

1 INTERNET-DRAFT
2 draft-ietf-ipp-implementers-guide-v11-01.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
May 30, 2000

15 Internet Printing Protocol/1.1: Implementer's Guide

16 Copyright (C) The Internet Society (2000). All Rights Reserved.

17 Status of this Memo

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

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

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

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

27

28 Abstract

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

37 The full set of IPP documents includes:

38 Design Goals for an Internet Printing Protocol [RFC2567]

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

40 Internet Printing Protocol/1.1: Model and Semantics [IPP-MOD]

41 Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]

42 Mapping between LPD and IPP Protocols [RFC2569]

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

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

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

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

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

60

61		TABLE OF CONTENTS	
62	1	INTRODUCTION	6
63	1.1	CONFORMANCE LANGUAGE	6
64	1.2	OTHER TERMINOLOGY	6
65	1.3	ISSUES RAISED FROM INTEROPERABILITY BAKE OFFS	6
66	2	IPP OBJECTS	7
67	3	IPP OPERATIONS	8
68	3.1	COMMON SEMANTICS	8
69	3.1.1	<i>Summary of Operation Attributes</i>	8
70	3.1.2	<i>Suggested Operation Processing Steps for IPP Objects</i>	12
71	3.1.2.1	Suggested Operation Processing Steps for all Operations	13
72	3.1.2.1.1	Validate version number	13
73	3.1.2.1.2	Validate operation identifier	14
74	3.1.2.1.3	Validate the request identifier	15
75	3.1.2.1.4	Validate attribute group and attribute presence and order	15
76	3.1.2.1.4.1	Validate the presence and order of attribute groups	15
77	3.1.2.1.4.2	Ignore unknown attribute groups in the expected position	15
78	3.1.2.1.4.3	Validate the presence of a single occurrence of required Operation attributes	16
79	3.1.2.1.5	Validate the values of the REQUIRED Operation attributes	22
80	3.1.2.1.6	Validate the values of the OPTIONAL Operation attributes	25
81	3.1.2.2	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add Documents	28
82	3.1.2.2.1	Default "ipp-attribute-fidelity" if not supplied	28
83	3.1.2.2.2	Check that the Printer object is accepting jobs	28
84	3.1.2.2.3	Validate the values of the Job Template attributes	29
85	3.1.2.3	Algorithm for job validation	29
86	3.1.2.3.1	Check for conflicting Job Template attributes values	34
87	3.1.2.3.2	Decide whether to REJECT the request	34
88	3.1.2.3.3	For the Validate-Job operation, RETURN one of the success status codes	35
89	3.1.2.3.4	Create the Job object with attributes to support	36
90	3.1.2.3.5	Return one of the success status codes	37
91	3.1.2.3.6	Accept appended Document Content	37
92	3.1.2.3.7	Scheduling and Starting to Process the Job	37
93	3.1.2.3.8	Completing the Job	38
94	3.1.2.3.9	Destroying the Job after completion	38
95	3.1.2.3.10	Interaction with "ipp-attribute-fidelity"	38
96	3.1.2.3.11	Character set code conversion support	38
97	3.1.2.3.12	What charset to return when an unsupported charset is requested (Issue 1.19)?	39
98	3.1.2.3.13	Natural Language Override (NLO)	40
99	3.1.3	<i>Status codes returned by operation</i>	41
100	3.1.3.1	Printer Operations	41
101	3.1.3.1.1	Print-Job	41
102	3.1.3.1.2	Print-URI	43
103	3.1.3.1.3	Validate-Job	43
104	3.1.3.1.4	Create-Job	43
105	3.1.3.1.5	Get-Printer-Attributes	43
106	3.1.3.1.6	Get-Jobs	44
107	3.1.3.1.7	Pause-Printer	45
108	3.1.3.1.8	Resume-Printer	45
109	3.1.3.1.8.1	What about Printers unable to change state due to an error condition?	46
110	3.1.3.1.8.2	How is 'printer-state' handled on Resume-Printer?	46
111	3.1.3.1.9	Purge-Printer	46
112	3.1.3.2	Job Operations	47
113	3.1.3.2.1	Send-Document	47
114	3.1.3.2.2	Send-URI	47
115	3.1.3.2.3	Cancel-Job	48

116	3.1.3.2.4	Get-Job-Attributes	48
117	3.1.3.2.5	Hold-Job	49
118	3.1.3.2.6	Release-Job	49
119	3.1.3.2.7	Restart-Job	50
120	3.1.3.2.7.1	Can documents be added to a restarted job?	50
121	3.1.4	Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)	50
122	3.1.5	Sending empty attribute groups	50
123	3.2	PRINTER OPERATIONS	51
124	3.2.1	Print-Job operation	51
125	3.2.1.1	Flow controlling the data portion of a Print-Job request (Issue 1.22)	51
126	3.2.1.2	Returning job-state in Print-Job response (Issue 1.30)	51
127	3.2.2	Get-Printer-Attributes operation	52
128	3.2.3	Get-Jobs operation	52
129	3.2.3.1	Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?	52
130	3.2.3.2	Why is there a "limit" attribute in the Get-Jobs operation?	52
131	3.2.4	Create-Job operation	53
132	3.3	JOB OPERATIONS	53
133	3.3.1	Validate-Job	53
134	3.3.2	Restart-Job	53
135	4	OBJECT ATTRIBUTES	54
136	4.1	ATTRIBUTE SYNTAX'S	54
137	4.1.1	The 'none' value for empty sets (Issue 1.37)	54
138	4.1.2	Multi-valued attributes (Issue 1.31)	54
139	4.1.3	Case Sensitivity in URIs (issue 1.6)	54
140	4.1.4	Maximum length for xxxWithLanguage and xxxWithoutLanguage	55
141	4.2	JOB TEMPLATE ATTRIBUTES	55
142	4.2.1	multiple-document-handling (type2 keyword)	55
143	4.2.1.1	Support of multiple document jobs	55
144	4.3	JOB DESCRIPTION ATTRIBUTES	56
145	4.4	PRINTER DESCRIPTION ATTRIBUTES	56
146	4.4.1	queued-job-count	56
147	4.4.1.1	Why is "queued-job-count" RECOMMENDED (Issue 1.14)?	56
148	4.4.1.2	Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?	56
149	4.4.2	printer-current-time (dateTime)	57
150	4.4.3	'Printer-uri'	57
151	4.5	EMPTY JOBS	57
152	5	<u>DNS DIRECTORY</u> CONSIDERATIONS	58
153	5.1	<u>DIRECTORY SCHEMA CONSIDERATIONS</u>	<u>57</u>
154	5.2	IPP PRINTER WITH A DNS NAME	58
155	6	SECURITY CONSIDERATIONS	58
156	6.1	QUERYING JOBS WITH IPP THAT WERE SUBMITTED USING OTHER JOB SUBMISSION PROTOCOLS (ISSUE 1.32)	58
157	7	ENCODING AND TRANSPORT	59
158	7.1	GENERAL HEADERS	60
159	7.2	REQUEST HEADERS	61
160	7.3	RESPONSE HEADERS	62
161	7.4	ENTITY HEADERS	62
162	7.5	OPTIONAL SUPPORT FOR HTTP/1.0	63
163	7.6	HTTP/1.1 CHUNKING	64
164	7.6.1	Disabling IPP Server Response Chunking	64
165	7.6.2	Warning About the Support of Chunked Requests	64
166	8	REFERENCES	64

