFT
2 draft-ietf-ipp-implementers-guide-v11-02.txt

T. Hastings
Xerox Corporation
C. Manros
Xerox Corporation
C. Kugler
IBM Printing Systems Co
H. Holst
i-data Printing Systems
P. Zehler
Xerox Corporation
December 1, 2000

Internet Printing Protocol/1.1: Implementer's Guide
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 [RFC2026].
19 Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working
20 groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

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

25 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 Protocol
29 (IPP). IPP is an application level protocol that can be used for distributed printing using Internet tools and
30 technologies. This document contains information that supplements the IPP Model and Semantics [RFC2911] and
31 the IPP Transport and Encoding [RFC2910] documents. It is intended to help implementers understand IPP/1.1,
32 as well as IPP/1.0, and some of the considerations that may assist them in the design of their client and/or IPP
33 object implementations. For example, a typical order of processing requests is given, including error checking.
34 Motivation for some of the specification decisions is also included.

35 The full set of IPP documents includes:

- 36 Design Goals for an Internet Printing Protocol [RFC2567]
- 37 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
- 38 Internet Printing Protocol/1.1: Model and Semantics [RFC2911]
- 39 Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
- 40 Mapping between LPD and IPP Protocols [RFC2569]

41 The document, "Design Goals for an Internet Printing Protocol", takes a broad look at distributed printing
42 functionality, and it enumerates real-life scenarios that help to clarify the features that need to be included in a
43 printing protocol for the Internet. It identifies requirements for three types of users: end users, operators, and
44 administrators. The design goal document calls out a subset of end user requirements that are satisfied in IPP/1.1.
45 Operator and administrator requirements are out of scope for version 1.1.

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

49 The document, "Internet Printing Protocol/1.1: Model and Semantics", describes a simplified model with abstract
50 objects, their attributes, and their operations. The model introduces a Printer and a Job. The Job supports multiple
51 documents per Job. The model document also addresses how security, internationalization, and directory issues
52 are addressed.

53 The document, "Internet Printing Protocol/1.1: Encoding and Transport", is a formal mapping of the abstract
54 operations and attributes defined in the model document onto HTTP/1.1. It also defines the encoding rules for a
55 new Internet media type called "application/ipp".

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

58
59

TABLE OF CONTENTS

59			
60	1	INTRODUCTION.....	6
61	1.1	CONFORMANCE LANGUAGE.....	6
62	1.2	OTHER TERMINOLOGY	6
63	1.3	ISSUES RAISED FROM INTEROPERABILITY BAKE OFFS.....	6
64	2	IPP OBJECTS	7
65	3	IPP OPERATIONS	8
66	3.1	COMMON SEMANTICS.....	8
67	3.1.1	<i>Summary of Operation Attributes.....</i>	8
68	3.1.2	<i>Suggested Operation Processing Steps for IPP Objects</i>	13
69	3.1.2.1	Suggested Operation Processing Steps for all Operations.....	14
70	3.1.2.1.1	Validate version number.....	15
71	3.1.2.1.2	Validate operation identifier.....	16
72	3.1.2.1.3	Validate the request identifier	16
73	3.1.2.1.4	Validate attribute group and attribute presence and order.....	16
74	3.1.2.1.4.1	Validate the presence and order of attribute groups.....	16
75	3.1.2.1.4.2	Ignore unknown attribute groups in the expected position.....	16
76	3.1.2.1.4.3	Validate the presence of a single occurrence of required Operation attributes	17
77	3.1.2.1.5	Validate the values of the REQUIRED Operation attributes	24
78	3.1.2.1.6	Validate the values of the OPTIONAL Operation attributes	27
79	3.1.2.2	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add Documents.....	30
80	3.1.2.2.1	Default "ipp-attribute-fidelity" if not supplied.....	30
81	3.1.2.2.2	Check that the Printer object is accepting jobs.....	31
82	3.1.2.2.3	Validate the values of the Job Template attributes	31
83	3.1.2.3	Algorithm for job validation	31
84	3.1.2.3.1	Check for conflicting Job Template attributes values	36
85	3.1.2.3.2	Decide whether to REJECT the request.....	36
86	3.1.2.3.3	For the Validate-Job operation, RETURN one of the success status codes	38
87	3.1.2.3.4	Create the Job object with attributes to support	38
88	3.1.2.3.5	Return one of the success status codes.....	39
89	3.1.2.3.6	Accept appended Document Content	40
90	3.1.2.3.7	Scheduling and Starting to Process the Job.....	40
91	3.1.2.3.8	Completing the Job.....	40
92	3.1.2.3.9	Destroying the Job after completion	40
93	3.1.2.3.10	Interaction with "ipp-attribute-fidelity".....	40
94	3.1.2.3.11	Character set code conversion support.....	41
95	3.1.2.3.12	What charset to return when an unsupported charset is requested (Issue 1.19)?	42
96	3.1.2.3.13	Natural Language Override (NLO).....	42
97	3.1.3	<i>Status codes returned by operation.....</i>	43
98	3.1.3.1	Printer Operations	43
99	3.1.3.1.1	Print-Job	44
100	3.1.3.1.2	Print-URI.....	45
101	3.1.3.1.3	Validate-Job	46
102	3.1.3.1.4	Create-Job.....	46
103	3.1.3.1.5	Get-Printer-Attributes	46
104	3.1.3.1.6	Get-Jobs.....	47
105	3.1.3.1.7	Pause-Printer.....	47
106	3.1.3.1.8	Resume-Printer	48
107	3.1.3.1.8.1	What about Printers unable to change state due to an error condition?.....	48
108	3.1.3.1.8.2	How is "printer-state" handled on Resume-Printer?	48

109	3.1.3.1.9	Purge-Printer.....	49
110	3.1.3.2	Job Operations	49
111	3.1.3.2.1	Send-Document.....	49
112	3.1.3.2.2	Send-URI	50
113	3.1.3.2.3	Cancel-Job.....	50
114	3.1.3.2.4	Get-Job-Attributes	51
115	3.1.3.2.5	Hold-Job	51
116	3.1.3.2.6	Release-Job	52
117	3.1.3.2.7	Restart-Job.....	52
118	3.1.3.2.7.1	Can documents be added to a restarted job?.....	52
119	3.1.4	Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)	53
120	3.1.5	Sending empty attribute groups.....	53
121	3.2	PRINTER OPERATIONS	54
122	3.2.1	Print-Job operation	54
123	3.2.1.1	Flow controlling the data portion of a Print-Job request (Issue 1.22)	54
124	3.2.1.2	Returning job-state in Print-Job response (Issue 1.30).....	54
125	3.2.2	Get-Printer-Attributes operation	55
126	3.2.3	Get-Jobs operation	55
127	3.2.3.1	Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?	55
128	3.2.3.2	Why is there a "limit" attribute in the Get-Jobs operation?.....	55
129	3.2.4	Create-Job operation	55
130	3.3	JOB OPERATIONS.....	56
131	3.3.1	Validate-Job.....	56
132	3.3.2	Restart-Job	56
133	4	OBJECT ATTRIBUTES	56
134	4.1	ATTRIBUTE SYNTAX'S.....	56
135	4.1.1	The 'none' value for empty sets (Issue 1.37).....	56
136	4.1.2	Multi-valued attributes (Issue 1.31).....	57
137	4.1.3	Case Sensitivity in URIs (issue 1.6).....	57
138	4.1.4	Maximum length for xxxWithLanguage and xxxWithoutLanguage.....	58
139	4.2	JOB TEMPLATE ATTRIBUTES.....	58
140	4.2.1	multiple-document-handling(type2 keyword)	58
141	4.2.1.1	Support of multiple document jobs.....	58
142	4.3	JOB DESCRIPTION ATTRIBUTES.....	58
143	4.3.1	Getting the date and time of day	58
144	4.4	PRINTER DESCRIPTION ATTRIBUTES	59
145	4.4.1	printer-state-reasons (1setOf type2 keyword).....	59
146	4.4.1.1	Is a suffix needed for the "printer-state-reasons" 'none' value (Issue 3.6)?.....	59
147	4.4.2	queued-job-count (integer(0:MAX)).....	59
148	4.4.2.1	Why is "queued-job-count" RECOMMENDED (Issue 1.14)?.....	59
149	4.4.2.2	Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?.....	59
150	4.4.3	printer-current-time (dateTime).....	59
151	4.4.4	Printer-uri.....	60
152	4.5	EMPTY JOBS.....	60
153	5	DIRECTORY CONSIDERATIONS	61
154	5.1	GENERAL DIRECTORY SCHEMA CONSIDERATIONS	61
155	5.2	IPP PRINTER WITH A DNS NAME.....	61
156	6	SECURITY CONSIDERATIONS	61
157	6.1	QUERYING JOBS WITH IPP THAT WERE SUBMITTED USING OTHER JOB SUBMISSION PROTOCOLS (ISSUE 1.32).....	61

158	7	ENCODING AND TRANSPORT	62
159	7.1	GENERAL HEADERS.....	63
160	7.2	REQUEST HEADERS.....	64
161	7.3	RESPONSE HEADERS.....	65
162	7.4	ENTITY HEADERS.....	65
163	7.5	OPTIONAL SUPPORT FOR HTTP/1.0.....	66
164	7.6	HTTP/1.1 CHUNKING	66
165	7.6.1	<i>Disabling IPP Server Response Chunking.....</i>	66
166	7.6.2	<i>Warning About the Support of Chunked Requests</i>	67
167	7.7	HTTP "CONTINUE" INTERIM RESPONSE	67
168	8	REFERENCES	67
169	9	AUTHORS' ADDRESS.....	69
170	10	NOTICES	69
171	11	CHANGE HISTORY (TO BE REMOVED AT TIME OF RFC PUBLISHING).....	70
172	11.1	CHANGES FROM 000509 TO 000530.....	70
173	11.2	CHANGES FROM 990927 TO 000509.....	71
174	11.3	CHANGES FROM 990914 TO 990927.....	71
175	11.4	CHANGES FROM 990726 TO 990914:.....	71
176	11.5	CHANGES TO PRODUCE THE FEBRUARY 12, 1999 VERSION FROM THE JANUARY 8, 1999 VERSION:.....	72
177	11.6	CHANGES TO PRODUCE THE JANUARY 8, 1999 VERSION FROM THE DECEMBER 6, 1998 VERSION:.....	72
178	11.7	CHANGES TO PRODUCE THE DECEMBER 6, 1998 VERSION FROM THE NOVEMBER 16, 1998 VERSION:.....	72
179			
180		TABLES	
181			
182		Table 1 - Summary of Printer operation attributes that sender MUST supply	8
183		Table 2 - Summary of Printer operation attributes that sender MAY supply	9
184		Table 3 - Summary of Job operation attributes that sender MUST supply.....	10
185		Table 4 - Summary of Job operation attributes that sender MAY supply	11
186		Table 5 - Printer operation response attributes.....	12
187		Table 6 - Examples of validating IPP version	15
188		Table 7 - Rules for validating single values X against Z.....	32
189			
190			

190
191
192

193 **1 Introduction**

194 The IPP Implementer's Guide (IIG) (this document) contains information that supplements the IPP Model and
195 Semantics [RFC2911] and the IPP Transport and Encoding [RFC2910] documents. As such this information is
196 not part of the formal specifications. Instead information is presented to help implementers understand the
197 specification, including some of the motivation for decisions taken by the committee in developing the specification.
198 Some of the implementation considerations are intended to help implementers design their client and/or IPP object
199 implementations. If there are any contradictions between this document and [RFC2911] or [RFC2910], those
200 documents take precedence over this document.

201 Platform-specific implementation considerations will be included in this guide as they become known.

202 In order to help the reader of the IIG and the IPP Model and Semantics document, the sections in this document
203 parallel the corresponding sections in the Model document and are numbered the same for ease of cross reference.
204 The sections that correspond to the IPP Transport and Encoding are correspondingly offset.

205 1.1 Conformance language

206 Usually, this document does not contain the terminology MUST, MUST NOT, MAY, NEED NOT, SHOULD,
207 SHOULD NOT, REQUIRED, and OPTIONAL. However, when those terms do appear in this document, their
208 intent is to repeat what the [RFC2911] and [RFC2910] documents require and allow, rather than specifying
209 additional conformance requirements. These terms are defined in section 13 on conformance terminology in
210 [RFC2911], most of which is taken from RFC 2119 [RFC2119].

211 Implementers should read section 13 (APPENDIX A) in [RFC2911] in order to understand these capitalized
212 words. The words MUST, MUST NOT, and REQUIRED indicate what implementations are required to support
213 in a client or IPP object in order to be conformant to [RFC2911] and [RFC2910]. MAY, NEED NOT, and
214 OPTIONAL indicate was is merely allowed as an implementer option. The verbs SHOULD and SHOULD NOT
215 indicate suggested behavior, but which is not required or disallowed, respectively, in order to conform to the
216 specification.

217 1.2 Other terminology

218 The term "sender" refers to the client that sends a request or an IPP object that returns a response. The term
219 "receiver" refers to the IPP object that receives a request and to a client that receives a response.

220 1.3 Issues Raised from Interoperability Bake Offs

221 The IPP WG has conducted three open interoperability "Bake Offs". The first bake off was held in September
222 1998, Bake Off2 was held in March 1999, and Bake Off3 was held in October 2000. See the summary reports
223 in:

224 ftp://ftp.pwg.org/pub/pwg/ipp/new_TES/

225 The issues raised from the first bake off are numbered 1.n in this document and are described in:

226 <ftp://ftp.pwg.org/pub/pwg/ipp/approved-clarifications/ipp-agreed-fixes-981030.pdf>

227 These issue resolutions have been incorporated into the November 16, "IPP/1.0 Model and Semantics"
228 [RFC2566] and the "IPP/1.0 Encoding and Transport" [RFC2565] documents. However, some of the discussion
229 is left here in the Implementer's Guide to help understanding.

230 The issues raised from Bake Off2 are numbered 2.n in this document and are described in:

231 <ftp://ftp.pwg.org/pub/pwg/ipp/issues/issues-raised-at-bake-off2.pdf>

232 The issues raised from Bake Off3 are number 3.n in this document and are described in:

233 <ftp://ftp.pwg.org/pub/pwg/ipp/issues/issues-raised-at-bake-off3.pdf>

234 **2 IPP Objects**

235 The term "client" in IPP is intended to mean any client that issues IPP operation requests and accepts IPP operation
236 responses, whether it be a desktop or a server. In other words, the term "client" does not just mean end-user
237 clients, such as those associated with desktops.

238 The term "IPP Printer" in IPP is intended to mean an object that accepts IPP operation requests and returns IPP
239 operation responses, whether implemented in a server or a device. An IPP Printer object MAY, if implemented in
240 a server, turn around and forward received jobs (and other requests) to other devices and print servers/services,
241 either using IPP or some other protocol.

242

3 IPP Operations

This section corresponds to Section 3 "IPP Operations" in the IPP/1.1 Model and Semantics document [RFC2911].

3.1 Common Semantics

This section discusses semantics common to all operations.

3.1.1 Summary of Operation Attributes

Legend for the following table:

R indicates a REQUIRED operation that MUST be supported by the IPP object (Printer or Job). For attributes, R indicates that the attribute MUST be supported by the IPP object supports the associated operation.

O indicates an OPTIONAL operation or attribute that MAY be supported by the IPP object (Printer or Job).

+ indicates that this is not an IPP/1.0 feature, but is only a part of IPP/1.1 and future versions of IPP.

Table 1 - Summary of Printer operation attributes that sender MUST supply

Operation Attributes	Printer Operations						
	Requests						Responses
	Print-Job, Validate-Job (R)	Print-URI (O)	Create-Job (O)	Get-Printer-Attributes (R)	Get-Jobs (R)	Pause-Printer, Resume-Printer, Purge-Printer (O+)	All Operations
Operation parameters--REQUIRED to be supplied by the sender:							
operation-id	R	R	R	R	R	R	
status-code							R
request-id	R	R	R	R	R	R	R
version-number	R	R	R	R	R	R	R
Operation attributes--REQUIRED to be supplied by the sender:							
attributes-charset	R	R	R	R	R	R	R
attributes-natural-language	R	R	R	R	R	R	R
document-uri		R					
job-id*							
job-uri*							
last-document							
printer-uri	R	R	R	R	R	R	
Operation attributes--RECOMMENDED to be supplied by the sender:							

Operation Attributes	Printer Operations						
	Requests						Responses
	Print-Job, Validate-Job (R)	Print-URI (O)	Create-Job (O)	Get-Printer-Attributes (R)	Get-Jobs (R)	Pause-Printer, Resume-Printer, Purge-Printer (O+)	All Operations
job-name	R	R	R				
requesting-user-name	R	R	R	R	R	R	

Table 2 - Summary of Printer operation attributes that sender MAY supply

Operation Attributes	Printer Operations						
	Requests						Responses
	Print-Job, Validate-Job (R)	Print-URI (O)	Create-Job (O)	Get-Printer-Attributes (R)	Get-Jobs (R)	Pause-Printer, Resume-Printer, Purge-Printer (O+)	All Operations
Operation attributes--OPTIONAL to be supplied by the sender:							
status-message							O
detailed-status-message							O
document-access-error							O**
compression	O	O					
document-format	R	R		R			
document-name	O	O					
document-natural-language	O	O					
ipp-attribute-fidelity	R	R	R				
job-impressions	O	O	O				
job-k-octets	O	O	O				
job-media-sheets	O	O	O				
limit					R		
message							
my-jobs					R		
requested-attributes				R	R		
which-jobs					R		

* "job-id" is REQUIRED only if used together with "printer-uri" to identify the target job; otherwise, "job-uri" is REQUIRED.

** "document-access-error" applies to the Print-URI response only.

Table 3 - Summary of Job operation attributes that sender MUST supply

Operation Attributes	Job Operations					
	Requests					Responses
	Send-Document (O)	Send-URI (O)	Cancel-Job (R)	Get-Job-Attributes (R)	Hold-Job, Release-Job, Restart-Job (O+)	All Operations
Operation parameters--REQUIRED to be supplied by the sender:						
operation-id	R	R	R	R	R	
status-code						R
request-id	R	R	R	R	R	R
version-number	R	R	R	R	R	R
Operation attributes--REQUIRED to be supplied by the sender:						
attributes-charset	R	R	R	R	R	R
attributes-natural-language	R	R	R	R	R	R
document-uri		R				
job-id*	R	R	R	R	R	
job-uri*	R	R	R	R	R	
last-document	R	R				
printer-uri	R	R	R	R	R	
Operation attributes--RECOMMENDED to be supplied by the sender:						
job-name						
requesting-user-name	R	R	R	R	R	

Table 4 - Summary of Job operation attributes that sender MAY supply

Operation Attributes	Job Operations						
	Requests						Responses
	Send-Document (O)	Send-URI (O)	Cancel-Job (R)	Get-Job-Attributes (R)	Hold-Job, Restart-Job (O+)	Release-Job (O+)	All Operations
Operation attributes--OPTIONAL to be supplied by the sender:							
status-message							O
detailed-status-message							O
document-access-error							O**
compression	O	O					
document-format	R	R					
document-name	O	O					
document-natural-language	O	O					
ipp-attribute-fidelity							
job-impressions							
job-k-octets							
job-media-sheets							
limit							
message			O		O	O	
job-hold-until					R		
my-jobs							
requested-attributes				R			
which-jobs							

* "job-id" is REQUIRED only if used together with "printer-uri" to identify the target job; otherwise, "job-uri" is REQUIRED.

** "document-access-error" applies to the Send-URI operation only.

263

264

Table 5 - Printer operation response attributes

	Printer Operations						
	Response						
Operation Attributes	Print-Job (R), Send-Document (O)	Validate -Job (R)	Print -URI (O), Send-URI (O)	Create-Job (O)	Get-Printer - Attributes (R)	Get-Jobs (R)	Pause-Printer, Resume-Printer, Purge-Printer (O+)
job-uri	R		R	R			
job-id	R		R	R			
job-state	R		R	R			
job-state-reasons	R+		R+	R+			
number-of-intervening-jobs	O		O	O			
document-access-error+			O				

265

266

266

267 3.1.2 Suggested Operation Processing Steps for IPP Objects

268 This section suggests the steps and error checks that an IPP object MAY perform when processing requests and
269 returning responses. An IPP object MAY perform some or all of the error checks. However, some
270 implementations MAY choose to be more forgiving than the error checks shown here, in order to be able to accept
271 requests from non-conforming clients. Not performing all of these error checks is a so-called "forgiving"
272 implementation. On the other hand, clients that successfully submit requests to IPP objects that do perform all the
273 error checks will be more likely to be able to interoperate with other IPP object implementations. Thus an
274 implementer of an IPP object needs to decide whether to be a "forgiving" or a "strict" implementation. Therefore,
275 the error status codes returned may differ between implementations. Consequentially, client SHOULD NOT
276 expect exactly the error code processing described in this section.

277 When an IPP object receives a request, the IPP object either accepts or rejects the request. In order to determine
278 whether or not to accept or reject the request, the IPP object SHOULD execute the following steps. The order of
279 the steps may be rearranged and/or combined, including making one or multiple passes over the request.

280 A client MUST supply requests that would pass all of the error checks indicated here in order to be a conforming
281 client. Therefore, a client SHOULD supply requests that are conforming, in order to avoid being rejected by some
282 IPP object implementations and/or risking different semantics by different implementations of forgiving
283 implementations. For example, a forgiving implementation that accepts multiple occurrences of the same attribute,
284 rather than rejecting the request might use the first occurrences, while another might use the last occurrence. Thus
285 such a non-conforming client would get different results from the two forgiving implementations.

286 In the following, processing continues step by step until a "RETURNS the xxx status code ..." statement is
287 encountered. Error returns are indicated by the verb: "REJECTS". Since clients have difficulty getting the status
288 code before sending all of the document data in a Print-Job request, clients SHOULD use the Validate-Job
289 operation before sending large documents to be printed, in order to validate whether the IPP Printer will accept the
290 job or not.

291 It is assumed that security authentication and authorization has already taken place at a lower layer.

292

292 3.1.2.1 Suggested Operation Processing Steps for all Operations

293 This section is intended to apply to all operations. The next section contains the additional steps for the Print-Job,
 294 Validate-Job, Print-URI, Create-Job, Send-Document, and Send-URI operations that create jobs, adds
 295 documents, and validates jobs.

296 IIG Sect #	Flow	IPP error status codes
297 -----	----	-----
298		
299	v	err
300 3.1.2.1.1 <Validate version>	-->	server-error-version-not-
301 supported		
302	ok	
303	v	err
304 3.1.2.1.2 <Validate operation>	-->	server-error-operation-not-
305 supported		
306	ok	
307	v	err
308 3.1.2.1.4.1- <Validate presence>	-->	client-error-bad-request
309 3.1.2.1.4.2 <of attributes>		
310	ok	
311	v	err
312 3.1.2.1.4.3 <Validate presence>	-->	client-error-bad-request
313 <of operation attr>		
314	ok	
315	v	err
316 3.1.2.1.5 <Valied values of>	-->	client-error-bad-request
317 <operation attrs>		client-error-request-value-too-
318 long		
319 <(length, tag, range,>		
320 <multi-value)>		
321	ok	
322	v	err
323 3.1.2.1.5 <Validate values>	-->	client-error-bad-request
324 <with supported values>		client-error-charset-not-
325 supported		
326	ok	client-error-attributes-or-
327 values-		
328		not-supported
329	v	err
330 3.1.2.1.6 <Validate optionally>	-->	client-error-bad-request
331 <operation attr>		client-error-natural-language-
332 not-		
333	ok	supported
334		client-error-request-value-too-
335 long		

336 | client-error-attributes-or-
 337 values-
 338 v not-supported
 339

340 3.1.2.1.1 Validate version number

341 Every request and every response contains the "version-number" attribute. The value of this attribute is the major
 342 and minor version number of the syntax and semantics that the client and IPP object is using, respectively. The
 343 "version-number" attribute remains in a fixed position across all future versions so that all clients and IPP object that
 344 support future versions can determine which version is being used. The IPP object checks to see if the major
 345 version number supplied in the request is supported. If not, the Printer object REJECTS the request and
 346 RETURNS the 'server-error-version-not-supported' status code in the response. The IPP object returns in the
 347 "version-number" response attribute the major and minor version for the error response. Thus the client can learn
 348 at least one major and minor version that the IPP object supports. The IPP object is encouraged to return the
 349 closest version number to the one supplied by the client.

350 The checking of the minor version number is implementation dependent, however if the client supplied minor
 351 version is explicitly supported, the IPP object MUST respond using that identical minor version number. If the
 352 major version number matches, but the minor version number does not, the Printer SHOULD accept and attempt
 353 to process the request, or MAY reject the request and return the 'server-error-version-not-supported' status code.
 354 In all cases, the Printer MUST return the nearest version number that it supports. For example, suppose that an
 355 IPP/1.2 Printer supports versions '1.1' and '1.2'. The following responses are conforming:

356 **Table 6 - Examples of validating IPP version**

Client supplies	Printer Accept Request?	Printer returns
1.0	yes (SHOULD)	1.1
	no (SHOULD NOT)	1.1
1.1	yes (MUST)	1.1
1.2	yes (MUST)	1.2
1.3	yes (SHOULD)	1.2
	no (SHOULD NOT)	1.2

357

358 It is advantageous for Printers to support both IPP/1.1 and IPP/1.0, so that they can interoperate with either client
 359 implementations. Some implementations may allow an Administrator to explicitly disable support for one or the
 360 other by setting the "ipp-versions-supported" Printer description attribute.

361 Likewise, it is advantageous for clients to support both versions to allow interoperability with new and legacy
362 Printers.

363 3.1.2.1.2 Validate operation identifier

364 The Printer object checks to see if the "operation-id" attribute supplied by the client is supported as indicated in the
365 Printer object's "operations-supported" attribute. If not, the Printer REJECTS the request and returns the 'server-
366 error-operation-not-supported' status code in the response.

367 3.1.2.1.3 Validate the request identifier

368 The Printer object SHOULD NOT check to see if the "request-id" attribute supplied by the client is in range:
369 between 1 and $2^{31} - 1$ (inclusive), but copies all 32 bits.

370 Note: The "version-number", "operation-id", and the "request-id" parameters are in fixed octet positions in the
371 IPP/1.1 encoding. The "version-number" parameter will be the same fixed octet position in all versions of the
372 protocol. These fields are validated before proceeding with the rest of the validation.

373 3.1.2.1.4 Validate attribute group and attribute presence and order

374 The order of the following validation steps depends on implementation.

375 3.1.2.1.4.1 Validate the presence and order of attribute groups

376 Client requests and IPP object responses contain attribute groups that Section 3 requires to be present and in a
377 specified order. An IPP object verifies that the attribute groups are present and in the correct order in requests
378 supplied by clients (attribute groups without an * in the following tables).

379 If an IPP object receives a request with (1) required attribute groups missing, or (2) the attributes groups are out of
380 order, or (3) the groups are repeated, the IPP object REJECTS the request and RETURNS the 'client-error-bad-
381 request' status code. For example, it is an error for the Job Template Attributes group to occur before the
382 Operation Attributes group, for the Operation Attributes group to be omitted, or for an attribute group to occur
383 more than once, except in the Get-Jobs response.

384 Since this kind of attribute group error is most likely to be an error detected by a client developer rather than by a
385 customer, the IPP object NEED NOT return an indication of which attribute group was in error in either the
386 Unsupported Attributes group or the Status Message. Also, the IPP object NEED NOT find all attribute group
387 errors before returning this error.

388 3.1.2.1.4.2 Ignore unknown attribute groups in the expected position

389 Future attribute groups may be added to the specification at the end of requests just before the Document Content
390 and at the end of response, except for the Get-Jobs response, where it maybe there or before the first job
391 attributes returned. If an IPP object receives an unknown attribute group in these positions, it ignores the entire
392 group, rather than returning an error, since that group may be a new group in a later minor version of the protocol
393 that can be ignored. (If the new attribute group cannot be ignored without confusing the client, the major version
394 number would have been increased in the protocol document and in the request). If the unknown group occurs in a
395 different position, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' status code.

396 Clients also ignore unknown attribute groups returned in a response.

397 Note: By validating that requests are in the proper form, IPP objects force clients to use the proper form which, in
398 turn, increases the chances that customers will be able to use such clients from multiple vendors with IPP objects
399 from other vendors.

400 3.1.2.1.4.3 Validate the presence of a single occurrence of required Operation attributes
401 Client requests and IPP object responses contain Operation attributes that [RFC2911] Section 3 requires to be
402 present. Attributes within a group may be in any order, except for the ordering of target, charset, and natural
403 languages attributes. These attributes MUST be first, and MUST be supplied in the following order: charset,
404 natural language, and then target. An IPP object verifies that the attributes that Section 4 requires to be supplied by
405 the client have been supplied in the request (attributes without an * in the following tables). An asterisk (*)
406 indicates groups and Operation attributes that the client may omit in a request or an IPP object may omit in a
407 response.

408 If an IPP object receives a request with required attributes missing or repeated from a group or in the wrong
409 position, the behavior of the IPP object is IMPLEMENTATION DEPENDENT. Some of the possible
410 implementations are:

- 411 1. REJECTS the request and RETURNS the 'client-error-bad-request' status code
- 412 2. accepts the request and uses the first occurrence of the attribute no matter where it is
- 413 3. accepts the request and uses the last occurrence of the attribute no matter where it is
- 414 4. accept the request and assume some default value for the missing attribute

415 Therefore, client MUST send conforming requests, if they want to receive the same behavior from all IPP object
416 implementations. For example, it is an error for the "attributes-charset" or "attributes-natural-language" attribute to
417 be omitted in any operation request, or for an Operation attribute to be supplied in a Job Template group or a Job
418 Template attribute to be supplied in an Operation Attribute group in a create request. It is also an error to supply
419 the "attributes-charset" attribute twice.

420 Since these kinds of attribute errors are most likely to be detected by a client developer rather than by a customer,
421 the IPP object NEED NOT return an indication of which attribute was in error in either the Unsupported Attributes
422 group or the Status Message. Also, the IPP object NEED NOT find all attribute errors before returning this error.

423 The following tables list all the attributes for all the operations by attribute group in each request and each response.
424 The order of the groups is the order that the client supplies the groups as specified in [RFC2911] Section 3. The
425 order of the attributes within a group is arbitrary, except as noted for some of the special operation attributes
426 (charset, natural language, and target). The tables below use the following notation:

- 427 R indicates a REQUIRED attribute or operation that an IPP object MUST support
- 428 O indicates an OPTIONAL attribute or operation that an IPP object NEED NOT support
- 429 * indicates that a client MAY omit the attribute in a request and that an IPP object MAY omit the attribute in
430 a response. The absence of an * means that a client MUST supply the attribute in a request
431 and an IPP object MUST supply the attribute in a response.
- 432 + indicates that this is not a IPP/1.0 operation, but is only a part of IPP/1.1 and future versions of IPP.

433

434 Operation Requests

435 The tables below show the attributes in their proper attribute groups for operation requests:

436 Note: All operation requests contain "version-number", "operation-
437 id", and "request-id" parameters.

438

439 Print-Job Request (R):

440 Group 1: Operation Attributes (R)

441 attributes-charset (R)

442 attributes-natural-language (R)

443 printer-uri (R)

444 requesting-user-name (R*)

445 job-name (R*)

446 ipp-attribute-fidelity (R*)

447 document-name (R*)

448 document-format (R*)

449 document-natural-language (O*)

450 compression (O*)

451 job-k-octets (O*)

452 job-impressions (O*)

453 job-media-sheets (O*)

454 Group 2: Job Template Attributes (R*)

455 <Job Template attributes> (O*)

456 (see [RFC2911] Section 4.2)

457 Group 3: Document Content (R)

458 <document content>

459

```
460 Validate-Job Request (R):
461     Group 1: Operation Attributes (R)
462         attributes-charset (R)
463         attributes-natural-language (R)
464         printer-uri (R)
465         requesting-user-name (R*)
466         job-name (R*)
467         ipp-attribute-fidelity (R*)
468         document-name (R*)
469         document-format (R*)
470         document-natural-language (O*)
471         compression (O*)
472         job-k-octets (O*)
473         job-impressions (O*)
474         job-media-sheets (O*)
475     Group 2: Job Template Attributes (R*)
476         <Job Template attributes> (O*)
477         (see [RFC2911] Section 4.2)
478
479 Print-URI Request (O):
480     Group 1: Operation Attributes (R)
481         attributes-charset (R)
482         attributes-natural-language (R)
483         printer-uri (R)
484         document-uri (R)
485         requesting-user-name (R*)
486         job-name (R*)
487         ipp-attribute-fidelity (R*)
488         document-name (R*)
489         document-format (R*)
490         document-natural-language (O*)
491         compression (O*)
492         job-k-octets (O*)
493         job-impressions (O*)
494         job-media-sheets (O*)
495     Group 2: Job Template Attributes (R*)
496         <Job Template attributes> (O*) (see
497         (see [RFC2911] Section 4.2)
498
499 Create-Job Request (O):
500     Group 1: Operation Attributes (R)
501         attributes-charset (R)
502         attributes-natural-language (R)
503         printer-uri (R)
504         requesting-user-name (R*)
505         job-name (R*)
506         ipp-attribute-fidelity (R*)
```

```
507         job-k-octets (O*)
508         job-impressions (O*)
509         job-media-sheets (O*)
510     Group 2: Job Template Attributes (R*)
511         <Job Template attributes> (O*) (see
512             (see [RFC2911] Section 4.2)
513
514 Get-Printer-Attributes Request (R):
515     Group 1: Operation Attributes (R)
516         attributes-charset (R)
517         attributes-natural-language (R)
518         printer-uri (R)
519         requesting-user-name (R*)
520         requested-attributes (R*)
521         document-format (R*)
522
523 Get-Jobs Request (R):
524     Group 1: Operation Attributes (R)
525         attributes-charset (R)
526         attributes-natural-language (R)
527         printer-uri (R)
528         requesting-user-name (R*)
529         limit (R*)
530         requested-attributes (R*)
531         which-jobs (R*)
532         my-jobs (R*)
533
534 Send-Document Request (O):
535     Group 1: Operation Attributes (R)
536         attributes-charset (R)
537         attributes-natural-language (R)
538         (printer-uri & job-id) | job-uri (R)
539         last-document (R)
540         requesting-user-name (R*)
541         document-name (R*)
542         document-format (R*)
543         document-natural-language (O*)
544         compression (O*)
545     Group 2: Document Content (R*)
546         <document content>
547
548 Send-URI Request (O):
549     Group 1: Operation Attributes (R)
550         attributes-charset (R)
551         attributes-natural-language (R)
552         (printer-uri & job-id) | job-uri (R)
553         last-document (R)
```

```

554         document-uri (R)
555         requesting-user-name (R*)
556         document-name (R*)
557         document-format (R*)
558         document-natural-language (O*)
559         compression (O*)
560
561 Cancel-Job Request (R):
562 Release-Job Request (O+):
563     Group 1: Operation Attributes (R)
564         attributes-charset (R)
565         attributes-natural-language (R)
566         (printer-uri & job-id) | job-uri (R)
567         requesting-user-name (R*)
568         message (O*)
569
570 Get-Job-Attributes Request (R):
571     Group 1: Operation Attributes (R)
572         attributes-charset (R)
573         attributes-natural-language (R)
574         (printer-uri & job-id) | job-uri (R)
575         requesting-user-name (R*)
576         requested-attributes (R*)
577
578 Pause-Printer Request (O+):
579 Resume-Printer Request (O+):
580 Purge-Printer Request (O+):
581     Group 1: Operation Attributes (R)
582         attributes-charset (R)
583         attributes-natural-language (R)
584         printer-uri (R)
585         requesting-user-name (R*)
586
587 Hold-Job Request (O+):
588 Restart-Job Request (O+):
589     Group 1: Operation Attributes (R)
590         attributes-charset (R)
591         attributes-natural-language (R)
592         (printer-uri & job-id) | job-uri (R)
593         requesting-user-name (R*)
594         job-hold-until (R*)
595         message (O*)
596