167 8.1 AUTHORS' ADDRESS.....66

168 **9 NOTICES.....66**

169 **10 CHANGE HISTORY (TO BE REMOVED AT TIME OF RFC PUBLISHING).....67**

170 10.1 CHANGES FROM 990914 TO 99092767

171 10.2 CHANGES FROM 990726 TO 990914:68

172 10.3 CHANGES TO PRODUCE THE FEBRUARY 12, 1999 VERSION FROM THE JANUARY 8, 1999 VERSION:69

173 10.4 CHANGES TO PRODUCE THE JANUARY 8, 1999 VERSION FROM THE DECEMBER 6, 1998 VERSION:.....69

174 10.5 CHANGES TO PRODUCE THE DECEMBER 6, 1998 VERSION FROM THE NOVEMBER 16, 1998 VERSION:69

175

176 **TABLES**

177

178 Table 1 - Summary of Printer operation attributes that sender **MUST** supply..... 8

179 Table 2 - Summary of Printer operation attributes that sender **MAY** supply 9

180 Table 3 - Summary of Job operation attributes that sender **MUST** supply..... 10

181 Table 4 - Summary of Job operation attributes that sender **MAY** supply..... 11

182 Table 5 - Printer operation response attributes 11

183 Table 6 - Examples of validating IPP version..... 14

184 Table 7 - Rules for validating single values X against Z 30

185

186
187
188

189 **1 Introduction**

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

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

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

202 1.1 Conformance language

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

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

214 1.2 Other terminology

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

217 1.3 Issues Raised from Interoperability Bake Offs

218 The IPP WG has conducted two open interoperability "Bake Offs". The first bake off was held in
219 September 1998 and Bake Off2 was held in March 1999. See the summary reports in:

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

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

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

223 These issue resolutions have been incorporated into the November 16, "IPP/1.0 Model and Semantics" [ipp-
224 mod] and the "IPP/1.0 Encoding and Transport" [IPP-PRO] documents. However, some of the discussion
225 is left here in the Implementer's Guide to help understanding.

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

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

228 **2 IPP Objects**

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

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

236 3 IPP Operations

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

239 3.1 Common Semantics

240 This section discusses semantics common to all operations.

241 3.1.1 Summary of Operation Attributes

242 Legend for the following table:

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

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

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

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

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

249

250

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

Operation Attributes	Printer Operations						Responses
	Requests						
	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
<u>document-access-error</u>							<u>O**</u>
compression	O	O					
document-format	R	R		<u>R</u>			
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.

251

252

253

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	

254

255

256

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.

257

Table 5 - Printer operation response attributes

Operation Attributes	Printer Operations						
	Response						
	Print-Job (R), Send-Document (O), Send-URI (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				

258

259

260 3.1.2 Suggested Operation Processing Steps for IPP Objects

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

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

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

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

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

287 3.1.2.1 Suggested Operation Processing Steps for all Operations

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

291	IIG Sect #	Flow	IPP error status codes
292	-----	----	-----
293			
294		v	err
295	3.1.2.1.1	<Validate version>	--> server-error-version-not-supported
296		ok	
297		v	err
298	3.1.2.1.2	<Validate operation>	--> server-error-operation-not-supported
299		ok	
300		v	err
301	3.1.2.1.4.1-	<Validate presence>	--> client-error-bad-request
302	3.1.2.1.4.2	<of attributes>	
303		ok	
304		v	err
305	3.1.2.1.4.3	<Validate presence>	--> client-error-bad-request
306		<of operation attr>	
307		ok	
308		v	err
309	3.1.2.1.5	<Valied values of>	--> client-error-bad-request
310		<operation attrs>	client-error-request-value-too-long
311		<(length, tag, range,>	
312		<multi-value)>	
313		ok	
314		v	err
315	3.1.2.1.5	<Validate values>	--> client-error-bad-request
316		<with supported values>	client-error-charset-not-supported
317		ok	client-error-attributes-or-values-
318			not-supported
319		v	err
320	3.1.2.1.6	<Validate optionally>	--> client-error-bad-request
321		<operation attr>	client-error-natural-language-not-
322		ok	supported
323			client-error-request-value-too-long
324			client-error-attributes-or-values-
325		v	not-supported
326			

327 3.1.2.1.1 Validate version number

328 Every request and every response contains the "version-number" attribute. The value of this attribute is the
 329 major and minor version number of the syntax and semantics that the client and IPP object is using,
 330 respectively. The "version-number" attribute remains in a fixed position across all future versions so that

331 all clients and IPP object that support future versions can determine which version is being used. The IPP
 332 object checks to see if the major version number supplied in the request is supported. If not, the Printer
 333 object REJECTS the request and RETURNS the 'server-error-version-not-supported' status code in the
 334 response. The IPP object returns in the "version-number" response attribute the major and minor version
 335 for the error response. Thus the client can learn at least one major and minor version that the IPP object
 336 supports. The IPP object is encouraged to return the closest version number to the one supplied by the
 337 client.

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

345 **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

346

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

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

352 3.1.2.1.2 Validate operation identifier

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

356 3.1.2.1.3 Validate the request identifier

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

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

362 3.1.2.1.4 Validate attribute group and attribute presence and order

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

364 3.1.2.1.4.1 Validate the presence and order of attribute groups

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

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

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

377 3.1.2.1.4.2 Ignore unknown attribute groups in the expected position

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

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

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

390 3.1.2.1.4.3 Validate the presence of a single occurrence of required Operation attributes
391 Client requests and IPP object responses contain Operation attributes that [IPP-MOD] Section 3 requires to
392 be present. Attributes within a group may be in any order, except for the ordering of target, charset, and
393 natural languages attributes. These attributes MUST be first, and MUST be supplied in the following order:
394 charset, natural language, and then target. An IPP object verifies that the attributes that Section 4 requires to
395 be supplied by the client have been supplied in the request (attributes without an * in the following tables).
396 An asterisk (*) indicates groups and Operation attributes that the client may omit in a request or an IPP
397 object may omit in a response.

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

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

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

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

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

- 418 R indicates a REQUIRED attribute or operation that an IPP object MUST support
- 419 O indicates an OPTIONAL attribute or operation that an IPP object NEED NOT support
- 420 * indicates that a client MAY omit the attribute in a request and that an IPP object MAY omit the
421 attribute in a response. The absence of an * means that a client MUST supply the
422 attribute in a request and an IPP object MUST supply the attribute in a response.
- 423 + indicates that this is not a IPP/1.0 operation, but is only a part of IPP/1.1 and future versions of IPP.