```

597 Operation Responses

598 The tables below show the response attributes in their proper attribute groups for responses.

599 Note: All operation responses contain "version-number", "status-
600 code", and "request-id" parameters.

601

602 Print-Job Response (R):

603 Create-Job Response (O):

604 Send-Document Response (O):

605 Group 1: Operation Attributes (R)

606 attributes-charset (R)

607 attributes-natural-language (R)

608 status-message (O*)

609 detailed-status-message (O*)

610 Group 2: Unsupported Attributes (R*) (see Note 3)

611 <unsupported attributes> (R*)

612 Group 3: Job Object Attributes(R*) (see Note 2)

613 job-uri (R)

614 job-id (R)

615 job-state (R)

616 job-state-reasons (O* | R+)

617 job-state-message (O*)

618 number-of-intervening-jobs (O*)

619

620 Validate-Job Response (R):

621 Cancel-Job Response (R):

622 Hold-Job Response (O+):

623 Release-Job Response (O+):

624 Restart-Job Response (O+):

625 Group 1: Operation Attributes (R)

626 attributes-charset (R)

627 attributes-natural-language (R)

628 status-message (O*)

629 detailed-status-message (O*)

630 Group 2: Unsupported Attributes (R*) (see Note 3)

631 <unsupported attributes> (R*)

632

633 Print-URI Response (O):

634 Send-URI Response (O):

635 Group 1: Operation Attributes (R)

636 attributes-charset (R)

637 attributes-natural-language (R)

638 status-message (O*)

639 detailed-status-message (O*)

640 document-access-error (O*)

641 Group 2: Unsupported Attributes (R*) (see Note 3)

642 <unsupported attributes> (R*)

643 Group 3: Job Object Attributes(R*) (see Note 2)

644 job-uri (R)

```
645         job-id (R)
646         job-state (R)
647         job-state-reasons (O* | R+)
648         job-state-message (O*)
649         number-of-intervening-jobs (O*)
650
651 Get-Printer-Attributes Response (R):
652     Group 1: Operation Attributes (R)
653         attributes-charset (R)
654         attributes-natural-language (R)
655         status-message (O*)
656         detailed-status-message (O*)
657     Group 2: Unsupported Attributes (R*) (see Note 4)
658         <unsupported attributes> (R*)
659     Group 3: Printer Object Attributes(R*) (see Note 2)
660         <requested attributes> (R*)
661
662 Get-Jobs Response (R):
663     Group 1: Operation Attributes (R)
664         attributes-charset (R)
665         attributes-natural-language (R)
666         status-message (O*)
667         detailed-status-message (O*)
668     Group 2: Unsupported Attributes (R*) (see Note 4)
669         <unsupported attributes> (R*)
670     Group 3: Job Object Attributes(R*) (see Note 2, 5)
671         <requested attributes> (R*)
672
673 Get-Job-Attributes Response (R):
674     Group 1: Operation Attributes (R)
675         attributes-charset (R)
676         attributes-natural-language (R)
677         status-message (O*)
678         detailed-status-message (O*)
679     Group 2: Unsupported Attributes (R*) (see Note 4)
680         <unsupported attributes> (R*)
681     Group 3: Job Object Attributes(R*) (see Note 2)
682         <requested attributes> (R*)
683
684 Pause-Printer Response (O+):
685 Resume-Printer Response (O+):
686 Purge-Printer Response (O+):
687     Group 1: Operation Attributes (R)
688         attributes-charset (R)
689         attributes-natural-language (R)
690         status-message (O*)
691         detailed-status-message (O*)
```

692 Group 2: Unsupported Attributes (R*) (see Note 4)
693 <unsupported attributes> (R*)
694

695 Note 2 - the Job Object Attributes and Printer Object Attributes are returned only if the IPP object returns one of
696 the success status codes.

697 Note 3 - the Unsupported Attributes Group is present only if the client included some Operation and/or Job
698 Template attributes or values that the Printer doesn't support whether a success or an error return.

699 Note 4 - the Unsupported Attributes Group is present only if the client included some Operation attributes that the
700 Printer doesn't support whether a success or an error return.

701 Note 5: for the Get-Jobs operation the response contains a separate Job Object Attributes group 3 to N
702 containing requested-attributes for each job object in the response.

703 3.1.2.1.5 Validate the values of the REQUIRED Operation attributes

704 An IPP object validates the values supplied by the client of the REQUIRED Operation attribute that the IPP object
705 MUST support. The next section specifies the validation of the values of the OPTIONAL Operation attributes that
706 IPP objects MAY support.

707 The IPP object performs the following syntactic validation checks of each Operation attribute value:

- 708 a) that the length of each Operation attribute value is correct for the attribute syntax tag supplied by the
709 client according to [RFC2911] Section 4.1,
- 710 b) that the attribute syntax tag is correct for that Operation attribute according to [RFC2911] Section 3,
- 711 c) that the value is in the range specified for that Operation attribute according to [RFC2911] Section 3,
- 712 d) that multiple values are supplied by the client only for operation attributes that are multi-valued, i.e., that
713 are 1setOf X according to [RFC2911] Section 3.

714

715 If any of these checks fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' or
716 the 'client-error-request-value-too-long' status code. Since such an error is most likely to be an error detected by
717 a client developer, rather than by an end-user, the IPP object NEED NOT return an indication of which attribute
718 had the error in either the Unsupported Attributes Group or the Status Message. The description for each of these
719 syntactic checks is explicitly expressed in the first IF statement in the following table.

720 In addition, the IPP object checks each Operation attribute value against some Printer object attribute or some
721 hard-coded value if there is no "xxx-supported" Printer object attribute defined. If its value is not among those
722 supported or is not in the range supported, then the IPP object REJECTS the request and RETURNS the error

723 status code indicated in the table by the second IF statement. If the value of the Printer object's "xxx-supported"
724 attribute is 'no-value' (because the system administrator hasn't configured a value), the check always fails.

725 -----

726 attributes-charset (charset)

727 IF NOT a single non-empty 'charset' value, REJECT/RETURN 'client-error-bad-request'.
728 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-long'.
729 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-charset-not-
730 supported".
731

732 attributes-natural-language(naturalLanguage)

733 IF NOT a single non-empty 'naturalLanguage' value, REJECT/RETURN 'client-error-bad-request'.
734 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-long'.
735 ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-language-
736 supported" attribute. If the supplied value is not a member of the Printer object's "generated-
737 natural-language-supported" attribute, use the Printer object's "natural-language-configured" value.
738

739 requesting-user-name

740 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
741 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
742 IF the IPP object can obtain a better-authenticated name, use it instead.
743

744 job-name(name)

745 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
746 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
747 IF NOT supplied by the client, the Printer object creates a name from the document-name or document-
748 uri.
749

750 document-name (name)

751 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
752 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
753

754 ipp-attribute-fidelity (boolean)

755 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-
756 request'.
757 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'.
758 IF NOT supplied by the client, the IPP object assumes the value 'false'.

759

760 document-format (mimeType)

761 IF NOT a single non-empty 'mimeType' value, REJECT/RETURN 'client-error-bad-request'.

762 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.

763 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN 'client-error-
764 document-format-not-supported'765 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's "document-format-
766 default" attribute.

767

768 document-uri (uri)

769 IF NOT a single non-empty 'uri' value, REJECT/RETURN 'client-error-bad-request'.

770 IF the value length is greater than 1023 octets, REJECT/RETURN 'client-error-request-value-too-long'.

771 IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.

772 If the client-supplied URI scheme is not supported, i.e. the value is not in the Printer object's referenced-
773 uri-scheme-supported" attribute, the Printer object MUST reject the request and return the 'client-
774 error-uri-scheme-not-supported' status code. The Printer object MAY check to see if the
775 document exists and is accessible. If the document is not found or is not accessible,
776 REJECT/RETURN 'client-error-not found'.

777 last-document (boolean)

778 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-
779 request'.

780 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'

781

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

783 IF NOT an single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN
784 'client-error-bad-request'.785 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-error-
786 gone' status code, if keep track of recently deleted jobs.

787

788 requested-attributes (1setOf keyword)

789 IF NOT one or more 'keyword' values, REJECT/RETURN 'client-error-bad-request'.

790 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.

791 Ignore unsupported values, which are the keyword names of unsupported attributes. Don't bother to copy
792 such requested (unsupported) attributes to the Unsupported Attribute response group since the
793 response will not return them.

794

795 which-jobs (type2 keyword)

796 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.

797 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
798 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to the
799 Unsupported Attributes response group and REJECT/RETURN 'client-error-attributes-or-values-
800 not-supported'.
801 Note: a Printer still supports the 'completed' value even if it keeps no completed/canceled/aborted jobs: by
802 returning no jobs when so queried.
803 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.
804
805 my-jobs (boolean)
806 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-
807 request'.
808 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'
809 IF NOT supplied by the client, the IPP object assumes the 'false' value.
810
811 limit (integer(1:MAX))
812 IF NOT a single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN 'client-
813 error-bad-request'.
814 IF NOT supplied by the client, the IPP object returns all jobs, no matter how many.
815
816 -----
817
818 3.1.2.1.6 Validate the values of the OPTIONAL Operation attributes
819 OPTIONAL Operation attributes are those that an IPP object MAY or MAY NOT support. An IPP object
820 validates the values of the OPTIONAL attributes supplied by the client. The IPP object performs the same
821 syntactic validation checks for each OPTIONAL attribute value as in Section 3.1.2.1.5. As in Section 3.1.2.1.5, if
822 any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' or the 'client-error-
823 request-value-too-long' status code.
824 In addition, the IPP object checks each Operation attribute value against some Printer attribute or some hard-
825 coded value if there is no "xxx-supported" Printer attribute defined. If its value is not among those supported or is
826 not in the range supported, then the IPP object REJECTS the request and RETURNS the error status code
827 indicated in the table. If the value of the Printer object's "xxx-supported" attribute is 'no-value' (because the system
828 administrator hasn't configured a value), the check always fails.
829 If the IPP object doesn't recognize/support an attribute, the IPP object treats the attribute as an unknown or
830 unsupported attribute (see the last row in the table below).
831 -----

832 document-natural-language (naturalLanguage)

833 IF NOT a single non-empty 'naturalLanguage' value, REJECT/RETURN 'client-error-bad-request'.

834 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-long'.

835 IF NOT a value that the Printer object supports in document formats, (no corresponding "xxx-supported"
836 Printer attribute), REJECT/RETURN 'client-error-natural-language-not-supported'.

837

838 compression (type3 keyword)

839 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.

840 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.

841 IF NOT in the Printer object's "compression-supported" attribute, copy the attribute and the unsupported
842 value to the Unsupported Attributes response group and REJECT/RETURN 'client-error-
843 attributes-or-values-not-supported'.

844 Note to IPP/1.0 implementers: Support for the "compression" attribute was optional in IPP/1.0 and was
845 changed to REQUIRED in IPP/1.1. However, an IPP/1.0 object SHOULD at least check for the
846 "compression" attribute being present and reject the create request, if they don't support
847 "compression". Not checking is a bug, since the data will be unintelligible.

848

849 job-k-octets (integer(0:MAX))

850 IF NOT a single 'integer' value equal to 4 octets,

851 REJECT/RETURN 'client-error-bad-request'.

852 IF NOT in the range of the Printer object's "job-k-octets-supported" attribute, copy the attribute and the
853 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-
854 error-attributes-or-values-not-supported'.

855

856 job-impressions (integer(0:MAX))

857 IF NOT a single 'integer' value equal to 4 octets,

858 REJECT/RETURN 'client-error-bad-request'.

859 IF NOT in the range of the Printer object's "job-impressions-supported" attribute, copy the attribute and
860 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN
861 'client-error-attributes-or-values-not-supported'.

862

863 job-media-sheets (integer(0:MAX))

864 IF NOT a single 'integer' value equal to 4 octets,

865 REJECT/RETURN 'client-error-bad-request'.

866 IF NOT in the range of the Printer object's "job-media-sheets-supported" attribute, copy the attribute and
867 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN
868 'client-error-attributes-or-values-not-supported'.

869

870 message (text(127))

871 IF NOT a single 'text' value, REJECT/RETURN 'client-error-bad-request'.
872 IF the value length is greater than 127 octets,
873 REJECT/RETURN 'client-error-request-value-too-long'.
874

875 unknown or unsupported attribute

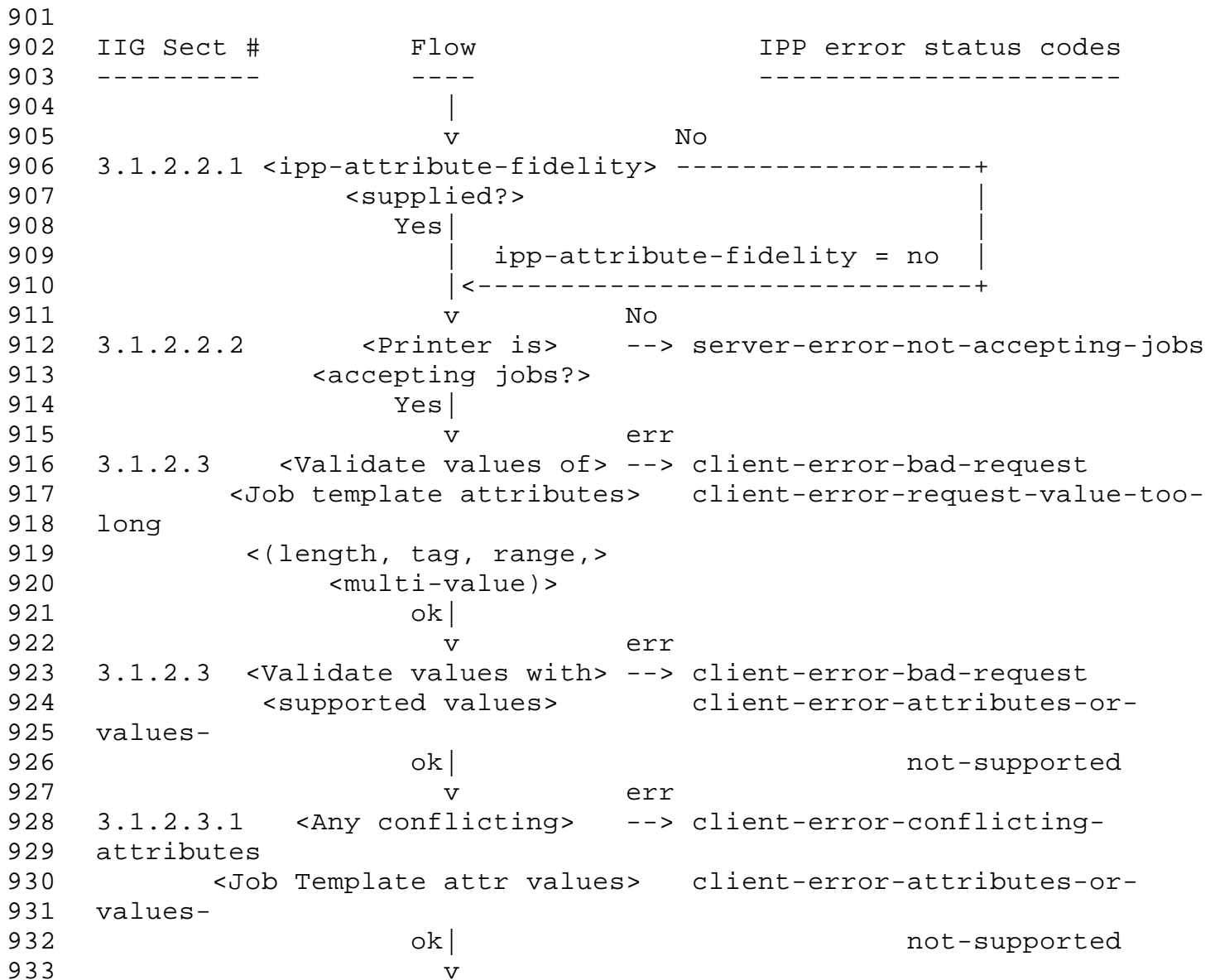
876 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute syntax,
877 REJECT/RETURN 'client-error-request-value-too-long'.
878 ELSE copy the attribute and value to the Unsupported Attributes response group and change the attribute
879 value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.
880

881 Note: Future Operation attributes may be added to the protocol specification that may occur anywhere in the
882 specified group. When the operation is otherwise successful, the IPP object returns the 'successful-ok-ignored-or-
883 substituted-attributes' status code. Ignoring unsupported Operation attributes in all operations is analogous to the
884 handling of unsupported Job Template attributes in the create and Validate-Job operations when the client supplies
885 the "ipp-attribute-fidelity" Operation attribute with the 'false' value. This last rule is so that we can add OPTIONAL
886 Operation attributes to future versions of IPP so that older clients can inter-work with new IPP objects and newer
887 clients can inter-work with older IPP objects. (If the new attribute cannot be ignored without performing
888 unexpectedly, the major version number would have been increased in the protocol document and in the request).
889 This rule for Operation attributes is independent of the value of the "ipp-attribute-fidelity" attribute. For example, if
890 an IPP object doesn't support the OPTIONAL "job-k-octets" attribute', the IPP object treats "job-k-octets" as an
891 unknown attribute and only checks the length for the 'integer' attribute syntax supplied by the client. If it is not four
892 octets, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' status code, else the
893 IPP object copies the attribute to the Unsupported Attribute response group, setting the value to the "out-of-band"
894 'unsupported' value, but otherwise ignores the attribute.

895

895 3.1.2.2 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add Documents

896 This section in combination with the previous section recommends the
 897 processing steps for the Print-Job, Validate-Job, Print-URI, Create-
 898 Job, Send-Document, and Send-URI operations that IPP objects SHOULD
 899 use. These are the operations that create jobs, validate a Print-Job
 900 request, and add documents to a job.



934 3.1.2.2.1 Default "ipp-attribute-fidelity" if not supplied

935 The Printer object checks to see if the client supplied an "ipp-attribute-fidelity" Operation attribute. If the attribute
 936 is not supplied by the client, the IPP object assumes that the value is 'false'.

9 3 7 3.1.2.2.2 Check that the Printer object is accepting jobs

9 3 8 If the value of the Printer objects "printer-is-accepting-jobs" is 'false', the Printer object REJECTS the request and
9 3 9 RETURNS the 'server-error-not-accepting-jobs' status code.

9 4 0 3.1.2.2.3 Validate the values of the Job Template attributes

9 4 1 An IPP object validates the values of all Job Template attribute supplied by the client. The IPP object performs
9 4 2 the analogous syntactic validation checks of each Job Template attribute value that it performs for Operation
9 4 3 attributes (see Section 3.1.2.1.5.):

9 4 4 a) that the length of each value is correct for the attribute syntax tag supplied by the client according to
9 4 5 [RFC2911] Section 4.1.

9 4 6 b) that the attribute syntax tag is correct for that attribute according to [RFC2911] Sections 4.2 to
9 4 7 4.4.

9 4 8 c) that multiple values are supplied only for multi-valued attributes, i.e., that are 1setOf X according
9 4 9 to [RFC2911] Sections 4.2 to 4.4.

9 5 0 As in Section 3.1.2.1.5, if any of these syntactic checks fail, the IPP object REJECTS the request and RETURNS
9 5 1 the 'client-error-bad-request' or 'client-error-request-value-too-long' status code as appropriate, independent of
9 5 2 the value of the "ipp-attribute-fidelity". Since such an error is most likely to be an error detected by a client
9 5 3 developer, rather than by an end-user, the IPP object NEED NOT return an indication of which attribute had the
9 5 4 error in either the Unsupported Attributes Group or the Status Message. The description for each of these
9 5 5 syntactic checks is explicitly expressed in the first IF statement in the following table.

9 5 6 Each Job Template attribute MUST occur no more than once. If an IPP Printer receives a create request with
9 5 7 multiple occurrences of a Job Template attribute, it MAY:

9 5 8 1. reject the operation and return the 'client-error-bad-request' error status code

9 5 9 2. accept the operation and use the first occurrence of the attribute

9 6 0 3. accept the operation and use the last occurrence of the attribute

9 6 1 depending on implementation. Therefore, clients MUST NOT supply multiple occurrences of the same Job
9 6 2 Template attribute in the Job Attributes group in the request.

9 6 3 3.1.2.3 Algorithm for job validation

9 6 4 The process of validating a Job-Template attribute "xxx" against a Printer attribute "xxx-supported" can use the
9 6 5 following validation algorithm (see section 3.2.1.2 in [RFC2911]).

To validate the value U of Job-Template attribute "xxx" against the value V of Printer "xxx-supported", perform the following algorithm:

1. If U is multi-valued, validate each value X of U by performing the algorithm in Table 7 with each value X. Each validation is separate from the standpoint of returning unsupported values. Example: If U is "finishings" that the client supplies with 'staple', 'bind' values, then X takes on the successive values: 'staple', then 'bind'
2. If V is multi-valued, validate X against each Z of V by performing the algorithm in Table 7 with each value Z. If a value Z validates, the validation for the attribute value X succeeds. If it fails, the algorithm is applied to the next value Z of V. If there are no more values Z of V, validation fails. Example" If V is "sides-supported" with values: 'one-sided', 'two-sided-long', and 'two-sided-short', then Z takes on the successive values: 'one-sided', 'two-sided-long', and 'two-sided-short'. If the client supplies "sides" with 'two-sided-long', the first comparison fails ('one-sided' is not equal to 'two-sided-long'), the second comparison succeeds ('two-sided-long' is equal to 'two-sided-long'), and the third comparison ('two-sided-short' with 'two-sided-long') is not even performed.
3. If both U and V are single-valued, let X be U and Z be V and use the validation rules in Table 7.

Table 7 - Rules for validating single values X against Z

Attribute syntax of X	attribute syntax of Z	validated if:
integer	rangeOfInteger	X is within the range of Z
uri	uriScheme	the uri scheme in X is equal to Z
any	boolean	the value of Z is TRUE
any	any	X and Z are of the same type and are equal.

If the value of the Printer object's "xxx-supported" attribute is 'no-value' (because the system administrator hasn't configured a value), the check always fails. If the check fails, the IPP object copies the attribute to the Unsupported Attributes response group with its unsupported value. If the attribute contains more than one value, each value is checked and each unsupported value is separately copied, while supported values are not copied. If an IPP object doesn't recognize/support a Job Template attribute, i.e., there is no corresponding Printer object "xxx-supported" attribute, the IPP object treats the attribute as an unknown or unsupported attribute (see the last row in the table below).

If some Job Template attributes are supported for some document formats and not for others or the values are different for different document formats, the IPP object SHOULD take that into account in this validation using the value of the "document-format" supplied by the client (or defaulted to the value of the Printer's "document-format-default" attribute, if not supplied by the client). For example, if "number-up" is supported for the 'text/plain' document format, but not for the 'application/postscript' document format, the check SHOULD (though it NEED NOT) depend on the value of the "document-format" operation attribute. See "document-format" in [RFC2911] section 3.2.1.1 and 3.2.5.1.

995 Note: whether the request is accepted or rejected is determined by the value of the "ipp-attribute-fidelity" attribute
996 in a subsequent step, so that all Job Template attribute supplied are examined and all unsupported attributes and/or
997 values are copied to the Unsupported Attributes response group.

998 -----

999 job-priority (integer(1:100))

000 IF NOT a single 'integer' value with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
001 request'.
002 IF NOT supplied by the client, use the value of the Printer object's "job-priority-default" attribute at job
003 submission time.
004 IF NOT in the range 1 to 100, inclusive, copy the attribute and the unsupported value to the Unsupported
005 Attributes response group.
006 Map the value to the nearest supported value in the range 1:100 as specified by the number of discrete
007 values indicated by the value of the Printer's "job-priority-supported" attribute. See the formula in
008 [RFC2911] Section 4.2.1.
009

010 job-hold-until (type3 keyword | name)

011 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
012 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
013 IF NOT supplied by the client, use the value of the Printer object's "job-hold-until" attribute at job
014 submission time.
015 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the unsupported
016 value to the Unsupported Attributes response group.
017

018 job-sheets (type3 keyword | name)

019 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
020 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
021 IF NOT in the Printer object's "job-sheets-supported" attribute, copy the attribute and the unsupported
022 value to the Unsupported Attributes response group.
023

024 multiple-document-handling (type2 keyword)

025 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
026 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
027 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute and the
028 unsupported value to the Unsupported Attributes response group.
029

030 copies (integer(1:MAX))

031 IF NOT a single 'integer' value with a length equal to 4 octets,

032 REJECT/RETURN 'client-error-bad-request'.
033 IF NOT in range of the Printer object's "copies-supported" attribute
034 copy the attribute and the unsupported value to the Unsupported Attributes response group.
035
036 finishings (1setOf type2 enum)
037 IF NOT an 'enum' value(s) each with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
038 request'.
039 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the unsupported
040 value(s), but not any supported values, to the Unsupported Attributes response group.
041
042 page-ranges (1setOf rangeOfInteger(1:MAX))
043 IF NOT a 'rangeOfInteger' value(s) each with a length equal to 8 octets, REJECT/RETURN 'client-error-
044 bad-request'.
045 IF first value is greater than second value in any range, the ranges are not in ascending order, or ranges
046 overlap, REJECT/RETURN 'client-error-bad-request'.
047 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to the
048 Unsupported Attributes response group and set the value to the "out-of-band" 'unsupported' value.
049
050 sides (type2 keyword)
051 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
052 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
053 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported value to
054 the Unsupported Attributes response group.
055
056 number-up (integer(1:MAX))
057 IF NOT a single 'integer' value with a length equal to 4 octets,
058 REJECT/RETURN 'client-error-bad-request'.
059 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"
060 attribute, copy the attribute and value to the Unsupported Attribute response group.
061
062 orientation-requested (type2 enum)
063 IF NOT a single 'enum' value with a length equal to 4 octets,
064 REJECT/RETURN 'client-error-bad-request'.
065 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the
066 unsupported value to the Unsupported Attributes response group.
067
068 media (type3 keyword | name)

069 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
 070 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
 071 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported value to
 072 the Unsupported Attributes response group.
 073

074 printer-resolution (resolution)

075 IF NOT a single 'resolution' value with a length equal to 9 octets,
 076 REJECT/RETURN 'client-error-bad-request'.
 077 IF NOT in the Printer object's "printer-resolution-supported" attribute, copy the attribute and the
 078 unsupported value to the Unsupported Attributes response group.
 079

080 print-quality (type2 enum)

081 IF NOT a single 'enum' value with a length equal to 4 octets,
 082 REJECT/RETURN 'client-error-bad-request'.
 083 IF NOT in the Printer object's "print-quality-supported" attribute, copy the attribute and the unsupported
 084 value to the Unsupported Attributes response group.
 085

086 unknown or unsupported attribute (i.e., there is no corresponding Printer object "xxx-supported" attribute)

087 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute syntax,
 088 REJECT/RETURN 'client-error-bad-request' if the length of the attribute syntax is fixed or 'client-error-
 089 request-value-too-long' if the length of the attribute syntax is variable.
 090 ELSE copy the attribute and value to the Unsupported Attributes response group and change the attribute
 091 value to the "out-of-band" 'unsupported' value. Any remaining Job Template Attributes are either
 092 unknown or unsupported Job Template attributes and are validated algorithmically according to
 093 their attribute syntax for proper length (see below).
 094

094 -----

095 If the attribute syntax is supported AND the length check fails, the IPP object REJECTS the request and
 096 RETURNS the 'client-error-bad-request' if the length of the attribute syntax is fixed or the 'client-error-request-
 097 value-too-long' status code if the length of the attribute syntax is variable. Otherwise, the IPP object copies the
 098 unsupported Job Template attribute to the Unsupported Attributes response group and changes the attribute value
 099 to the "out-of-band" 'unsupported' value. The following table shows the length checks for all attribute syntaxes. In
 100 the following table: "<=" means less than or equal, "=" means equal to:

101 Name	Octet length check for read-write attributes
102 -----	-----
103 'textWithLanguage	<= 1023 AND 'naturalLanguage' <= 63
104 'textWithoutLanguage'	<= 1023
105 'nameWithLanguage'	<= 255 AND 'naturalLanguage' <= 63
106 'nameWithoutLanguage'	<= 255
107 'keyword'	<= 255

```
108 'enum' = 4
109 'uri' <= 1023
110 'uriScheme' <= 63
111 'charset' <= 63
112 'naturalLanguage' <= 63
113 'mimeType' <= 255
114 'octetString' <= 1023
115 'boolean' = 1
116 'integer' = 4
117 'rangeOfInteger' = 8
118 'dateTime' = 11
119 'resolution' = 9
120 '1setOf X'
121
```

122 Note: It's possible for a Printer to receive a zero length keyword in a request. Since this is a keyword, its value
123 needs to be compared with the supported values. Assuming that the printer doesn't have any values in its
124 corresponding "xxx-supported" attribute that are keywords of zero length, the comparison will fail. Then the
125 request will be accepted or rejected depending on the value of "ipp-attributes-fidelity" being 'false' or 'true',
126 respectively. No special handling is required for

127 3.1.2.3.1 Check for conflicting Job Template attributes values

128 Once all the Operation and Job Template attributes have been checked individually, the Printer object SHOULD
129 check for any conflicting values among all the supported values supplied by the client. For example, a Printer
130 object might be able to staple and to print on transparencies, however due to physical stapling constraints, the
131 Printer object might not be able to staple transparencies. The IPP object copies the supported attributes and their
132 conflicting attribute values to the Unsupported Attributes response group. The Printer object only copies over
133 those attributes that the Printer object either ignores or substitutes in order to resolve the conflict, and it returns the
134 original values which were supplied by the client. For example suppose the client supplies "finishings" equals
135 'staple' and "media" equals 'transparency', but the Printer object does not support stapling transparencies. If the
136 Printer chooses to ignore the stapling request in order to resolve the conflict, the Printer objects returns "finishings"
137 equal to 'staple' in the Unsupported Attributes response group. If any attributes are multi-valued, only the
138 conflicting values of the attributes are copied.

139 Note: The decisions made to resolve the conflict (if there is a choice) is implementation dependent.

140 3.1.2.3.2 Decide whether to REJECT the request

141 If there were any unsupported Job Template attributes or unsupported/conflicting Job Template attribute values
142 and the client supplied the "ipp-attribute-fidelity" attribute with the 'true' value, the Printer object REJECTS the
143 request and return the status code:

1. 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes supplied by the client.
2. 'client-error-attributes-or-values-not-supported' status code, otherwise.

Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this step. If the unsupported Operation attribute was a serious error, the above already rejected the request in a previous step. If control gets to this step with unsupported Operation attributes being returned, they are not serious errors.

In general, the final results of Job processing are unknown at Job submission time. The client has to rely on notifications or polling to find out what happens at Job processing time. However, there are cases in which some Printers can determine at Job submission time that Job processing is going to fail. As an optimization, we'd like to have the Printer reject the Job in these cases.

There are three types of "processing" errors that might be detectable at Job submission time:

1. 'client-error-document-format-not-supported': For the Print-Job, Send-Document, Print-URI, and Send-URI operations, if all these conditions are true:

- the Printer supports auto-sensing,
- the request "document-format" operation attribute is 'application/octet-stream',
- the Printer receives document data before responding,
- the Printer auto-senses the document format before responding,
- the sensed document format is not supported by the Printer

then the Printer should respond with 'client-error-document-format-not-supported' status.

2. 'client-error-compression-error': For the Print-Job, Send-Document, Print-URI, and Send-URI operations, if all these conditions are true:

- the client supplies a supported value for the "compression" operation attribute in the request
- the Printer receives document data before responding,
- the Printer attempts to decompress the document data before responding,
- the document data cannot be decompressed using the algorithm specified by the "compression" operation attribute

then the Printer should respond with 'client-error-compression-error' status.

3. 'client-error-document-access-error': For the Print-URI, and Send-URI operations, if the Printer attempts and fails to pull the referenced document data before responding, it should respond with 'client-error-document-access-error' status.

Some Printers are not able to detect these errors until Job processing time. In that case, the errors are recorded in the corresponding job-state and job-state reason attributes. (There is no standard way for a client to determine whether a Printer can detect these errors at Job submission time.) For example, if auto-sensing happens AFTER the job is accepted (as opposed to auto-sensing at submit time before returning the response), the implementation

179 aborts the job, puts the job in the 'aborted' state and sets the 'unsupported-document-format' value in the job's
180 "job-state-reasons".

181 A client should always provide a valid "document-format" operation attribute whenever practical. In the absence of
182 other information, a client itself may sniff the document data to determine document format.

183 Auto sensing at Job submission time may be more difficult for the Printer when combined with compression. For
184 auto-sensed Jobs, a client may be better off deferring compression to the transfer protocol layer, e.g.; by using the
185 HTTP Content-Encoding header.

186 3.1.2.3.3 For the Validate-Job operation, RETURN one of the success status codes

187 If the requested operation is the Validate-Job operation, the Printer object returns:

- 188 1. the "successful-ok" status code, if there are no unsupported or conflicting Job Template attributes or
189 values.
- 190 2. the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or values.
- 191 3. the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template attributes
192 or values.

193

194 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this step. If
195 the unsupported Operation attribute was a serious error, the above already rejected the request in a previous step.
196 If control gets to this step with unsupported Operation attributes being returned, they are not serious errors.

197 3.1.2.3.4 Create the Job object with attributes to support

198 If "ipp-attribute-fidelity" is set to 'false' (or it was not supplied by the client), the Printer object:

- 199 1. creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and initializes all
200 of the job's other supported Job Description attributes.
- 201 2. removes all unsupported attributes from the Job object.
- 202 3. for each unsupported value, removes either the unsupported value or substitutes the unsupported attribute
203 value with some supported value. If an attribute has no values after removing unsupported values from it,
204 the attribute is removed from the Job object (so that the normal default behavior at job processing time
205 will take place for that attribute).
- 206 4. for each conflicting value, removes either the conflicting value or substitutes the conflicting attribute value
207 with some other supported value. If an attribute has no values after removing conflicting values from it, the
208 attribute is removed from the Job object (so that the normal default behavior at job processing time will
209 take place for that attribute).

210

211 If there were no attributes or values flagged as unsupported, or the value of 'ipp-attribute-fidelity' was 'false', the
212 Printer object is able to accept the create request and create a new Job object. If the "ipp-attribute-fidelity"
213 attribute is set to 'true', the Job Template attributes that populate the new Job object are necessarily all the Job
214 Template attributes supplied in the create request. If the "ipp-attribute-fidelity" attribute is set to 'false', the Job
215 Template attributes that populate the new Job object are all the client supplied Job Template attributes that are
216 supported or that have value substitution. Thus, some of the requested Job Template attributes may not appear in
217 the Job object because the Printer object did not support those attributes. The attributes that populate the Job
218 object are persistently stored with the Job object for that Job. A Get-Job-Attributes operation on that Job object
219 will return only those attributes that are persistently stored with the Job object.