424

425

Operation Requests

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

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

429

430 Print-Job Request (R):

431 Group 1: Operation Attributes (R)
432 attributes-charset (R)
433 attributes-natural-language (R)
434 printer-uri (R)
435 requesting-user-name (R*)
436 job-name (R*)
437 ipp-attribute-fidelity (R*)
438 document-name (R*)
439 document-format (R*)
440 document-natural-language (O*)
441 compression (O*)
442 job-k-octets (O*)
443 job-impressions (O*)
444 job-media-sheets (O*)
445 Group 2: Job Template Attributes (R*)
446 <Job Template attributes> (O*)
447 (see [IPP-MOD] Section 4.2)
448 Group 3: Document Content (R)
449 <document content>

450

451 Validate-Job Request (R):

452 Group 1: Operation Attributes (R)
453 attributes-charset (R)
454 attributes-natural-language (R)
455 printer-uri (R)
456 requesting-user-name (R*)
457 job-name (R*)
458 ipp-attribute-fidelity (R*)
459 document-name (R*)
460 document-format (R*)
461 document-natural-language (O*)
462 compression (O*)
463 job-k-octets (O*)
464 job-impressions (O*)
465 job-media-sheets (O*)
466 Group 2: Job Template Attributes (R*)
467 <Job Template attributes> (O*)
468 (see [IPP-MOD] Section 4.2)

469

470 Print-URI Request (O):

471 Group 1: Operation Attributes (R)
472 attributes-charset (R)

473 attributes-natural-language (R)
474 printer-uri (R)
475 document-uri (R)
476 requesting-user-name (R*)
477 job-name (R*)
478 ipp-attribute-fidelity (R*)
479 document-name (R*)
480 document-format (R*)
481 document-natural-language (O*)
482 compression (O*)
483 job-k-octets (O*)
484 job-impressions (O*)
485 job-media-sheets (O*)
486 Group 2: Job Template Attributes (R*)
487 <Job Template attributes> (O*) (see
488 (see [IPP-MOD] Section 4.2)
489
490 Create-Job Request (O):
491 Group 1: Operation Attributes (R)
492 attributes-charset (R)
493 attributes-natural-language (R)
494 printer-uri (R)
495 requesting-user-name (R*)
496 job-name (R*)
497 ipp-attribute-fidelity (R*)
498 job-k-octets (O*)
499 job-impressions (O*)
500 job-media-sheets (O*)
501 Group 2: Job Template Attributes (R*)
502 <Job Template attributes> (O*) (see
503 (see [IPP-MOD] Section 4.2)
504
505 Get-Printer-Attributes Request (R):
506 Group 1: Operation Attributes (R)
507 attributes-charset (R)
508 attributes-natural-language (R)
509 printer-uri (R)
510 requesting-user-name (R*)
511 requested-attributes (R*)
512 document-format (R*)
513
514 Get-Jobs 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 limit (R*)
521 requested-attributes (R*)

522 which-jobs (R*)
523 my-jobs (R*)
524
525 Send-Document Request (O):
526 Group 1: Operation Attributes (R)
527 attributes-charset (R)
528 attributes-natural-language (R)
529 (printer-uri & job-id) | job-uri (R)
530 last-document (R)
531 requesting-user-name (R*)
532 document-name (R*)
533 document-format (R*)
534 document-natural-language (O*)
535 compression (O*)
536 Group 2: Document Content (R*)
537 <document content>
538
539 Send-URI Request (O):
540 Group 1: Operation Attributes (R)
541 attributes-charset (R)
542 attributes-natural-language (R)
543 (printer-uri & job-id) | job-uri (R)
544 last-document (R)
545 document-uri (R)
546 requesting-user-name (R*)
547 document-name (R*)
548 document-format (R*)
549 document-natural-language (O*)
550 compression (O*)
551
552 Cancel-Job Request (R):
553 Release-Job Request (O+):
554 Group 1: Operation Attributes (R)
555 attributes-charset (R)
556 attributes-natural-language (R)
557 (printer-uri & job-id) | job-uri (R)
558 requesting-user-name (R*)
559 message (O*)
560
561 Get-Job-Attributes Request (R):
562 Group 1: Operation Attributes (R)
563 attributes-charset (R)
564 attributes-natural-language (R)
565 (printer-uri & job-id) | job-uri (R)
566 requesting-user-name (R*)
567 requested-attributes (R*)
568
569 Pause-Printer Request (O+):
570 Resume-Printer Request (O+):

617 attributes-charset (R)
618 attributes-natural-language (R)
619 status-message (O*)
620 detailed-status-message (O*)
621 Group 2: Unsupported Attributes (R*) (see Note 3)
622 <unsupported attributes> (R*)
623
624 Print-URI Response (O):
625 Send-URI Response (O):
626 Group 1: Operation Attributes (R)
627 attributes-charset (R)
628 attributes-natural-language (R)
629 status-message (O*)
630 detailed-status-message (O*)
631 document-access-error (O*)
632 Group 2: Unsupported Attributes (R*) (see Note 3)
633 <unsupported attributes> (R*)
634 Group 3: Job Object Attributes(R*) (see Note 2)
635 job-uri (R)
636 job-id (R)
637 job-state (R)
638 job-state-reasons (O* | R+)
639 job-state-message (O*)
640 number-of-intervening-jobs (O*)
641
642 Get-Printer-Attributes Response (R):
643 Group 1: Operation Attributes (R)
644 attributes-charset (R)
645 attributes-natural-language (R)
646 status-message (O*)
647 detailed-status-message (O*)
648 Group 2: Unsupported Attributes (R*) (see Note 4)
649 <unsupported attributes> (R*)
650 Group 3: Printer Object Attributes(R*) (see Note 2)
651 <requested attributes> (R*)
652
653 Get-Jobs Response (R):
654 Group 1: Operation Attributes (R)
655 attributes-charset (R)
656 attributes-natural-language (R)
657 status-message (O*)
658 detailed-status-message (O*)
659 Group 2: Unsupported Attributes (R*) (see Note 4)
660 <unsupported attributes> (R*)
661 Group 3: Job Object Attributes(R*) (see Note 2, 5)
662 <requested attributes> (R*)
663
664 Get-Job-Attributes Response (R):
665 Group 1: Operation Attributes (R)

666 attributes-charset (R)
667 attributes-natural-language (R)
668 status-message (O*)
669 detailed-status-message (O*)
670 Group 2: Unsupported Attributes (R*) (see Note 4)
671 <unsupported attributes> (R*)
672 Group 3: Job Object Attributes (R*) (see Note 2)
673 <requested attributes> (R*)
674
675 Pause-Printer Response (O+):
676 Resume-Printer Response (O+):
677 Purge-Printer Response (O+):
678 Group 1: Operation Attributes (R)
679 attributes-charset (R)
680 attributes-natural-language (R)
681 status-message (O*)
682 detailed-status-message (O*)
683 Group 2: Unsupported Attributes (R*) (see Note 4)
684 <unsupported attributes> (R*)
685

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

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

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

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

694 3.1.2.1.5 Validate the values of the REQUIRED Operation attributes

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

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

- 699 a) that the length of each Operation attribute value is correct for the attribute syntax tag supplied by
700 the client according to [IPP-MOD] Section 4.1,
- 701 b) that the attribute syntax tag is correct for that Operation attribute according to [IPP-MOD]
702 Section 3,

703 c) that the value is in the range specified for that Operation attribute according to [IPP-MOD]
704 Section 3,

705 d) that multiple values are supplied by the client only for operation attributes that are multi-valued,
706 i.e., that are 1setOf X according to [IPP-MOD] Section 3.

707

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

714 In addition, the IPP object checks each Operation attribute value against some Printer object attribute or
715 some hard-coded value if there is no "xxx-supported" Printer object attribute defined. If its value is not
716 among those supported or is not in the range supported, then the IPP object REJECTS the request and
717 RETURNS the error status code indicated in the table by the second IF statement. If the value of the Printer
718 object's "xxx-supported" attribute is 'no-value' (because the system administrator hasn't configured a value),
719 the check always fails.

720 -----

721 attributes-charset (charset)

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

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

724 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-charset-
725 not-supported".

726

727 attributes-natural-language(naturalLanguage)

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

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

730 ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-language-
731 supported" attribute. If the supplied value is not a member of the Printer object's "generated-natural-
732 language-supported" attribute, use the Printer object's "natural-language-configured" value.

733

734 requesting-user-name

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

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

737 IF the IPP object can obtain a better-authenticated name, use it instead.

738

739 job-name(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 NOT supplied by the client, the Printer object creates a name from the document-name or document-
743 uri.
744

745 document-name (name)

746 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
747 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
748

749 ipp-attribute-fidelity (boolean)

750 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-
751 request'.
752 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'.
753 IF NOT supplied by the client, the IPP object assumes the value 'false'.
754

755 document-format (mimeMediaType)

756 IF NOT a single non-empty 'mimeMediaType' value, REJECT/RETURN 'client-error-bad-request'.
757 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
758 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN 'client-error-
759 document-format-not-supported'.
760 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's "document-
761 format-default" attribute.
762

763 document-uri (uri)

764 IF NOT a single non-empty 'uri' value, REJECT/RETURN 'client-error-bad-request'.
765 IF the value length is greater than 1023 octets, REJECT/RETURN 'client-error-request-value-too-long'.
766 IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.
767 IF the client-supplied URI scheme is not supported, i.e. the value is not in the Printer object's referenced-
768 uri-scheme-supported" attribute, the Printer object MUST reject the request and return the 'client-
769 error-uri-scheme-not-supported' status code. The Printer object MAY check to see if the document
770 exists and is accessible. If the document is not found or is not accessible, REJECT/RETURN
771 'client-error-not found'.

772 last-document (boolean)

773 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-
774 request'.
775 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'.
776

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

778 IF NOT an single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN
779 'client-error-bad-request'.

780 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-error-
781 gone' status code, if keep track of recently deleted jobs.
782

783 requested-attributes (1setOf keyword)

784 IF NOT one or more 'keyword' values, REJECT/RETURN 'client-error-bad-request'.
785 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
786 Ignore unsupported values, which are the keyword names of unsupported attributes. Don't bother to
787 copy such requested (unsupported) attributes to the Unsupported Attribute response group since the
788 response will not return them.
789

790 which-jobs (type2 keyword)

791 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
792 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
793 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to the
794 Unsupported Attributes response group and REJECT/RETURN 'client-error-attributes-or-values-
795 not-supported'.
796 Note: a Printer still supports the 'completed' value even if it keeps no completed/canceled/aborted jobs:
797 by returning no jobs when so queried.
798 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.
799

800 my-jobs (boolean)

801 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-
802 request'.
803 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'
804 IF NOT supplied by the client, the IPP object assumes the 'false' value.
805

806 limit (integer(1:MAX))

807 IF NOT a single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN
808 'client-error-bad-request'.
809 IF NOT supplied by the client, the IPP object returns all jobs, no matter how many.
810
811 -----
812

813 3.1.2.1.6 Validate the values of the OPTIONAL Operation attributes

814 OPTIONAL Operation attributes are those that an IPP object MAY or MAY NOT support. An IPP object
815 validates the values of the OPTIONAL attributes supplied by the client. The IPP object performs the same
816 syntactic validation checks for each OPTIONAL attribute value as in Section 3.1.2.1.5. As in Section
817 3.1.2.1.5, if any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' or
818 the 'client-error-request-value-too-long' status code.

819 In addition, the IPP object checks each Operation attribute value against some Printer attribute or some
820 hard-coded value if there is no "xxx-supported" Printer attribute defined. If its value is not among those
821 supported or is not in the range supported, then the IPP object REJECTS the request and RETURNS the
822 error status code indicated in the table. If the value of the Printer object's "xxx-supported" attribute is 'no-
823 value' (because the system administrator hasn't configured a value), the check always fails.

824 If the IPP object doesn't recognize/support an attribute, the IPP object treats the attribute as an unknown or
825 unsupported attribute (see the last row in the table below).

826 -----

827 document-natural-language (naturalLanguage)

828 IF NOT a single non-empty 'naturalLanguage' value, REJECT/RETURN 'client-error-bad-request'.
829 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-long'.
830 IF NOT a value that the Printer object supports in document formats, (no corresponding "xxx-
831 supported" Printer attribute), REJECT/RETURN 'client-error-natural-language-not-supported'.
832

833 compression (type3 keyword)

834 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
835 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
836 IF NOT in the Printer object's "compression-supported" attribute, copy the attribute and the unsupported
837 value to the Unsupported Attributes response group and REJECT/RETURN 'client-error-attributes-
838 or-values-not-supported'.

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

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

844 IF NOT a single 'integer' value equal to 4 octets,
845 REJECT/RETURN 'client-error-bad-request'.
846 IF NOT in the range of the Printer object's "job-k-octets-supported" attribute, copy the attribute and the
847 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-
848 error-attributes-or-values-not-supported'.
849

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

851 IF NOT a single 'integer' value equal to 4 octets,
852 REJECT/RETURN 'client-error-bad-request'.
853 IF NOT in the range of the Printer object's "job-impressions-supported" attribute, copy the attribute and
854 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-
855 error-attributes-or-values-not-supported'.
856

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

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

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

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

863

864 message (text(127))

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

866 IF the value length is greater than 127 octets,

867 REJECT/RETURN 'client-error-request-value-too-long'.

868

869 unknown or unsupported attribute

870 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute
871 syntax, REJECT/RETURN 'client-error-request-value-too-long'.

872 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
873 attribute value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.

874

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

890 3.1.2.2 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add
891 Documents

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

897 898 IIG Sect #	Flow	IPP error status codes
899 -----	----	-----
	v	No
902 3.1.2.2.1 <ipp-attribute-fidelity>	-----+	
903 <supplied?>		
904 Yes		
905 ipp-attribute-fidelity = no		
906 <-----+		
	v	No
908 3.1.2.2.2 <Printer is>	-->	server-error-not-accepting-jobs
909 <accepting jobs?>		
910 Yes		
	v	err
912 3.1.2.3 <Validate values of>	-->	client-error-bad-request
913 <Job template attributes>		client-error-request-value-too-long
914 <(length, tag, range,>		
915 <multi-value)>		
916 ok		
	v	err
918 3.1.2.3 <Validate values with>	-->	client-error-bad-request
919 <supported values>		client-error-attributes-or-values-
920 ok		not-supported
	v	err
922 3.1.2.3.1 <Any conflicting>	-->	client-error-conflicting-attributes
923 <Job Template attr values>		client-error-attributes-or-values-
924 ok		not-supported
925 v		

926 3.1.2.2.1 Default "ipp-attribute-fidelity" if not supplied

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

929 3.1.2.2.2 Check that the Printer object is accepting jobs

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

932 3.1.2.2.3 Validate the values of the Job Template attributes

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

936 a) that the length of each value is correct for the attribute syntax tag supplied by the client
937 according to [IPP-MOD] Section 4.1.