220 Note: All Job Template attributes that are persistently stored with the Job object are intended to be "override
221 values"; that is, they take precedence over whatever other embedded instructions might be in the document
222 data itself. However, it is not possible for all Printer objects to realize the semantics of "override". End users may
223 query the Printer's "pdl-override-supported" attribute to determine if the Printer either attempts or does not attempt
224 to override document data instructions with IPP attributes.

225 There are some cases, where a Printer supports a Job Template attribute and has an associated default value set
226 for that attribute. In the case where a client does not supply the corresponding attribute, the Printer does not use its
227 default values to populate Job attributes when creating the new Job object; only Job Template attributes actually in
228 the create request are used to populate the Job object. The Printer's default values are only used later at Job
229 processing time if no other IPP attribute or instruction embedded in the document data is present.

230 Note: If the default values associated with Job Template attributes that the client did not supply were to be used to
231 populate the Job object, then these values would become "override values" rather than defaults. If the Printer
232 supports the 'attempted' value of the "pdl-override-supported" attribute, then these override values could replace
233 values specified within the document data. This is not the intent of the default value mechanism. A default value for
234 an attribute is used only if the create request did not specify that attribute (or it was ignored when allowed by "ipp-
235 attribute-fidelity" being 'false') and no value was provided within the content of the document data.

236 If the client does not supply a value for some Job Template attribute, and the Printer does not support that
237 attribute, as far as IPP is concerned, the result of processing that Job (with respect to the missing attribute) is
238 undefined.

239 3.1.2.3.5 Return one of the success status codes

240 Once the Job object has been created, the Printer object accepts the request and returns to the client:

- 241 1. the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes or
242 values.
- 243 2. the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template attribute or
244 values.
- 245 3. the 'successful-ok-ignored-or-substituted-attributes' status code, if there are only unsupported Job
246 Template attributes or values.

247

248 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this step. If
249 the unsupported Operation attribute was a serious error, the above already rejected the request in a previous step.
250 If control gets to this step with unsupported Operation attributes being returned, they are not serious errors.

251 The Printer object also returns Job status attributes that indicate the initial state of the Job ('pending', 'pending-held',
252 'processing', etc.), etc. See Print-Job Response, [RFC2911] section 3.2.1.2.

253 3.1.2.3.6 Accept appended Document Content

254 The Printer object accepts the appended Document Content data and either starts it printing, or spools it for later
255 processing.

256 3.1.2.3.7 Scheduling and Starting to Process the Job

257 The Printer object uses its own configuration and implementation specific algorithms for scheduling the Job in the
258 correct processing order. Once the Printer object begins processing the Job, the Printer changes the Job's state to
259 'processing'. If the Printer object supports PDL override (the "pdl-override-supported" attribute set to 'attempted'),
260 the implementation does its best to see that IPP attributes take precedence over embedded instructions in the
261 document data.

262 3.1.2.3.8 Completing the Job

263 The Printer object continues to process the Job until it can move the Job into the 'completed' state. If an Cancel-
264 Job operation is received, the implementation eventually moves the Job into the 'canceled' state. If the system
265 encounters errors during processing that do not allow it to progress the Job into a completed state, the
266 implementation halts all processing, cleans up any resources, and moves the Job into the 'aborted' state.

267 3.1.2.3.9 Destroying the Job after completion

268 Once the Job moves to the 'completed', 'aborted', or 'canceled' state, it is an implementation decision as to when to
269 destroy the Job object and release all associated resources. Once the Job has been destroyed, the Printer would
270 return either the "client-error-not-found" or "client-error-gone" status codes for operations directed at that Job.

271 Note: the Printer object SHOULD NOT re-use a "job-uri" or "job-id" value for a sufficiently long time after a job
272 has been destroyed, so that stale references kept by clients are less likely to access the wrong (newer) job.

273 3.1.2.3.10 Interaction with "ipp-attribute-fidelity"

274 Some Printer object implementations may support "ipp-attribute-fidelity" set to 'true' and "pdl-override-supported"
275 set to 'attempted' and yet still not be able to realize exactly what the client specifies in the create request. This is
276 due to legacy decisions and assumptions that have been made about the role of job instructions embedded within
277 the document data and external job instructions that accompany the document data and how to handle conflicts
278 between such instructions. The inability to be 100% precise about how a given implementation will behave is also
279 compounded by the fact that the two special attributes, "ipp-attribute-fidelity" and "pdl-override-supported", apply
280 to the whole job rather than specific values for each attribute. For example, some implementations may be able to
281 override almost all Job Template attributes except for "number-up". Character Sets, natural languages, and
282 internationalization

283 This section discusses character set support, natural language support and internationalization.

284 3.1.2.3.11 Character set code conversion support

285 IPP clients and IPP objects are REQUIRED to support UTF-8. They MAY support additional charsets. It is
286 RECOMMENDED that an IPP object also support US-ASCII, since many clients support US-ASCII, and
287 indicate that UTF-8 and US-ASCII are supported by populating the Printer's "charset-supported" with 'utf-8' and
288 'us-ascii' values. An IPP object is required to code covert with as little loss as possible between the charsets that it
289 supports, as indicated in the Printer's "charsets-supported" attribute.

290 How should the server handle the situation where the "attributes-charset" of the response itself is "us-ascii", but one
291 or more attributes in that response is in the "utf-8" format?

292 Example: Consider a case where a client sends a Print-Job request with "utf-8" as the value of "attributes-charset"
293 and with the "job-name" attribute supplied. Later another client submits a Get-Job-Attribute or Get-Jobs request.
294 This second request contains the "attributes-charset" with value "us-ascii" and "requested-attributes" attribute with
295 exactly one value "job-name".

296 According to the RFC2911 document (section 3.1.4.2), the value of the "attributes-charset" for the response of the
297 second request must be "us-ascii" since that is the charset specified in the request. The "job-name" value, however,
298 is in "utf-8" format. Should the request be rejected even though both "utf-8" and "us-ascii" charsets are supported
299 by the server? or should the "job-name" value be converted to "us-ascii" and return "successful-ok-conflicting-
300 attributes" (0x0002) as the status code?

301 Answer: An IPP object that supports both utf-8 (REQUIRED) and us-ascii, the second paragraph of section
302 3.1.4.2 applies so that the IPP object MUST accept the request, perform code set conversion between these two
303 charsets with "the highest fidelity possible" and return 'successful-ok', rather than a warning 'successful-ok-
304 conflicting-attributes', or an error. The printer will do the best it can to convert between each of the character sets
305 that it supports--even if that means providing a string of question marks because none of the characters are
306 representable in US ASCII. If it can't perform such conversion, it MUST NOT advertise us-ascii as a value of its
307 "attributes-charset-supported" and MUST reject any request that requests 'us-ascii'.

308 One IPP object implementation strategy is to convert all request text and name values to a Unicode internal
309 representation. This is 16-bit and virtually universal. Then convert to the specified operation attributes-charset on
310 output.

311 Also it would be smarter for a client to ask for 'utf-8', rather than 'us-ascii' and throw away characters that it
312 doesn't understand, rather than depending on the code conversion of the IPP object.

313 3.1.2.3.12 What charset to return when an unsupported charset is requested (Issue 1.19)?

314 Section 3.1.4.1 Request Operation attributes was clarified in November 1998 as follows:

315 All clients and IPP objects MUST support the 'utf-8' charset [RFC2044] and MAY support additional charsets
316 provided that they are registered with IANA [IANA-CS]. If the Printer object does not support the client
317 supplied charset value, the Printer object MUST reject the request, set the "attributes-charset" to 'utf-8' in the
318 response, and return the 'client-error-charset-not-supported' status code and any 'text' or 'name' attributes using
319 the 'utf-8' charset.

320 Since the client and IPP object MUST support UTF-8, returning any text or name attributes in UTF-8 when the
321 client requests a charset that is not supported should allow the client to display the text or name.

322 Since such an error is a client error, rather than a user error, the client should check the status code first so that it
323 can avoid displaying any other returned 'text' and 'name' attributes that are not in the charset requested.

324 Furthermore, [RFC2911] section 14.1.4.14 client-error-charset-not-supported (0x040D) was clarified in
325 November 1998 as follows:

326 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-charset"
327 operation attribute, the Printer MUST reject the operation and return this status and any 'text' or 'name' attributes
328 using the 'utf-8' charset (see Section 3.1.4.1).

329 3.1.2.3.13 Natural Language Override (NLO)

330 The 'text' and 'name' attributes each have two forms. One has an implicit natural language, and the other has an
331 explicit natural language. The 'textWithoutLanguage' and 'textWithLanguage' are the two 'text' forms. The
332 'nameWithoutLanguage' and 'nameWithLanguage' are the two 'name' forms. If a receiver (IPP object or IPP
333 client) supports an attribute with attribute syntax 'text', it MUST support both forms in a request and a response. A
334 sender (IPP client or IPP object) MAY send either form for any such attribute. When a sender sends a
335 WithoutLanguage form, the implicit natural language is specified in the "attributes-natural-language" operation
336 attribute, which all senders MUST include in every request and response.

337 When a sender sends a WithLanguage form, it MAY be different from the implicit natural language supplied by the
338 sender or it MAY be the same. The receiver MUST treat either form equivalently.

3 3 9 There is an implementation decision for senders, whether to always send the WithLanguage forms or use the
3 4 0 WithoutLanguage form when the attribute's natural language is the same as the request or response. The former
3 4 1 approach makes the sender implementation simpler. The latter approach is more efficient on the wire and allows
3 4 2 inter-working with non-conforming receivers that fail to support the WithLanguage forms. As each approach have
3 4 3 advantages, the choice is completely up to the implementer of the sender.

3 4 4 Furthermore, when a client receives a 'text' or 'name' job attribute that it had previously supplied, that client MUST
3 4 5 NOT expect to see the attribute in the same form, i.e., in the same WithoutLanguage or WithLanguage form as the
3 4 6 client supplied when it created the job. The IPP object is free to transform the attribute from the WithLanguage
3 4 7 form to the WithoutLanguage form and vice versa, as long as the natural language is preserved. However, in order
3 4 8 to meet this latter requirement, it is usually simpler for the IPP object implementation to store the natural language
3 4 9 explicitly with the attribute value, i.e., to store using an internal representation that resembles the WithLanguage
3 5 0 form.

3 5 1 The IPP Printer MUST copy the natural language of a job, i.e., the value of the "attributes-natural-language"
3 5 2 operation attribute supplied by the client in the create operation, to the Job object as a Job Description attribute, so
3 5 3 that a client is able to query it. In returning a Get-Job-Attributes response, the IPP object MAY return one of
3 5 4 three natural language values in the response's "attributes-natural-language" operation attribute: (1) that requested
3 5 5 by the requester, (2) the natural language of the job, or (3) the configured natural language of the IPP Printer, if the
3 5 6 requested language is not supported by the IPP Printer.

3 5 7 This "attributes-natural-language" Job Description attribute is useful for an IPP object implementation that prints
3 5 8 start sheets in the language of the user who submitted the job. This same Job Description attribute is useful to a
3 5 9 multi-lingual operator who has to communicate with different job submitters in different natural languages. This
3 6 0 same Job Description attribute is expected to be used in the future to generate notification messages in the natural
3 6 1 language of the job submitter.

3 6 2 Early drafts of [RFC2911] contained a job-level natural language override (NLO) for the Get-Jobs response. A
3 6 3 job-level (NLO) is an (unrequested) Job Attribute which then specified the implicit natural language for any other
3 6 4 WithoutLanguage job attributes returned in the response for that job. Interoperability testing of early
3 6 5 implementations showed that no one was implementing the job-level NLO in Get-Job responses. So the job-level
3 6 6 NLO was eliminated from the Get-Jobs response. This simplification makes all requests and responses consistent
3 6 7 in that the implicit natural language for any WithoutLanguage 'text' or 'name' form is always supplied in the request's
3 6 8 or response's "attributes-natural-language" operation attribute.

3 6 9 3.1.3 Status codes returned by operation

3 7 0 This section corresponds to [RFC2911] section 3.1.6 "Operation Response Status Codes and Status Messages".
3 7 1 This section lists all status codes once in the first operation (Print-Job). Then it lists the status codes that are
3 7 2 different or specialized for subsequent operations under each operation.

3 7 3 3.1.3.1 Printer Operations

374 3.1.3.1.1 Print-Job

375 The Printer object **MUST** return one of the following "status-code" values for the indicated reason. Whether all of
376 the document data has been accepted or not before returning the success or error response depends on
377 implementation. See Section 13 in [RFC2911] for a more complete description of each status code.

378 For the following success status codes, the Job object has been created and the "job-id", and "job-uri" assigned
379 and returned in the response:

- 380 successful-ok: no request attributes were substituted or ignored.
- 381 successful-ok-ignored-or-substituted-attributes: some supplied (1) attributes were ignored or (2) unsupported
382 attribute syntaxes or values were substituted with supported values or were ignored. Unsupported
383 attributes, attribute syntax's, or values **MUST** be returned in the Unsupported Attributes group of the
384 response.
- 385 successful-ok-conflicting-attributes: some supplied attribute values conflicted with the values of other supplied
386 attributes and were either substituted or ignored. Attributes or values which conflict with other attributes
387 and have been substituted or ignored **MUST** be returned in the Unsupported Attributes group of the
388 response as supplied by the client.

389

390 [RFC2911] section 3.1.6 Operation Status Codes and Messages states:

391 If the Printer object supports the "status-message" operation attribute, it **SHOULD** use the **REQUIRED** 'utf-8'
392 charset to return a status message for the following error status codes (see section 13 in [RFC2911]): 'client-
393 error-bad-request', 'client-error-charset-not-supported', 'server-error-internal-error', 'server-error-operation-not-
394 supported', and 'server-error-version-not-supported'. In this case, it **MUST** set the value of the "attributes-
395 charset" operation attribute to 'utf-8' in the error response.

396 For the following error status codes, no job is created and no "job-id" or "job-uri" is returned:

- 397 client-error-bad-request: The request syntax does not conform to the specification.
- 398 client-error-forbidden: The request is being refused for authorization or authentication reasons. The
399 implementation security policy is to not reveal whether the failure is one of authentication or authorization.
- 400 client-error-not-authenticated: Either the request requires authentication information to be supplied or the
401 authentication information is not sufficient for authorization.
- 402 client-error-not-authorized: The requester is not authorized to perform the request on the target object.
- 403 client-error-not-possible: The request cannot be carried out because of the state of the system. See also
404 'server-error-not-accepting-jobs' status code, which **MUST** take precedence if the Printer object's
405 "printer-accepting-jobs" attribute is 'false'.
- 406 client-error-timeout: not applicable.
- 407 client-error-not-found: the target object does not exist.
- 408 client-error-gone: the target object no longer exists and no forwarding address is known.
- 409 client-error-request-entity-too-large: the size of the request and/or print data exceeds the capacity of the IPP
410 Printer to process it.

411 client-error-request-value-too-long: the size of request variable length attribute values, such as 'text' and 'name'
412 attribute syntax's, exceed the maximum length specified in [RFC2911] for the attribute and MUST be
413 returned in the Unsupported Attributes Group.

414 client-error-document-format-not-supported: the document format supplied is not supported. The
415 "document-format" attribute with the unsupported value MUST be returned in the Unsupported Attributes
416 Group. This error SHOULD take precedence over any other 'xxx-not-supported' error, except 'client-
417 error-charset-not-supported'.

418 client-error-attributes-or-values-not-supported: one or more supplied attributes, attribute syntax's, or values
419 are not supported and the client supplied the "ipp-attributes-fidelity" operation attribute with a 'true' value.
420 They MUST be returned in the Unsupported Attributes Group as explained below.

421 client-error-uri-scheme-not-supported: not applicable.

422 client-error-charset-not-supported: the charset supplied in the "attributes-charset" operation attribute is not
423 supported. The Printer's "configured-charset" MUST be returned in the response as the value of the
424 "attributes-charset" operation attribute and used for any 'text' and 'name' attributes returned in the error
425 response. This error SHOULD take precedence over any other error, unless the request syntax is so bad
426 that the client's supplied "attributes-charset" cannot be determined.

427 client-error-conflicting-attributes: one or more supplied attribute values conflicted with each other and the
428 client supplied the "ipp-attributes-fidelity" operation attribute with a 'true' value. They MUST be returned
429 in the Unsupported Attributes Group as explained below.

430 server-error-internal-error: an unexpected condition prevents the request from being fulfilled.

431 server-error-operation-not-supported: not applicable (since Print-Job is REQUIRED).

432 server-error-service-unavailable: the service is temporarily overloaded.

433 server-error-version-not-supported: the version in the request is not supported. The "closest" version number
434 supported MUST be returned in the response.

435 server-error-device-error: a device error occurred while receiving or spooling the request or document data or
436 the IPP Printer object can only accept one job at a time.

437 server-error-temporary-error: a temporary error such as a buffer full write error, a memory overflow, or a
438 disk full condition occurred while receiving the request and/or the document data.

439 server-error-not-accepting-jobs: the Printer object's "printer-is-not-accepting-jobs" attribute is 'false'.

440 server-error-busy: the Printer is too busy processing jobs to accept another job at this time.

441 server-error-job-canceled: the job has been canceled by an operator or the system while the client was
442 transmitting the document data.

443 3.1.3.1.2 Print-URI

444 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Print-URI
445 with the following specializations and differences. See Section 14 for a more complete description of each status
446 code.

447 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation attribute is
448 not supported and is returned in the Unsupported Attributes group.