938 b) that the attribute syntax tag is correct for that attribute according to [IPP-MOD] Sections 4.2 to
939 4.4.

940 c) that multiple values are supplied only for multi-valued attributes, i.e., that are 1setOf X
941 according to [IPP-MOD] Sections 4.2 to 4.4.

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

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

951 1. reject the operation and return the 'client-error-bad-request' error status code

952 2. accept the operation and use the first occurrence of the attribute

953 3. accept the operation and use the last occurrence of the attribute

954 depending on implementation. Therefore, clients MUST NOT supply multiple occurrences of the same Job
955 Template attribute in the Job Attributes group in the request.

956 3.1.2.3 Algorithm for job validation

957 The process of validating a Job-Template attribute "xxx" against a Printer attribute "xxx-supported" can use
958 the following validation algorithm (see section 3.2.1.2 in [ipp-mod]).

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

961 1. If U is multi-valued, validate each value X of U by performing the algorithm in Table 7 with each value
962 X. Each validation is separate from the standpoint of returning unsupported values. Example: If U is

963 "finishings" that the client supplies with 'staple', 'bind' values, then X takes on the successive values:
 964 'staple', then 'bind'

965 2. If V is multi-valued, validate X against each Z of V by performing the algorithm in Table 7 with each
 966 value Z. If a value Z validates, the validation for the attribute value X succeeds. If it fails, the algorithm
 967 is applied to the next value Z of V. If there are no more values Z of V, validation fails. Example" If V is
 968 "sides-supported" with values: 'one-sided', 'two-sided-long', and 'two-sided-short', then Z takes on the
 969 successive values: 'one-sided', 'two-sided-long', and 'two-sided-short'. If the client supplies "sides" with
 970 'two-sided-long', the first comparison fails ('one-sided' is not equal to 'two-sided-long'), the second
 971 comparison succeeds ('two-sided-long' is equal to 'two-sided-long'), and the third comparison ('two-
 972 sided-short' with 'two-sided-long') is not even performed.

973 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.

974 **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.

975

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

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

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

993 -----

- 994 job-priority (integer(1:100))
- 995 IF NOT a single 'integer' value with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
- 996 request'.
- 997 IF NOT supplied by the client, use the value of the Printer object's "job-priority-default" attribute at job
- 998 submission time.
- 999 IF NOT in the range 1 to 100, inclusive, copy the attribute and the unsupported value to the
- 1000 Unsupported Attributes response group.
- 1001 Map the value to the nearest supported value in the range 1:100 as specified by the number of discrete
- 1002 values indicated by the value of the Printer's "job-priority-supported" attribute. See the formula in
- 1003 [IPP-MOD] Section 4.2.1.
- 1004
- 1005 job-hold-until (type3 keyword | name)
- 1006 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
- 1007 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
- 1008 IF NOT supplied by the client, use the value of the Printer object's "job-hold-until" attribute at job
- 1009 submission time.
- 1010 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the
- 1011 unsupported value to the Unsupported Attributes response group.
- 1012
- 1013 job-sheets (type3 keyword | name)
- 1014 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
- 1015 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
- 1016 IF NOT in the Printer object's "job-sheets-supported" attribute, copy the attribute and the unsupported
- 1017 value to the Unsupported Attributes response group.
- 1018
- 1019 multiple-document-handling (type2 keyword)
- 1020 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
- 1021 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
- 1022 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute and
- 1023 the unsupported value to the Unsupported Attributes response group.
- 1024
- 1025 copies (integer(1:MAX))
- 1026 IF NOT a single 'integer' value with a length equal to 4 octets,
- 1027 REJECT/RETURN 'client-error-bad-request'.
- 1028 IF NOT in range of the Printer object's "copies-supported" attribute
- 1029 copy the attribute and the unsupported value to the Unsupported Attributes response group.
- 1030
- 1031 finishings (1setOf type2 enum)
- 1032 IF NOT an 'enum' value(s) each with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
- 1033 request'.

- 1034 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the unsupported
1035 value(s), but not any supported values, to the Unsupported Attributes response group.
1036
- 1037 page-ranges (1setOf rangeOfInteger(1:MAX))
- 1038 IF NOT a 'rangeOfInteger' value(s) each with a length equal to 8 octets, REJECT/RETURN 'client-
1039 error-bad-request'.
1040 IF first value is greater than second value in any range, the ranges are not in ascending order, or ranges
1041 overlap, REJECT/RETURN 'client-error-bad-request'.
1042 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to the
1043 Unsupported Attributes response group and set the value to the "out-of-band" 'unsupported' value.
1044
- 1045 sides (type2 keyword)
- 1046 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
1047 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
1048 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported value
1049 to the Unsupported Attributes response group.
1050
- 1051 number-up (integer(1:MAX))
- 1052 IF NOT a single 'integer' value with a length equal to 4 octets,
1053 REJECT/RETURN 'client-error-bad-request'.
1054 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"
1055 attribute, copy the attribute and value to the Unsupported Attribute response group.
1056
- 1057 orientation-requested (type2 enum)
- 1058 IF NOT a single 'enum' value with a length equal to 4 octets,
1059 REJECT/RETURN 'client-error-bad-request'.
1060 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the
1061 unsupported value to the Unsupported Attributes response group.
1062
- 1063 media (type3 keyword | name)
- 1064 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
1065 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
1066 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported value
1067 to the Unsupported Attributes response group.
1068
- 1069 printer-resolution (resolution)
- 1070 IF NOT a single 'resolution' value with a length equal to 9 octets,
1071 REJECT/RETURN 'client-error-bad-request'.
1072 IF NOT in the Printer object's "printer-resolution-supported" attribute, copy the attribute and the
1073 unsupported value to the Unsupported Attributes response group.

1074

1075 print-quality (type2 enum)

1076 IF NOT a single 'enum' value with a length equal to 4 octets,

1077 REJECT/RETURN 'client-error-bad-request'.

1078 IF NOT in the Printer object's "print-quality-supported" attribute, copy the attribute and the unsupported
1079 value to the Unsupported Attributes response group.

1080

1081 unknown or unsupported attribute (i.e., there is no corresponding Printer object "xxx-supported" attribute)

1082 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute

1083 syntax,

1084 REJECT/RETURN 'client-error-bad-request' if the length of the attribute syntax is fixed or 'client-error-
1085 request-value-too-long' if the length of the attribute syntax is variable.1086 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
1087 attribute value to the "out-of-band" 'unsupported' value. Any remaining Job Template Attributes are
1088 either unknown or unsupported Job Template attributes and are validated algorithmically according
1089 to their attribute syntax for proper length (see below).

1090

1091 -----
 1091 If the attribute syntax is supported AND the length check fails, the IPP object REJECTS the request and
 1092 RETURNS the 'client-error-bad-request' if the length of the attribute syntax is fixed or the 'client-error-
 1093 request-value-too-long' status code if the length of the attribute syntax is variable. Otherwise, the IPP object
 1094 copies the unsupported Job Template attribute to the Unsupported Attributes response group and changes
 1095 the attribute value to the "out-of-band" 'unsupported' value. The following table shows the length checks
 1096 for all attribute syntaxes. In the following table: "<=" means less than or equal, "=" means equal to:

1097 Name	Octet length check for read-write attributes
1098 -----	-----
1099 'textWithLanguage	<= 1023 AND 'naturalLanguage' <= 63
1100 'textWithoutLanguage'	<= 1023
1101 'nameWithLanguage'	<= 255 AND 'naturalLanguage' <= 63
1102 'nameWithoutLanguage'	<= 255
1103 'keyword'	<= 255
1104 'enum'	= 4
1105 'uri'	<= 1023
1106 'uriScheme'	<= 63
1107 'charset'	<= 63
1108 'naturalLanguage'	<= 63
1109 'mimeType'	<= 255
1110 'octetString'	<= 1023
1111 'boolean'	= 1
1112 'integer'	= 4
1113 'rangeOfInteger'	= 8
1114 'dateTime'	= 11
1115 'resolution'	= 9
1116 'lsetOf X'	

1117

1118 Note: It's possible for a Printer to receive a zero length keyword in a request. Since this is a keyword, its
1119 value needs to be compared with the supported values. Assuming that the printer doesn't have any values in
1120 its corresponding "xxx-supported" attribute that are keywords of zero length, the comparison will fail.
1121 Then the request will be accepted or rejected depending on the value of "ipp-attributes-fidelity" being 'false'
1122 or 'true', respectively. No special handling is required for

1123 3.1.2.3.1 Check for conflicting Job Template attributes values

1124 Once all the Operation and Job Template attributes have been checked individually, the Printer object
1125 SHOULD check for any conflicting values among all the supported values supplied by the client. For
1126 example, a Printer object might be able to staple and to print on transparencies, however due to physical
1127 stapling constraints, the Printer object might not be able to staple transparencies. The IPP object copies the
1128 supported attributes and their conflicting attribute values to the Unsupported Attributes response group.
1129 The Printer object only copies over those attributes that the Printer object either ignores or substitutes in
1130 order to resolve the conflict, and it returns the original values which were supplied by the client. For
1131 example suppose the client supplies "finishings" equals 'staple' and "media" equals 'transparency', but the
1132 Printer object does not support stapling transparencies. If the Printer chooses to ignore the stapling request
1133 in order to resolve the conflict, the Printer objects returns "finishings" equal to 'staple' in the Unsupported
1134 Attributes response group. If any attributes are multi-valued, only the conflicting values of the attributes
1135 are copied.

1136 Note: The decisions made to resolve the conflict (if there is a choice) is implementation dependent.

1137 3.1.2.3.2 Decide whether to REJECT the request

1138 If there were any unsupported Job Template attributes or unsupported/conflicting Job Template attribute
1139 values and the client supplied the "ipp-attribute-fidelity" attribute with the 'true' value, the Printer object
1140 REJECTS the request and return the status code:

- 1141 (1) 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes supplied
1142 by the client.
- 1143 (2) 'client-error-attributes-or-values-not-supported' status code, otherwise.

1144
1145 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this
1146 step. If the unsupported Operation attribute was a serious error, the above already rejected the request in a
1147 previous step. If control gets to this step with unsupported Operation attributes being returned, they are not
1148 serious errors.

1149 In general, the final results of Job processing are unknown at Job submission time. The client has to rely on
1150 notifications or polling to find out what happens at Job processing time. However, there are cases in which
1151 some Printers can determine at Job submission time that Job processing is going to fail. As an
1152 optimization, we'd like to have the Printer reject the Job in these cases.

1153 There are three types of "processing" errors that might be detectable at Job submission time:

1154 1. 'client-error-document-format-not-supported': For the Print-Job, Send-Document, Print-URI, and Send-
1155 URI operations, if all these conditions are true:

- 1156 · the Printer supports auto-sensing,
- 1157 · the request "document-format" operation attribute is 'application/octet-stream',
- 1158 · the Printer receives document data before responding,
- 1159 · the Printer auto-senses the document format before responding,
- 1160 · the sensed document format is not supported by the Printer

1161 then the Printer should respond with 'client-error-document-format-not-supported' status.

1162 2. 'client-error-compression-error': For the Print-Job, Send-Document, Print-URI, and Send-URI
1163 operations, if all these conditions are true:

- 1164 · the client supplies a supported value for the "compression" operation attribute in the request
- 1165 · the Printer receives document data before responding,
- 1166 · the Printer attempts to decompress the document data before responding,
- 1167 · the document data cannot be decompressed using the algorithm specified by the

1168 "compression" operation attribute

1169 then the Printer should respond with 'client-error-compression-error' status.

1170 3. 'client-error-document-access-error': For the Print-URI, and Send-URI operations, if the Printer attempts
1171 and fails to pull the referenced document data before responding, it should respond with 'client-error-
1172 document-access-error' status.

1173 Some Printers are not able to detect these errors until Job processing time. In that case, the errors are
1174 recorded in the corresponding job-state and job-state reason attributes. (There is no standard way for a
1175 client to determine whether a Printer can detect these errors at Job submission time.) For example, if auto-
1176 sensing happens AFTER the job is accepted (as opposed to auto-sensing at submit time before returning the
1177 response), the implementation aborts the job, puts the job in the 'aborted' state and sets the 'unsupported-
1178 document-format' value in the job's "job-state-reasons".

1179 A client should always provide a valid "document-format" operation attribute whenever practical. In the
1180 absence of other information, a client itself may sniff the document data to determine document format.

1181 Auto sensing at Job submission time may be more difficult for the Printer when combined with
1182 compression. For auto-sensed Jobs, a client may be better off deferring compression to the transfer
1183 protocol layer, e.g.; by using the HTTP Content-Encoding header.

1184 3.1.2.3.3 For the Validate-Job operation, RETURN one of the success status codes

1185 If the requested operation is the Validate-Job operation, the Printer object returns:

- 1186 (1) the "successful-ok" status code, if there are no unsupported or conflicting Job Template attributes or
1187 values.
1188 (2) the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or
1189 values.
1190 (3) the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template
1191 attributes or values.
1192

1193 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this
1194 step. If the unsupported Operation attribute was a serious error, the above already rejected the request in a
1195 previous step. If control gets to this step with unsupported Operation attributes being returned, they are not
1196 serious errors.

1197 3.1.2.3.4 Create the Job object with attributes to support

1198 If "ipp-attribute-fidelity" is set to 'false' (or it was not supplied by the client), the Printer object:

- 1199 (1) creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and
1200 initializes all of the job's other supported Job Description attributes.
1201 (2) removes all unsupported attributes from the Job object.
1202 (3) for each unsupported value, removes either the unsupported value or substitutes the unsupported
1203 attribute value with some supported value. If an attribute has no values after removing unsupported
1204 values from it, the attribute is removed from the Job object (so that the normal default behavior at
1205 job processing time will take place for that attribute).
1206 (4) for each conflicting value, removes either the conflicting value or substitutes the conflicting attribute
1207 value with some other supported value. If an attribute has no values after removing conflicting
1208 values from it, the attribute is removed from the Job object (so that the normal default behavior at
1209 job processing time will take place for that attribute).
1210

1211 If there were no attributes or values flagged as unsupported, or the value of "ipp-attribute-fidelity" was
1212 'false', the Printer object is able to accept the create request and create a new Job object. If the "ipp-
1213 attribute-fidelity" attribute is set to 'true', the Job Template attributes that populate the new Job object are
1214 necessarily all the Job Template attributes supplied in the create request. If the "ipp-attribute-fidelity"
1215 attribute is set to 'false', the Job Template attributes that populate the new Job object are all the client
1216 supplied Job Template attributes that are supported or that have value substitution. Thus, some of the
1217 requested Job Template attributes may not appear in the Job object because the Printer object did not
1218 support those attributes. The attributes that populate the Job object are persistently stored with the Job
1219 object for that Job. A Get-Job-Attributes operation on that Job object will return only those attributes that
1220 are persistently stored with the Job object.