449 server-error-operation-not-supported: the Print-URI operation is not supported.

450

451 3.1.3.1.3 Validate-Job

452 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Validate-
453 Job. See Section 13 in [RFC2911] for a more complete description of each status code.

454 3.1.3.1.4 Create-Job

455 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Create-Job
456 with the following specializations and differences. See Section 13 in [RFC2911] for a more complete description
457 of each status code.

458 server-error-operation-not-supported: the Create-Job operation is not supported.

459 client-error-multiple-document-jobs-not-supported: while the Create-Job and Send-Document operations are
460 supported, this implementation doesn't support more than one document with data.

461 3.1.3.1.5 Get-Printer-Attributes

462 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the Get-
463 Printer-Attributes operation with the following specialization's and differences. See Section 13 in [RFC2911] for
464 a more complete description of each status code.

465 For the following success status codes, the requested attributes are returned in Group 3 in the response:

466 successful-ok: no operation attributes or values were substituted or ignored (same as Print-Job) and no
467 requested attributes were unsupported.

468 *Note to client implementers: If the client requests attributes that are not supported, the Printer is*
469 *supposed to return 'successful-ok-ignored-or-substituted-attributes', rather than 'successful-ok'.*

470 *However, a number of implementations have been found not to conform to this requirement, so*
471 *clients should be tolerant of such Printers.*

472 successful-ok-ignored-or-substituted-attributes: The "requested-attributes" operation attribute SHOULD be
473 returned with the unsupported values in the Unsupported Attributes Group.

474 *Note to client implementers: Although not recommended, the Unsupported Attribute Group and its*
475 *contents may be omitted. Clients should be prepared for this behavior.*

476 successful-ok-conflicting-attributes: same as Print-Job.

477 For the error status codes, Group 3 is returned containing no attributes or is not returned at all:

478 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.

479 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.

480 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes and/or
481 values MUST be ignored and an appropriate success code returned (see above).

482 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.

483 server-error-operation-not-supported: not applicable (since Get-Printer-Attributes is REQUIRED).

484 server-error-device-error: same as Print-Job, except that no document data is involved.

485 server-error-temporary-error: same as Print-Job, except that no document data is involved.
486 server-error-not-accepting-jobs: not applicable..
487 server-error-busy: same as Print-Job, except the IPP object is too busy to accept even query requests.
488 server-error-job-canceled: not applicable..

489 3.1.3.1.6 Get-Jobs

490 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the Get-Jobs
491 operation with the following specialization's and differences. See Section 13 in [RFC2911] for a more complete
492 description of each status code.

493 For the following success status codes, the requested attributes are returned in Group 3 in the response:

494 successful-ok: same as Get-Printer-Attributes (see section 3.1.3.1.5).
495 successful-ok-ignored-or-substituted-attributes: same as Get-Printer-Attributes (see section 3.1.3.1.5).
496 successful-ok-conflicting-attributes: same as Get-Printer-Attributes (see section 3.1.3.1.5).

497 For any error status codes, Group 3 is returned containing no attributes or is not returned at all. The following brief
498 error status code descriptions contain unique information for use with Get-Jobs operation. See section 14 for the
499 other error status codes that apply uniformly to all operations:

500 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.
501 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.
502 client-error-document-format-not-supported: not applicable.
503 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes and/or
504 values MUST be ignored and an appropriate success code returned (see above).
505 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.
506 server-error-operation-not-supported: not applicable (since Get-Jobs is REQUIRED).
507 server-error-device-error: same as Print-Job, except that no document data is involved.
508 server-error-temporary-error: same as Print-Job, except that no document data is involved.
509 server-error-not-accepting-jobs: not applicable.
510 server-error-job-canceled: not applicable.

511 3.1.3.1.7 Pause-Printer

512 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Pause-
513 Printer with the following specializations and differences. See Section 13 in [RFC2911] for a more complete
514 description of each status code.

515 For the following success status codes, the Printer object is being stopped from scheduling jobs on all its devices.

516 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
517 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
518 successful-ok-conflicting-attributes: same as Print-Job.
519

5 2 0 For any of the error status codes, the Printer object has not been stopped from scheduling jobs on all its devices.

5 2 1 client-error-not-possible: not applicable.

5 2 2 client-error-not-found: the target Printer object does not exist.

5 2 3 client-error-gone: the target Printer object no longer exists and no forwarding address is known.

5 2 4 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.

5 2 5 client-error-document-format-not-supported: not applicable.

5 2 6 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-jobs"
5 2 7 attribute is not involved.

5 2 8 server-error-operation-not-supported: the Pause-Printer operation is not supported.

5 2 9 server-error-device-error: not applicable.

5 3 0 server-error-temporary-error: same as Print-Job, except no document data is involved.

5 3 1 server-error-not-accepting-jobs: not applicable.

5 3 2 server-error-job-canceled: not applicable.

5 3 3 3.1.3.1.8 Resume-Printer

5 3 4 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the specialization's
5 3 5 described for Pause-Printer are applicable to Resume-Printer. See Section 13 in [RFC2911] for a more complete
5 3 6 description of each status code.

5 3 7 For the following success status codes, the Printer object resumes scheduling jobs on all its devices.

5 3 8 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

5 3 9 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

5 4 0 successful-ok-conflicting-attributes: same as Print-Job.

5 4 1 For any of the error status codes, the Printer object does not resume scheduling jobs.

5 4 2 server-error-operation-not-supported: the Resume-Printer operation is not supported.

5 4 3

5 4 4 3.1.3.1.8.1 What about Printers unable to change state due to an error condition?

5 4 5 If, in case, the IPP printer is unable to change its state due to some problem with the actual printer device (say, it is
5 4 6 shut down or there is a media-jam as indicated in [RFC2911]), what should be the result of the "Resume-Printer"
5 4 7 operation? Should it still change the 'printer-state-reasons' and return success or should it fail ?

5 4 8 The Resume-Printer operation must clear the 'paused' or 'moving-to-paused' 'printer-state-message'. The
5 4 9 operation must return a 'successful-ok' status code.

5 5 0 3.1.3.1.8.2 How is "printer-state" handled on Resume-Printer?

5 5 1

552 If the Resume-Printer operation succeeds, what should be the value of "printer-state" and who should take care of
553 the "printer-state" attribute value later on ?

554 The Resume-Printer operation may change the "printer-state-reasons" value.

555 The "printer-state" will change to one of three states:

- 556 1. 'idle' - no additional jobs and no error conditions present
- 557 2. 'processing' - job available and no error conditions present
- 558 3. current state (i.e. no change) an error condition is present (e.g. media jam)

559 In the third case the "printer-state-reason" will be cleared by automata when it detects the error condition no
560 longer exists. The "printer-state" will move to 'idle' or 'processing' when conditions permit. (i.e. no more error
561 conditions)

562 3.1.3.1.9 Purge-Printer

563 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the specialization's
564 described for Pause-Printer are applicable to Purge-Printer. See Section 13 in [RFC2911] for a more complete
565 description of each status code.

566 For the following success status codes, the Printer object purges all it's jobs.

- 567 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
- 568 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
- 569 successful-ok-conflicting-attributes: same as Print-Job.

570 For any of the error status codes, the Printer object does not purge any jobs.

571 server-error-operation-not-supported: the Purge-Printer operation is not supported.

572 3.1.3.2 Job Operations

573 3.1.3.2.1 Send-Document

574 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the Get-
575 Printer-Attributes operation with the following specialization's and differences. See Section 13 in [RFC2911] for
576 a more complete description of each status code.

577 For the following success status codes, the document has been added to the specified Job object and the job's
578 "number-of-documents" attribute has been incremented:

- 579 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
- 580 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

581 successful-ok-conflicting-attributes: same as Print-Job.

582 For the error status codes, no document has been added to the Job object and the job's "number-of-documents"
583 attribute has not been incremented:

584 client-error-not-possible: Same as Print-Job, except that the Printer's "printer-is-accepting-jobs" attribute is not
585 involved, so that the client is able to finish submitting a job that was created with a Create-Job operation
586 after this attribute has been set to 'true'. Another condition is that the state of the job precludes Send-
587 Document, i.e., the job has already been closed out by the client. However, if the IPP Printer closed out
588 the job due to timeout, the 'client-error-timeout' error status SHOULD be returned instead.

589 client-error-timeout: This request was sent after the Printer closed the job, because it has not received a Send-
590 Document or Send-URI operation within the Printer's "multiple-operation-time-out" period .

591 client-error-request-entity-too-large: same as Print-Job.

592 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attributes-fidelity" operation attribute is
593 not involved..

594 server-error-operation-not-supported: the Send-Document request is not supported.

595 server-error-not-accepting-jobs: not applicable.

596 server-error-job-canceled: the job has been canceled by an operator or the system while the client was
597 transmitting the data.

598 3.1.3.2.2 Send-URI

599 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the specialization's
600 described for Send-Document are applicable to Send-URI. See Section 13 in [RFC2911] for a more complete
601 description of each status code.

602 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation attribute is
603 not supported and the "document-uri" attribute MUST be returned in the Unsupported Attributes group.

604 server-error-operation-not-supported: the Send-URI operation is not supported.

605

606 3.1.3.2.3 Cancel-Job

607 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Cancel-Job
608 with the following specializations and differences. See Section 13 in [RFC2911] for a more complete description
609 of each status code.

610 For the following success status codes, the Job object is being canceled or has been canceled:

611 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

612 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

613 successful-ok-conflicting-attributes: same as Print-Job.

614

615 For any of the error status codes, the Job object has not been canceled or was previously canceled.

616 client-error-not-possible: The request cannot be carried out because of the state of the Job object
617 ('completed', 'canceled', or 'aborted') or the state of the system.
618 client-error-not-found: the target Printer and/or Job object does not exist.
619 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding address is known.
620 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
621 client-error-document-format-not-supported: not applicable.
622 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes and
623 values MUST be ignored.
624 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-jobs"
625 attribute is not involved.
626 server-error-operation-not-supported: not applicable (Cancel-Job is REQUIRED).
627 server-error-device-error: same as Print-Job, except no document data is involved.
628 server-error-temporary-error: same as Print-Job, except no document data is involved.
629 server-error-not-accepting-jobs: not applicable..
630 server-error-job-canceled: not applicable.

631 3.1.3.2.4 Get-Job-Attributes

632 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Get-Job-
633 Attributes with the following specializations and differences. See Section 13 in [RFC2911] for a more complete
634 description of each status code.

635 For the following success status codes, the requested attributes are returned in Group 3 in the response:

636 successful-ok: same as Get-Printer-Attributes (see section 3.1.3.1.5).
637 successful-ok-ignored-or-substituted-attributes: same as Get-Printer-Attributes (see section 3.1.3.1.5).
638 successful-ok-conflicting-attributes: same as Get-Printer-Attributes (see section 3.1.3.1.5).

639 For the error status codes, Group 3 is returned containing no attributes or is not returned at all.

640 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.
641 client-error-document-format-not-supported: not applicable.
642 client-error-attributes-or-values-not-supported: not applicable.
643 client-error-uri-scheme-not-supported: not applicable.
644 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes and/or
645 values MUST be ignored and an appropriate success code returned (see above).
646 client-error-conflicting-attributes: not applicable
647 server-error-operation-not-supported: not applicable (since Get-Job-Attributes is REQUIRED).
648 server-error-device-error: same as Print-Job, except no document data is involved.
649 server-error-temporary-error: sane as Print-Job, except no document data is involved..
650 server-error-not-accepting-jobs: not applicable.
651 server-error-job-canceled: not applicable.

652 3.1.3.2.5 Hold-Job

653 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Hold-Job
654 with the following specializations and differences. See Section 13 in [RFC2911] for a more complete description
655 of each status code.

656 For the following success status codes, the Job object is being held or has been held:

657 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

658 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

659 successful-ok-conflicting-attributes: same as Print-Job.

660

661 For any of the error status codes, the Job object has not been held or was previously held.

662 client-error-not-possible: The request cannot be carried out because of the state of the Job object
663 ('completed', 'canceled', or 'aborted') or the state of the system.

664 client-error-not-found: the target Printer and/or Job object does not exist.

665 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding address is known.

666 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.

667 client-error-document-format-not-supported: not applicable.

668 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-jobs"
669 attribute is not involved.

670 server-error-operation-not-supported: the Hold-Job operation is not supported.

671 server-error-device-error: not applicable.

672 server-error-temporary-error: same as Print-Job, except no document data is involved.

673 server-error-not-accepting-jobs: not applicable.

674 server-error-job-canceled: not applicable.

675 3.1.3.2.6 Release-Job

676 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the specialization's
677 described for Hold-Job are applicable to Release-Job. See Section 13 in [RFC2911] for a more complete
678 description of each status code.

679 server-error-operation-not-supported: the Release-Job operation is not supported.

680 3.1.3.2.7 Restart-Job

681 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the specialization's
682 described for Hold-Job are applicable to Restart-Job. See Section 13 in [RFC2911] for a more complete
683 description of each status code.

684 server-error-operation-not-supported: the Restart-Job operation is not supported.

685

686 3.1.3.2.7.1 Can documents be added to a restarted job?

687 Assume I give a Create-Job request along with a set of 5 documents . All the documents get printed and the job
688 state is moved to completed . I issue a Restart-Job request on the job. Now the issue is that, if I try to add new
689 documents to the restarted job, will the IPP Server permit me to do so or return "client-error-not-possible " and
690 again print those 5 jobs?

691 A job can not move to the 'completed' state until all the documents have been processed. The 'last-document' flag
692 indicates when the last document for a job is being sent from the client. This is the semantic equivalent of closing a
693 job. No documents may be added once a job is closed. Section 3.3.7 of the IPP/1.1 model states "The job is
694 moved to the 'pending' job state and restarts the beginning on the same IPP Printer object with the same attribute
695 values." 'number-of-documents' is a job attribute.

696
697
698 3.1.4 Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)

699 In the Get-Printer-Attributes, Get-Jobs, or Get-Job-Attributes responses, the client cannot depend on getting
700 unsupported attributes returned in the Unsupported Attributes group that the client requested, but are not
701 supported by the IPP object. However, such unsupported requested attributes will not be returned in the Job
702 Attributes or Printer Attributes group (since they are unsupported). Furthermore, the IPP object is REQUIRED to
703 return the 'successful-ok-ignored-or-substituted-attributes' status code, so that the client knows that not all that was
704 requested has been returned. However, see the note in section 3.1.3.1.5 that some non-conforming Printers return
705 'successful-ok'.

706 3.1.5 Sending empty attribute groups

707 The [RFC2911] and [RFC2910] specifications RECOMMEND that a sender not send an empty attribute group
708 in a request or a response. However, they REQUIRE a receiver to accept an empty attribute group as equivalent
709 to the omission of that group. So a client SHOULD omit the Job Template Attributes group entirely in a create
710 operation that is not supplying any Job Template attributes. Similarly, an IPP object SHOULD omit an empty
711 Unsupported Attributes group if there are no unsupported attributes to be returned in a response.

712 The [RFC2910] specification REQUIRES a receiver to be able to receive either an empty attribute group or an
713 omitted attribute group and treat them equivalently. The term "receiver" means an IPP object for a request and a
714 client for a response. The term "sender" means a client for a request and an IPP object for a response.

715 There is an exception to the rule for Get-Jobs when there are no attributes to be returned. [RFC2910] contains
716 the following paragraph:

717 The syntax allows an xxx-attributes-tag to be present when the xxx-attribute-sequence that follows is empty. The
718 syntax is defined this way to allow for the response of Get-Jobs where no attributes are returned for some job-
719 objects. Although it is RECOMMENDED that the sender not send an xxx-attributes-tag if there are no attributes
720 (except in the Get-Jobs response just mentioned), the receiver MUST be able to decode such syntax.

721 3.2 Printer Operations

722 3.2.1 Print-Job operation

723 3.2.1.1 Flow controlling the data portion of a Print-Job request (Issue 1.22)

724 A paused printer, or one that is stopped due to paper out or jam or spool space full or buffer space full, may flow
725 control the data of a Print-Job operation (at the TCP/IP layer), so that the client is not able to send all the
726 document data. Consequently, the Printer will not return a response until the condition is changed.

727 The Printer should not return a Print-Job response with an error code in any of these conditions, since either the
728 printer will be resumed and/or the condition will be freed either by human intervention or as jobs print.

729 In writing test scripts to test IPP Printers, the script must also be written not to expect a response, if the printer has
730 been paused, until the printer is resumed, in order to work with all possible implementations.

731 3.2.1.2 Returning job-state in Print-Job response (Issue 1.30)

732 An IPP client submits a small job via Print-Job. By the time the IPP printer/print server is putting together a
733 response to the operation, the job has finished printing and been removed as an object from the print system.
734 What should the job-state be in the response?

735 The Model suggests that the Printer return a response before it even accepts the document content. The Job
736 Object Attributes are returned only if the IPP object returns one of the success status codes. Then the job-state
737 would always be "pending" or "pending-held".

738 This issue comes up for the implementation of an IPP Printer object as a server that forwards jobs to devices that
739 do not provide job status back to the server. If the server is reasonably certain that the job completed successfully,
740 then it should return the job-state as 'completed'. Also the server can keep the job in its "job history" long after the
741 job is no longer in the device. Then a user could query the server and see that the job was in the 'completed' state
742 and completed as specified by the jobs "time-at-completed" time, which would be the same as the server submitted
743 the job to the device.

744 An alternative is for the server to respond to the client before or while sending the job to the device, instead of
745 waiting until the server has finished sending the job to the device. In this case, the server can return the job's state
746 as 'pending' with the 'job-outgoing' value in the job's "job-state-reasons" attribute.

747 If the server doesn't know for sure whether the job completed successfully (or at all), it could return the (out-of-
748 band) 'unknown' value.