1221 Note: All Job Template attributes that are persistently stored with the Job object are intended to be
1222 "override values"; that is, they that take precedence over whatever other embedded instructions might be in
1223 the document data itself. However, it is not possible for all Printer objects to realize the semantics of
1224 "override". End users may query the Printer's "pdl-override-supported" attribute to determine if the Printer
1225 either attempts or does not attempt to override document data instructions with IPP attributes.

1226 There are some cases, where a Printer supports a Job Template attribute and has an associated default value
1227 set for that attribute. In the case where a client does not supply the corresponding attribute, the Printer does
1228 not use its default values to populate Job attributes when creating the new Job object; only Job Template
1229 attributes actually in the create request are used to populate the Job object. The Printer's default values are
1230 only used later at Job processing time if no other IPP attribute or instruction embedded in the document
1231 data is present.

1232 Note: If the default values associated with Job Template attributes that the client did not supply were to be
1233 used to populate the Job object, then these values would become "override values" rather than defaults. If
1234 the Printer supports the 'attempted' value of the "pdl-override-supported" attribute, then these override
1235 values could replace values specified within the document data. This is not the intent of the default value
1236 mechanism. A default value for an attribute is used only if the create request did not specify that attribute
1237 (or it was ignored when allowed by "ipp-attribute-fidelity" being 'false') and no value was provided within
1238 the content of the document data.

1239 If the client does not supply a value for some Job Template attribute, and the Printer does not support that
1240 attribute, as far as IPP is concerned, the result of processing that Job (with respect to the missing attribute)
1241 is undefined.

1242 3.1.2.3.5 Return one of the success status codes

1243 Once the Job object has been created, the Printer object accepts the request and returns to the client:

- 1244 (1) the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes or
1245 values.
- 1246 (2) the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template
1247 attribute or values.
- 1248 (3) the 'successful-ok-ignored-or-substituted-attributes' status code, if there are only unsupported Job
1249 Template attributes or values.

1250
1251 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this
1252 step. If the unsupported Operation attribute was a serious error, the above already rejected the request in a
1253 previous step. If control gets to this step with unsupported Operation attributes being returned, they are not
1254 serious errors.

1255 The Printer object also returns Job status attributes that indicate the initial state of the Job ('pending',
1256 'pending-held', 'processing', etc.), etc. See Print-Job Response, [IPP-MOD] section 3.2.1.2.

1257 3.1.2.3.6 Accept appended Document Content

1258 The Printer object accepts the appended Document Content data and either starts it printing, or spools it for
1259 later processing.

1260 3.1.2.3.7 Scheduling and Starting to Process the Job

1261 The Printer object uses its own configuration and implementation specific algorithms for scheduling the Job
1262 in the correct processing order. Once the Printer object begins processing the Job, the Printer changes the
1263 Job's state to 'processing'. If the Printer object supports PDL override (the "pdl-override-supported" attribute
1264 set to 'attempted'), the implementation does its best to see that IPP attributes take precedence over
1265 embedded instructions in the document data.

1266 3.1.2.3.8 Completing the Job

1267 The Printer object continues to process the Job until it can move the Job into the 'completed' state. If an
1268 Cancel-Job operation is received, the implementation eventually moves the Job into the 'canceled' state. If
1269 the system encounters errors during processing that do not allow it to progress the Job into a completed
1270 state, the implementation halts all processing, cleans up any resources, and moves the Job into the 'aborted'
1271 state.

1272 3.1.2.3.9 Destroying the Job after completion

1273 Once the Job moves to the 'completed', 'aborted', or 'canceled' state, it is an implementation decision as to
1274 when to destroy the Job object and release all associated resources. Once the Job has been destroyed, the
1275 Printer would return either the "client-error-not-found" or "client-error-gone" status codes for operations
1276 directed at that Job.

1277 Note: the Printer object SHOULD NOT re-use a "job-uri" or "job-id" value for a sufficiently long time
1278 after a job has been destroyed, so that stale references kept by clients are less likely to access the wrong
1279 (newer) job.

1280 3.1.2.3.10 Interaction with "ipp-attribute-fidelity"

1281 Some Printer object implementations may support "ipp-attribute-fidelity" set to 'true' and "pdl-override-
1282 supported" set to 'attempted' and yet still not be able to realize exactly what the client specifies in the create
1283 request. This is due to legacy decisions and assumptions that have been made about the role of job
1284 instructions embedded within the document data and external job instructions that accompany the document
1285 data and how to handle conflicts between such instructions. The inability to be 100% precise about how a
1286 given implementation will behave is also compounded by the fact that the two special attributes, "ipp-
1287 attribute-fidelity" and "pdl-override-supported", apply to the whole job rather than specific values for each
1288 attribute. For example, some implementations may be able to override almost all Job Template attributes
1289 except for "number-up". Character Sets, natural languages, and internationalization

1290 This section discusses character set support, natural language support and internationalization.

1291 3.1.2.3.11 Character set code conversion support

1292 IPP clients and IPP objects are REQUIRED to support UTF-8. They MAY support additional charsets. It
1293 is RECOMMENDED that an IPP object also support US-ASCII, since many clients support US-ASCII, and

1294 indicate that UTF-8 and US-ASCII are supported by populating the Printer's "charset-supported" with 'utf-8'
1295 and 'us-ascii' values. An IPP object is required to code covert with as little loss as possible between the
1296 charsets that it supports, as indicated in the Printer's "charsets-supported" attribute.

1297 How should the server handle the situation where the "attributes-charset" of the response itself is "us-ascii",
1298 but one or more attributes in that response is in the "utf-8" format?

1299 Example: Consider a case where a client sends a Print-Job request with "utf-8" as the value of "attributes-
1300 charset" and with the "job-name" attribute supplied. Later another client submits a Get-Job-Attribute or
1301 Get-Jobs request. This second request contains the "attributes-charset" with value "us-ascii" and
1302 "requested-attributes" attribute with exactly one value "job-name".

1303 According to the IPP-Mod document (section 3.1.4.2), the value of the "attributes-charset" for the response
1304 of the second request must be "us-ascii" since that is the charset specified in the request. The "job-name"
1305 value, however, is in "utf-8" format. Should the request be rejected even though both "utf-8" and "us-ascii"
1306 charsets are supported by the server? or should the "job-name" value be converted to "us-ascii" and return
1307 "successful-ok-conflicting-attributes" (0x0002) as the status code?