749 On the other hand, if the server is able to query the device and/or setup some sort of event notification that the
750 device initiates when the job makes state transitions, then the server can return the current job state in the Print-Job

751 response and in subsequent queries because the server knows what the job state is in the device (or can query the
752 device).

753 All of these alternatives depend on implementation of the server and the device.

754 3.2.2 Get-Printer-Attributes operation

755 If a Printer supports the "printer-make-and-model" attribute and returns the .INF file model name of the printer in
756 that attribute, the Microsoft client will automatically install the correct driver (if available).

757 Clients which poll periodically for printer status or queued-job-count should use the "requested-attributes"
758 operation attribute to limit the scope of the query in order to save Printer and network resources.

759 3.2.3 Get-Jobs operation

760 3.2.3.1 Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?

761 In [RFC2911] section 3.2.6.1 'Get-Jobs Request', if the attribute 'my-jobs' is present and set to TRUE, MUST the
762 'requesting-user-name' attribute be there too, and if it's not present what should the IPP printer do?

763 [RFC2911] Section 8.3 describes the various cases of "requesting-user-name" being present or not for any
764 operation. If the client does not supply a value for "requesting-user-name", the printer MUST assume that the
765 client is supplying some anonymous name, such as "anonymous".

766 3.2.3.2 Why is there a "limit" attribute in the Get-Jobs operation?

767 When using the Get-Jobs operation a client implementer might choose to limit the number of jobs that the client
768 shows on the first screenful. For example, if its UI can only display 50 jobs, it can defend itself against a printer that
769 would otherwise return 500 jobs, perhaps taking a long time on a slow dial-up line. The client can then go and ask
770 for a larger number of jobs in the background, while showing the user the first 50 jobs. Since the job history is
771 returned in reverse order, namely the most recently completed jobs are returned first, the user is most likely
772 interested in the first jobs that are returned. Limiting the number of jobs may be especially useful for a client that is
773 requesting 'completed' jobs from a printer that keeps a long job history. Clients that don't mind sometimes getting
774 very large responses, can omit the "limit" attribute in their Get-Jobs requests.

775 3.2.4 Create-Job operation

776 A Printer may respond to a Create-Job operation with "job-state" 'pending' or 'pending-held' and " job-state-
777 reason" 'job-data-insufficient' to indicate that operation has been accepted by the Printer, but the Printer is
778 expecting additional document data before it can move the job into the 'processing' state. Alternatively, it may
779 respond with "job-state" 'processing' and "job-state-reason" 'job-incoming' to indicate that the Create-Job
780 operation has been accepted by the Printer, but the Printer is expecting additional Send-Document and/or Send-

781 URI operations and/or is accessing/accepting document data. The second alternative is for non-spooling Printers
782 that don't implement the 'pending' state.

783 Should the server wait for the "last-document" operation attribute set to 'true' before starting to "process" the job?

784 It depends on implementation. Some servers spool the entire job, including all document data, before starting to
785 process, so such an implementation would wait for the "last-document" before starting to process the job. If the
786 time-out occurs without the "last-document", then the server takes one of the indicated actions in section 3.3.1 in
787 the [RFC2911] document. Other servers will start to process document data as soon as they have some. These
788 are the so-called "non-spooling" printers. Currently, there isn't a way for a client to determine whether the Printer
789 will spool all the data or will start to process (and print) as soon as it has some data.

790 3.3 Job Operations

791 3.3.1 Validate-Job

792 The Validate-Job operation has been designed so that its implementation may be a part of the Print-Job operation.
793 Therefore, requiring Validate-Job is not a burden on implementers. Also it is useful for client's to be able to count
794 on its presence in all conformance implementations, so that the client can determine before sending a long
795 document, whether the job will be accepted by the IPP Printer or not.

796 3.3.2 Restart-Job

797 The Restart-Job operation allows the reprocessing of a completed job. Some jobs store the document data on the
798 printer. Jobs created using the Print-Job operation are an example. It is required that the printer retains the job
799 data after the job has moved to a 'completed state' in order for the Restart-Job operation to succeed.

800 Some jobs contain only a reference to the job data. A job created using the Print-URI is an example of such a job.
801 When the Restart-Job operation is issued the job is reprocessed. The job data **MUST** be retrieved again to print
802 the job.

803 It is possible that a job fails while attempting to access the print data. When such a job is the target of a Restart-
804 Job the Printer **SHALL** attempt to retrieve the job data again.

805 4 Object Attributes

806 4.1 Attribute Syntax's

807 4.1.1 The 'none' value for empty sets (Issue 1.37)

808 [RFC2911] states that the 'none' value should be used as the value of a 1setOf when the set is empty. In most
809 cases, sets that are potentially empty contain keywords so the keyword 'none' is used, but for the 3 finishings
810 attributes, the values are enums and thus the empty set is represented by the enum 3. Currently there are no other
811 attributes with 1setOf values, which can be empty and can contain values that are not keywords. This exception
812 requires special code and is a potential place for bugs. It would have been better if we had chosen an out-of-band
813 value, either "no-value" or some new value, such as 'none'. Since we didn't, implementations have to deal with the
814 different representations of 'none', depending on the attribute syntax.

815 4.1.2 Multi-valued attributes (Issue 1.31)

816 What is the attribute syntax for a multi-valued attribute? Since some attributes support values in more than one
817 data type, such as "media", "job-hold-until", and "job-sheets", IPP semantics associate the attribute syntax with
818 each value, not with the attribute as a whole. The protocol associates the attribute syntax tag with each value.
819 Don't be fooled, just because the attribute syntax tag comes before the attribute keyword. All attribute values after
820 the first have a zero length attribute keyword as the indication of a subsequent value of the same attribute.

821 4.1.3 Case Sensitivity in URIs (issue 1.6)

822 IPP client and server implementations must be aware of the diverse uppercase/lowercase nature of URIs. RFC
823 2396 defines URL schemes and Host names as case insensitive but reminds us that the rest of the URL may well
824 demonstrate case sensitivity. When creating URL's for fields where the choice is completely arbitrary, it is
825 probably best to select lower case. However, this cannot be guaranteed and implementations MUST NOT rely on
826 any fields being case-sensitive or case-insensitive in the URL beyond the URL scheme and host name fields.

827 The reason that the IPP specification does not make any restrictions on URIs, is so that implementations of IPP
828 may use off-the-shelf components that conform to the standards that define URIs, such as RFC 2396 and the
829 HTTP/1.1 specifications [RFC2616]. See these specifications for rules of matching, comparison, and case-
830 sensitivity.

831 It is also recommended that System Administrators and implementations avoid creating URLs for different printers
832 that differ only in their case. For example, don't have Printer1 and printer1 as two different IPP Printers.

833 Example of equivalent URI's

834 `http://abc.com:80/~smith/home.html`

835 `http://ABC.com/%7Esmith/home.html`

836 `http:/ABC.com:/%7esmith/home.html`

837 Example of equivalent URI's using the IPP scheme

838 `ipp://abc.com:631/~smith/home.html`

839 ipp://ABC.com/%7Esmith/home.html

840 http://ABC.com:631/%7esmith/home.html

841 The HTTP/1.1 specification [RFC2616] contains more details on comparing URLs.

842 4.1.4 Maximum length for xxxWithLanguage and xxxWithoutLanguage

843 The 'textWithLanguage' and 'nameWithLanguage' are compound syntaxes that have two components. The first
844 component is the 'language' component that can contain up to 63 octets. The second component is the 'text' or
845 'name' component. The maximum length of these are 1023 octets and 255 octets respectively. The definition of
846 attributes with either syntax may further restrict the length. (e.g. printer-name (name(127)))

847 The length of the 'language' component has no effect on the allowable length of 'text' in 'textWithLanguage' or the
848 length of 'name' in 'nameWithLanguage'

849 4.2 Job Template Attributes

850 4.2.1 multiple-document-handling(type2 keyword)

851 4.2.1.1 Support of multiple document jobs

852 IPP/1.0 is silent on which of the four effects an implementation would perform if it supports Create-Job, but does
853 not support "multiple-document-handling" or multiple documents per job. IPP/1.1 was changed so that a Printer
854 could support Create-Job without having to support multiple document jobs. The "multiple-document-jobs-
855 supported" (boolean) Printer description attribute was added to IPP/1.1 along with the 'server-error-multiple-
856 document-jobs-not-supported' status code for a Printer to indicate whether or not it supports multiple document
857 jobs, when it supports the Create-Job operation. Also IPP/1.1 was clarified that the Printer MUST support the
858 "multiple-document-handling" (type2 keyword) Job Template attribute with at least one value if the Printer supports
859 multiple documents per job.

860 4.3 Job Description Attributes

861 4.3.1 Getting the date and time of day

862 The "date-time-at-creation", "date-time-at-processing", and "date-time-at-completed" attributes are returned as
863 dateTime syntax. There are various ways for a Printer to get the date and time of day. Some suggestions:

864 1. A Printer can get time from an NTP timeserver if there's one reachable on the network . See RFC
865 1305. Also DHCP option 32 in RFC 2132 returns the IP address of the NTP server.

866 2. Get the date and time at startup from a human operator

- 867 3. Have an operator set the date and time using a web administrative interface
- 868 4. Get the date and time from incoming HTTP requests, though the problems of spoofing need to be
869 considered. Perhaps comparing several HTTP requests could reduce the chances of spoofing.
- 870 5. Internal date time clock battery driven.
- 871 6. Query "<http://tycho.usno.navy.mil/cgi-bin/timer.pl>"

872 4.4 Printer Description Attributes

873 4.4.1 printer-state-reasons (1setOf type2 keyword)

874 4.4.1.1 Is a suffix needed for the "printer-state-reasons" 'none' value (Issue 3.6)?

875 The values of the "printer-state-reasons" MAY have suffixes of '-report', '-warning', and '-error'. If none of these
876 suffixes is included, the meaning is the same as 'error', i.e., the Printer is stopped. However, for the 'none' value it
877 is RECOMMENDED that no suffix be included, even though the Printer is not stopped. However, some
878 implementations do include the '-report' suffix, i.e., return 'none-report'. There is no semantic difference between
879 the "printer-state-reasons" of 'none', 'none-report', and 'none-error'. They all mean that no additional information
880 on the printer's state is available.

881 4.4.2 queued-job-count (integer(0:MAX))

882 4.4.2.1 Why is "queued-job-count" RECOMMENDED (Issue 1.14)?

883 The reason that "queued-job-count" is RECOMMENDED, is that some clients look at that attribute alone when
884 summarizing the status of a list of printers, instead of doing a Get-Jobs to determine the number of jobs in the
885 queue. Implementations that fail to support the "queued-job-count" will cause that client to display 0 jobs when
886 there are actually queued jobs.

887 We would have made it a REQUIRED Printer attribute, but some implementations had already been completed
888 before the issue was raised, so making it a SHOULD was a compromise.

889 4.4.2.2 Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?

890 The "queued-job-count" is not a good measure of how busy the printer is when there are held jobs. A future
891 registration could be to add a "held-job-count" (or an "active-job-count") Printer Description attribute if experience
892 shows that such an attribute (combination) is needed to quickly indicate how busy a printer really is.

893 4.4.3 printer-current-time (dateTime)

894 A Printer implementation MAY support this attribute by obtaining the date and time by any number of
895 implementation-dependent means at startup or subsequently. Examples include:

- 896 1. an internal date time clock,
- 897 2. from the operator at startup using the console,
- 898 3. from an operator using an administrative web page,
- 899 4. from HTTP headers supplied in client requests,
- 900 5. use HTTP to query "<http://tycho.usno.navy.mil/cgi-bin/timer.pl>"
- 901 6. from the network, using NTP [RFC1305] or DHCP option 32 [RFC2132] that returns the IP address of
902 the NTP server.

903 If an implementation supports this attribute by obtaining the current time from the network (at startup or later), but
904 the time is not available, then the implementation MUST return the value of this attribute using the out-of-band 'no-
905 value' meaning not configured. See the beginning of section 4.1.

906 Since the new "date-and-time-at-xxx" Job Description attributes refer to the "printer-current-time", they will be
907 covered also.

908 4.4.4 Printer-uri

909 Must the operational attribute for printer-uri match one of the values in "printer-uri-supported"?

910 A forgiving printer implementation would not reject the operation. But the implementation has its rights to reject a
911 printer or job operation if the operational attribute printer-uri is not a value of the printer-uri-supported. The printer
912 may not be improperly configured. The request obviously reached the printer. The printer could treat the printer-uri
913 as the logical equivalent of a value in the printer-uri-supported. It would be implementation dependent for which
914 value, and associated security policy, would apply. This does also apply to a job object specified with a printer-uri
915 and job-id, or with a job-uri. See section 4.1.3 for how to compare URI's.

916 4.5 Empty Jobs

917 The IPP object model does not prohibit a job that contains no documents. Such a job may be created in a number
918 of ways including a 'create-job' followed by an 'add-document' that contains no data and has the 'last-document'
919 flag set.

920 An empty job is processed just as any other job. The operation that "closes" an empty job is not rejected because
921 the job is empty. If no other conditions exist, other than the job is empty, the response to the operation will
922 indicate success. After the job is scheduled and processed, the job state SHALL be 'completed'.

9 2 3 There will be some variation in the value(s) of the "job-state-reasons" attribute. It is required that if no conditions,
9 2 4 other than the job being empty, exist the "job-state-reasons" SHALL include the 'completed-successfully'. If other
9 2 5 conditions existed, the 'completed-with-warnings' or 'completed-with-errors' values may be used.

9 2 6 **5 Directory Considerations**

9 2 7 5.1 General Directory Schema Considerations

9 2 8 The [RFC2911] document lists RECOMMENDED and OPTIONAL Printer object attributes for directory
9 2 9 schemas. See [RFC2911] APPENDIX E: Generic Directory Schema.

9 3 0 The SLP printer template is defined in the "Definition of the Printer Abstract Service Type v2.0" document [svrloc-
9 3 1 printer]. The LDAP printer template is defined in the "Internet Printing Protocol (IPP): LDAP Schema for Printer
9 3 2 Services" document [ldap-printer]. Both documents systematically add "printer-" to any attribute that doesn't
9 3 3 already start with "printer-" in order to keep the printer directory attributes distinct from other directory attributes.
9 3 4 Also, instead of using "printer-uri-supported", "uri-authentication-supported", and "uri-security-supported", they
9 3 5 use a "printer-xri-supported" attribute with special syntax to contain all of the same information in a single attribute.

9 3 6 5.2 IPP Printer with a DNS name

9 3 7 If the IPP printer has a DNS name should there be at least two values for the printer-uri-supported attribute. One
9 3 8 URL with the fully qualified DNS name the other with the IP address in the URL?

9 3 9 The printer may contain one or the other or both. It's up to the administrator to configure this attribute.

9 4 0 **6 Security Considerations**

9 4 1 This section corresponds to the RFC2911 Section 8 "Security Considerations.

9 4 2 6.1 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)

9 4 3 The following clarification was added to [RFC2911] section 8.5:

9 4 4 8.5 Queries on jobs submitted using non-IPP protocols

9 4 5 If the device that an IPP Printer is representing is able to accept jobs using other job submission protocols in
9 4 6 addition to IPP, it is RECOMMEND that such an implementation at least allow such "foreign" jobs to be queried
9 4 7 using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an implementation NEED NOT support all of
9 4 8 the same IPP job attributes as for IPP jobs. The IPP object returns the 'unknown' out-of-band value for any
9 4 9 requested attribute of a foreign job that is supported for IPP jobs, but not for foreign jobs.

9 5 0 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such "foreign jobs",
9 5 1 if possible, so that they may be targets of other IPP operations, such as Get-Job-Attributes and Cancel-Job. Such

an implementation also needs to deal with the problem of authentication of such foreign jobs. One approach would be to treat all such foreign jobs as belonging to users other than the user of the IPP client. Another approach would be for the foreign job to belong to 'anonymous'. Only if the IPP client has been authenticated as an operator or administrator of the IPP Printer object, could the foreign jobs be queried by an IPP request. Alternatively, if the security policy were to allow users to query other users' jobs, then the foreign jobs would also be visible to an end-user IPP client using Get-Jobs and Get-Job-Attributes.

Thus IPP MAY be implemented as a "universal" protocol that provides access to jobs submitted with any job submission protocol. As IPP becomes widely implemented, providing a more universal access makes sense.

7 Encoding and Transport

This section discusses various aspects of IPP/1.1 Encoding and Transport [RFC2910].

A server is not required to send a response until after it has received the client's entire request. Hence, a client must not expect a response until after it has sent the entire request. However, we recommend that the server return a response as soon as possible if an error is detected while the client is still sending the data, rather than waiting until all of the data is received. Therefore, we also recommend that a client listen for an error response that an IPP server MAY send before it receives all the data. In this case a client, if chunking the data, can send a premature zero-length chunk to end the request before sending all the data (and so the client can keep the connection open for other requests, rather than closing it). If the request is blocked for some reason, a client MAY determine the reason by opening another connection to query the server using Get-Printer-Attributes.

IPP, by design, uses TCP's built-in flow control mechanisms [RFC 793] to throttle clients when Printers are busy. Therefore, it is perfectly normal for an IPP client transmitting a Job to be blocked for a really long time. Accordingly, socket timeouts must be avoided. Some socket implementations have a timeout option, which specifies how long a write operation on a socket can be blocked before it times out and the blocking ends. A client should set this option for infinite timeout when transmitting Job submissions.

Some IPP client applications might be able to perform other useful work while a Job transmission is blocked. For example, the client may have other jobs that it could transmit to other Printers simultaneously. A client may have a GUI, which must remain responsive to the user while the Job transmission is blocked. These clients should be designed to spawn a thread to handle the Job transmission at its own pace, leaving the main application free to do other work. Alternatively, single-threaded applications could use non-blocking I/O.

Some Printer conditions, such as jam or lack of paper, could cause a client to be blocked indefinitely. Clients may open additional connections to the Printer to Get-Printer-Attributes, determine the state of the device, alert a user if the printer is stopped, and let a user decide whether to abort the job transmission or not.

In the following sections, there are tables of all HTTP headers, which describe their use in an IPP client or server. The following is an explanation of each column in these tables.

- the "header" column contains the name of a header
- the "request/client" column indicates whether a client sends the header.

- 987 – the "request/ server" column indicates whether a server supports the header when received.
 988 – the "response/ server" column indicates whether a server sends the header.
 989 – the "response /client" column indicates whether a client supports the header when received.
 990 – the "values and conditions" column specifies the allowed header values and the conditions for the
 991 header to be present in a request/response.

992 The table for "request headers" does not have columns for responses, and the table for "response headers" does
 993 not have columns for requests.

994 The following is an explanation of the values in the "request/client" and "response/ server" columns.

- 995 – **must:** the client or server **MUST** send the header,
 996 – **must-if:** the client or server **MUST** send the header when the condition described in the "values and
 997 conditions" column is met,
 998 – **may:** the client or server **MAY** send the header
 999 – **not:** the client or server **SHOULD NOT** send the header. It is not relevant to an IPP implementation.

000 The following is an explanation of the values in the "response/client" and "request/ server" columns.

- 001 – **must:** the client or server **MUST** support the header,
 002 – **may:** the client or server **MAY** support the header
 003 – **not:** the client or server **SHOULD NOT** support the header. It is not relevant to an IPP
 004 implementation.

005 7.1 General Headers

006 The following is a table for the general headers.

General-Header	Request		Response		Values and Conditions
	Client	Server	Server	Client	
Cache-Control	must	not	must	not	"no-cache" only
Connection	must-if	must	must-if	must	"close" only. Both client and server SHOULD keep a connection for the duration of a sequence of operations. The client and server MUST include this header for the last operation in such a sequence.
Date	may	may	must	may	per RFC 1123 [RFC1123] from RFC 2616 [RFC2616]
Pragma	must	not	must	not	"no-cache" only
Transfer-Encoding	must-if	must	must-if	must	"chunked" only . Header MUST be present if Content-Length is absent.
Upgrade	not	not	not	not	

General-Header	Request		Response		Values and Conditions
	Client	Server	Server	Client	
Via	not	not	not	not	

007 7.2 Request Headers

008 The following is a table for the request headers.

Request-Header	Client	Server	Request Values and Conditions
Accept	may	must	"application/ipp" only. This value is the default if the client omits it
Accept-Charset	not	not	Charset information is within the application/ipp entity
Accept-Encoding	may	must	empty and per RFC 2616 [RFC2616] and IANA registry for content-codings
Accept-Language	not	not	language information is within the application/ipp entity
Authorization	must-if	must	per RFC 2616. A client MUST send this header when it receives a 401 "Unauthorized" response and does not receive a "Proxy-Authenticate" header.
From	not	not	per RFC 2616. Because RFC recommends sending this header only with the user's approval, it is not very useful
Host	must	must	per RFC 2616
If-Match	not	not	
If-Modified-Since	not	not	
If-None-Match	not	not	
If-Range	not	not	
If-Unmodified-Since	not	not	
Max-Forwards	not	not	
Proxy- Authorization	must-if	not	per RFC 2616. A client MUST send this header when it receives a 401 "Unauthorized" response and a "Proxy-Authenticate" header.
Range	not	not	

Request-Header	Client	Server	Request Values and Conditions
Referrer	not	not	
User-Agent	not	not	

009 7.3 Response Headers

010 The following is a table for the request headers.

Response-Header	Server	Client	Response Values and Conditions
Accept-Ranges	not	not	
Age	not	not	
Location	must-if	may	per RFC 2616. When URI needs redirection.
Proxy-Authenticate	not	must	per RFC 2616
Public	may	may	per RFC 2616
Retry-After	may	may	per RFC 2616
Server	not	not	
Vary	not	not	
Warning	may	may	per RFC 2616
WWW-Authenticate	must-if	must	per RFC 2616. When a server needs to authenticate a client.

011 7.4 Entity Headers

012 The following is a table for the entity headers.

Entity-Header	Request		Response		Values and Conditions
	Client	Server	Server	Client	
Allow	not	not	not	not	
Content-Base	not	not	not	not	
Content-Encoding	may	must	must	must	per RFC 2616 and IANA registry for content codings.
Content-Language	not	not	not	not	Application/ipp handles language

Entity-Header	Request		Response		Values and Conditions
	Client	Server	Server	Client	
Content-Length	must-if	must	must-if	must	the length of the message-body per RFC 2616. Header MUST be present if Transfer-Encoding is absent..
Content-Location	not	not	not	not	
Content-MD5	may	may	may	may	per RFC 2616
Content-Range	not	not	not	not	
Content-Type	must	must	must	must	"application/ipp" only
ETag	not	not	not	not	
Expires	not	not	not	not	
Last-Modified	not	not	not	not	

013 7.5 Optional support for HTTP/1.0

014 IPP implementations consist of an HTTP layer and an IPP layer. In the following discussion, the term "client" refers
 015 to the HTTP client layer and the term "server" refers to the HTTP server layer. The Encoding and Transport
 016 document [RFC2910] requires that HTTP 1.1 MUST be supported by all clients and all servers. However, a
 017 client and/or a server implementation may choose to also support HTTP 1.0.

018 – This option means that a server may choose to communicate with a (non-conforming) client that only supports
 019 HTTP 1.0. In such cases the server should not use any HTTP 1.1 specific parameters or features and should
 020 respond using HTTP version number 1.0.

021 – This option also means that a client may choose to communicate with a (non-conforming) server that only
 022 supports HTTP 1.0. In such cases, if the server responds with an HTTP 'unsupported version number' to an
 023 HTTP 1.1 request, the client should retry using HTTP version number 1.0.

024 7.6 HTTP/1.1 Chunking

025 7.6.1 Disabling IPP Server Response Chunking

026 Clients MUST anticipate that the HTTP/1.1 server may chunk responses and MUST accept them in responses.
 027 However, a (non-conforming) HTTP client that is unable to accept chunked responses may attempt to request an
 028 HTTP 1.1 server not to use chunking in its response to an operation by using the following HTTP header:

029 TE: identity

030 This mechanism should not be used by a server to disable a client from chunking a request, since chunking of
031 document data is an important feature for clients to send long documents.

032 7.6.2 Warning About the Support of Chunked Requests

033 This section describes some problems with the use of chunked requests and HTTP/1.1 servers.

034 The HTTP/1.1 standard [RFC2616] requires that conforming servers support chunked requests for any method.
035 However, in spite of this requirement, some HTTP/1.1 implementations support chunked responses in the GET
036 method, but do not support chunked POST method requests. Some HTTP/1.1 implementations that support CGI
037 scripts [CGI] and/or servlets [Servlet] require that the client supply a Content-Length. These implementations
038 might reject a chunked POST method and return a 411 status code (Length Required), might attempt to buffer the
039 request and run out of room returning a 413 status code (Request Entity Too Large), or might successfully accept
040 the chunked request.

041 Because of this lack of conformance of HTTP servers to the HTTP/1.1 standard, the IPP standard [RFC2910]
042 REQUIRES that a conforming IPP Printer object implementation support chunked requests and that conforming
043 clients accept chunked responses. Therefore, IPP object implementers are warned to seek HTTP server
044 implementations that support chunked POST requests in order to conform to the IPP standard and/or use
045 implementation techniques that support chunked POST requests.

046 7.7 HTTP "continue" interim response

047 IPP Clients must be prepared at any time to receive an interim response with a status code of '100 Continue' This
048 includes receiving this response prior to sending an IPP request.

049 8 References

050 [CGI]

051 CGI/1.1 (<http://www.ietf.org/internet-drafts/draft-coar-cgi-v11-00.txt>).

052 [ldap-printer]

053 Fleming, P., Jones, K., Lewis, H., McDonald, I., "Internet Printing Protocol (IPP): LDAP Schema for
054 Printer Services", <draft-ietf-ipp-ldap-printer-schema-01.txt>, work in progress, April 27, 2000.

055 [RFC793]

056 J. Postel, "Transmission Control Protocol", RFC 793.

057 [RFC1123]

058 Braden, S., "Requirements for Internet Hosts - Application and Support", RFC 1123, October, 1989.

059 [RFC2026]

060 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

061 [RFC2119]

062 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

063 [RFC2396]

064 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC

065 2396, August 1998.

066 [RFC2565]

067 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and

068 Semantics", RFC 2566, April 1999.

069 [RFC2566]

070 Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.0: Encoding and Transport",

071 RFC 2565, April 1999.

072 [RFC2567]

073 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-03.txt, November, 1998.

074 [RFC2568]

075 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol", RFC

076 2568, April 1999.

077 [RFC2569]

078 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC 2569,

079 April 1999.

080 [RFC2616]

081 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer

082 Protocol - HTTP/1.1", RFC 2616, June 1999.

083 [RFC2910]

084 Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.0: Encoding and Transport",

085 RFC 2910, September, 2000.

086 [RFC2911]

087 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and

088 Semantics", RFC 2911, September, 2000.

089 [Servlet]

090 Servlet Specification Version 2.1 (<http://java.sun.com/products/servlet/2.1/index.html>).

091 [svrloc-printer]

092 St. Pierre, P., Isaacson, S., McDonald, I., "Definition of the Printer Abstract Service Type v2.0", <draft-
093 ietf-svrloc-printer-scheme-06.txt>, work in progress, March 8, 2000.

094 [SSL]
095 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.

096 **9 Authors' Address**

097 Thomas N. Hastings
098 Xerox Corporation
099 701 Aviation Blvd.
100 El Segundo, CA 90245
101 hastings@cp10.es.xerox.com

102
103 Carl-Uno Manros
104 Xerox Corporation
105 701 Aviation Blvd.
106 El Segundo, CA 90245
107 manros@cp10.es.xerox.com

108
109 Carl Kugler
110 Mail Stop 003G
111 IBM Printing Systems Co
112 6300 Diagonal Hwy
113 Boulder CO 80301
114 Kugler@us.ibm.com

115
116 Henrik Holst
117 i-data Printing Systems
118 Vadstrupvej 35-43
119 2880 Bagsvaerd, Denmark
120 hh@i-data.com

121
122 Peter Zehler
123 Xerox Corporation
124 800 Philips Road
125 Webster, NY 14580
126
127 peter.zehler@usa.xerox.com

128 **10 Notices**

1 2 9 The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be
1 3 0 claimed to pertain to the implementation or use of the technology described in this document or the extent to which
1 3 1 any license under such rights might or might not be available; neither does it represent that it has made any effort to
1 3 2 identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and
1 3 3 standards-related documentation can be found in BCP-11[BCP-11]. Copies of claims of rights made available for
1 3 4 publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a
1 3 5 general license or permission for the use of such proprietary rights by implementers or users of this specification can
1 3 6 be obtained from the IETF Secretariat.

1 3 7 The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other
1 3 8 proprietary rights, which may cover technology that may be required to practice this standard. Please address the
1 3 9 information to the IETF Executive Director.

1 4 0 Copyright (C) The Internet Society (1999). All Rights Reserved

1 4 1 This document and translations of it may be copied and furnished to others, and derivative works that comment on
1 4 2 or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole
1 4 3 or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included
1 4 4 on all such copies and derivative works. However, this document itself may not be modified in any way, such as
1 4 5 by removing the copyright notice or references to the Internet Society or other Internet organizations, except as
1 4 6 needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the
1 4 7 Internet Standards process must be followed, or as required to translate it into languages other than English.

1 4 8 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its
1 4 9 successors or assigns.

1 5 0 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET
1 5 1 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,
1 5 2 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF
1 5 3 THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
1 5 4 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1 5 5 **11 Change History (to be removed at time of RFC publishing)**

1 5 6 The change history is in *reverse* chronological order:

1 5 7 11.1 Changes from 000509 to 000530

1 5 8 The following changes were made to the 5/09/00 version to make the 5/30/00 version:

- 1 5 9 1. Added section 5.1 on General Directory Considerations which includes references to SLP and LDPA Printer
1 6 0 schemas and their introduction of the "printer-xri-supported" attribute which combines "printer-uri-supported",
1 6 1 "uri-security-supported", and "uri-authentication-supported" attributes.

1 6 2 11.2 Changes from 990927 to 000509

1 6 3 The following changes were made to the 9/27/99 version to make the 5/09/00 version:

- 1 6 4 1. Table 5 - Corrected some attributes returned by Send-Document and Send-URI to be the same as Print-Job
1 6 5 as in [RFC2911].
- 1 6 6 2. Corrected several uses of 'client-error-bad-syntax' to be 'client-error-bad-request' as in the [RFC2911].
- 1 6 7 3. Added section 3.1.3.1.8.1 to clarify what Resume-Printer does if the Printer is unable to resume the output
1 6 8 device and section 3.1.3.1.8.2 about the "printer-state" for such a condition.
- 1 6 9 4. Added section 3.3.2 to indicate that on a Restart-Job that a Printer MUST re-fetch the document data when
1 7 0 the job was created with Print-URI or Send-URI.
- 1 7 1 5. Section 4.1.4 - clarified that the length field for 'textWithLanguage' and 'nameWithLanguage' does *not* include
1 7 2 the language field, so that the same maximum length of the data applies to the WithLanguage as the
1 7 3 WithoutLanguage types, not counting the language field.
- 1 7 4 6. Added section 4.5 about empty jobs, i.e., with no documents. They are processed as any other job, possibly
1 7 5 producing start and/or end sheets.

1 7 6 11.3 Changes from 990914 to 990927

- 1 7 7 1. Add comments about this document is also IPP/1.0 relevant.
- 1 7 8 2. Section 4.1.3: Add more examples of URI's with the port 631 and the ipp scheme.
- 1 7 9 3. Section 4.4.3: Move the DNS stuff to the 'how to compare URI's.
- 1 8 0 4. Section 4.4.3.2: Swap lines, first tell about the forgiven printer and then what the printer is allowed to do.
- 1 8 1 5. Fixed some errors in the Summary Attribute tables 1-5 and broke them into five portrait tables, so that it can
1 8 2 be made into plain text for INTERNET-DRAFTS.

1 8 3 11.4 Changes from 990726 to 990914:

- 1 8 4 1. Added IPP/1.1 operations and attributes to table 1.
- 1 8 5 2. Validate version: Added text and table from issue 32
- 1 8 6 3. Printer-uri-supported: Added section 4.4.4
- 1 8 7 4. Added IPP/1.1 operations to section 3.1.2.1.4.3

- 188 5. Added answer to question "Should the server wait for the "last-document" operation attribute set to 'true'
189 before starting to "process" the job?" in section 3.2.4
- 190 6. Changed 'server-error-uri-scheme-not-supported' to 'client-error-uri-scheme-not-supported' in section
191 3.1.2.1.5 when talking about the 'document-uri' attribute.
- 192 7. Added 'Suggested Operation Processing Steps' and 'Suggested Additional Processing Steps for Operations
193 that Create/Validate Jobs and Add Document' flow-chart overview.
- 194 11.5 Changes to produce the February 12, 1999 version from the January 8, 1999 version:
- 195 1. Section 2.2.1.5: added check for document not found or accessible in Print-URI and Send-URI
- 196 2. Section 3.6.2: Clarified that the IPP standard requires that servers MUST accept chunked requests and
197 that clients MUST accept chunked responses, in spite of the lack of conformance of HTTP servers to the
198 HTTP/1.1 requirement to support chunking.
- 199 11.6 Changes to produce the January 8, 1999 version from the December 6, 1998 version:
- 200 1. Added section 3.6.2: Warning About the Use of Chunked Requests with CGI Script Implementations
- 201 2. Section 2.2.1.2: changed "printer-operations-supported" to "operations-supported".
- 202 3. Section 2.2.1.6: changed "job-media-supported" to "job-media-sheets-supported"
- 203 4. Section 2.2.3: separated the validation checks for variable length attributes into two separate tests: one for
204 correct attribute syntax and one for correct length.
- 205 5. Section 2.2.3: changed "multiple-document-handling-supported" to "printer-resolution-supported"
- 206 6. Section 2.6.1: recommended that an IPP object also support US-ASCII charset.
- 207 7. Section 3: Clarified that a server is not required to send a response until after it has received the client's
208 entire request, but recommend that the server return a response as soon as possible if an error is detected
209 while the client is still sending the data, rather than waiting until all of the data is received. Also
210 recommended that a client listen for an error response that an IPP server MAY send before it receives all
211 the data.
- 212 11.7 Changes to produce the December 6, 1998 version from the November 16, 1998 version:
- 213 Included all of the remaining agreed issues raised before the November 16, 1998 production of the Internet-Drafts
214 for IPP/1.0 that included adding explanations to the Implementers Guide.

215 Changes from 990422 to 990726:

- 216 1. Encoding and Transport: Address issues 4, 5, 20 from Issues-raised-at-Bake-Off2.doc
- 217 2. Decide whether to accept or reject the request: discuss issues 6, 9, 10
- 218 3. Get-Printer-Attributes: add notes about printer-make-and-model and .INF files; issue 7
- 219 4. Create-Job: clarify job-incoming vs. data-insufficient; issue 13
- 220 5. Get-Printer Attributes: polling -- issue 16
- 221 6. Job Description Attributes: ways to get time; issue 17
- 222 7. Validate the values of the Job Template Attributes: clarify zero-length keywords; issue 22
- 223 8. Validate Optional Operation Attributes: Note about checking for compression in IPP/1.0; issue 28
- 224 9. Validate version number: advantages to backward compatibility; issue 33
- 225 10. Note: examples for issue 2 seem to be covered sufficiently in the new MOD doc.