1308 Answer: An IPP object that supports both utf-8 (REQUIRED) and us-ascii, the second paragraph of section
1309 3.1.4.2 applies so that the IPP object MUST accept the request, perform code set conversion between these
1310 two charsets with "the highest fidelity possible" and return 'successful-ok', rather than a warning
1311 'successful-ok-conflicting-attributes, or an error. The printer will do the best it can to convert between each
1312 of the character sets that it supports--even if that means providing a string of question marks because none
1313 of the characters are representable in US ASCII. If it can't perform such conversion, it MUST NOT
1314 advertise us-ascii as a value of its "attributes-charset-supported" and MUST reject any request that requests
1315 'us-ascii'.

1316 One IPP object implementation strategy is to convert all request text and name values to a Unicode internal
1317 representation. This is 16-bit and virtually universal. Then convert to the specified operation attributes-
1318 charset on output.

1319 Also it would be smarter for a client to ask for 'utf-8', rather than 'us-ascii' and throw away characters that it
1320 doesn't understand, rather than depending on the code conversion of the IPP object.

1321 3.1.2.3.12 What charset to return when an unsupported charset is requested (Issue 1.19)?

1322 Section 3.1.4.1 Request Operation attributes was clarified in November 1998 as follows:

1323 All clients and IPP objects MUST support the 'utf-8' charset [RFC2044] and MAY support additional
1324 charsets provided that they are registered with IANA [IANA-CS]. If the Printer object does not support the
1325 client supplied charset value, the Printer object MUST reject the request, set the "attributes-charset" to 'utf-
1326 8' in the response, and return the 'client-error-charset-not-supported' status code and any 'text' or 'name'
1327 attributes using the 'utf-8' charset.

1328 Since the client and IPP object MUST support UTF-8, returning any text or name attributes in UTF-8 when
1329 the client requests a charset that is not supported should allow the client to display the text or name.

1330 Since such an error is a client error, rather than a user error, the client should check the status code first so
1331 that it can avoid displaying any other returned 'text' and 'name' attributes that are not in the charset
1332 requested.

1333 Furthermore, [ipp-mod] section 14.1.4.14 client-error-charset-not-supported (0x040D) was clarified in
1334 November 1998 as follows:

1335 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-
1336 charset" operation attribute, the Printer MUST reject the operation and return this status and any 'text' or
1337 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1).

1338 3.1.2.3.13 Natural Language Override (NLO)

1339 The 'text' and 'name' attributes each have two forms. One has an implicit natural language, and the other
1340 has an explicit natural language. The 'textWithoutLanguage' and 'textWithLanguage' are the two 'text'
1341 forms. The 'nameWithoutLanguage' and 'nameWithLanguage' are the two 'name' forms. If a receiver (IPP
1342 object or IPP client) supports an attribute with attribute syntax 'text', it MUST support both forms in a
1343 request and a response. A sender (IPP client or IPP object) MAY send either form for any such attribute.
1344 When a sender sends a WithoutLanguage form, the implicit natural language is specified in the "attributes-
1345 natural-language" operation attribute, which all senders MUST include in every request and response.

1346 When a sender sends a WithLanguage form, it MAY be different from the implicit natural language
1347 supplied by the sender or it MAY be the same. The receiver MUST treat either form equivalently.

1348 There is an implementation decision for senders, whether to always send the WithLanguage forms or use
1349 the WithoutLanguage form when the attribute's natural language is the same as the request or response. The
1350 former approach makes the sender implementation simpler. The latter approach is more efficient on the
1351 wire and allows inter-working with non-conforming receivers that fail to support the WithLanguage forms.
1352 As each approach have advantages, the choice is completely up to the implementer of the sender.

1353 Furthermore, when a client receives a 'text' or 'name' job attribute that it had previously supplied, that client
1354 MUST NOT expect to see the attribute in the same form, i.e., in the same WithoutLanguage or
1355 WithLanguage form as the client supplied when it created the job. The IPP object is free to transform the
1356 attribute from the WithLanguage form to the WithoutLanguage form and vice versa, as long as the natural
1357 language is preserved. However, in order to meet this latter requirement, it is usually simpler for the IPP
1358 object implementation to store the natural language explicitly with the attribute value, i.e., to store using an
1359 internal representation that resembles the WithLanguage form.

1360 The IPP Printer MUST copy the natural language of a job, i.e., the value of the "attributes-natural-language"
1361 operation attribute supplied by the client in the create operation, to the Job object as a Job Description
1362 attribute, so that a client is able to query it. In returning a Get-Job-Attributes response, the IPP object MAY
1363 return one of three natural language values in the response's "attributes-natural-language" operation

1364 attribute: (1) that requested by the requester, (2) the natural language of the job, or (3) the configured
1365 natural language of the IPP Printer, if the requested language is not supported by the IPP Printer.

1366 This "attributes-natural-language" Job Description attribute is useful for an IPP object implementation that
1367 prints start sheets in the language of the user who submitted the job. This same Job Description attribute is
1368 useful to a multi-lingual operator who has to communicate with different job submitters in different natural
1369 languages. This same Job Description attribute is expected to be used in the future to generate notification
1370 messages in the natural language of the job submitter.

1371 Early drafts of [IPP-MOD] contained a job-level natural language override (NLO) for the Get-Jobs
1372 response. A job-level (NLO) is an (unrequested) Job Attribute which then specified the implicit natural
1373 language for any other WithoutLanguage job attributes returned in the response for that job.
1374 Interoperability testing of early implementations showed that no one was implementing the job-level NLO
1375 in Get-Job responses. So the job-level NLO was eliminated from the Get-Jobs response. This
1376 simplification makes all requests and responses consistent in that the implicit natural language for any
1377 WithoutLanguage 'text' or 'name' form is always supplied in the request's or response's "attributes-natural-
1378 language" operation attribute.

1379 3.1.3 Status codes returned by operation

1380 This section corresponds to [IPP-MOD] section 3.1.6 "Operation Response Status Codes and Status
1381 Messages". This section lists all status codes once in the first operation (Print-Job). Then it lists the status
1382 codes that are different or specialized for subsequent operations under each operation.

1383 3.1.3.1 Printer Operations

1384 3.1.3.1.1 Print-Job

1385 The Printer object MUST return one of the following "status-code" values for the indicated reason.
1386 Whether all of the document data has been accepted or not before returning the success or error response
1387 depends on implementation. See Section 13 in [IPP-MOD] for a more complete description of each status
1388 code.

1389 For the following success status codes, the Job object has been created and the "job-id", and "job-uri"
1390 assigned and returned in the response:

1391 successful-ok: no request attributes were substituted or ignored.

1392 successful-ok-ignored-or-substituted-attributes: some supplied (1) attributes were ignored or (2)
1393 unsupported attribute syntaxes or values were substituted with supported values or were ignored.
1394 Unsupported attributes, attribute syntax's, or values MUST be returned in the Unsupported
1395 Attributes group of the response.

1396 successful-ok-conflicting-attributes: some supplied attribute values conflicted with the values of other
1397 supplied attributes and were either substituted or ignored. Attributes or values which conflict with
1398 other attributes and have been substituted or ignored MUST be returned in the Unsupported
1399 Attributes group of the response as supplied by the client.

1400

1401 [ipp-mod] section 3.1.6 Operation Status Codes and Messages states:

1402 If the Printer object supports the "status-message" operation attribute, it SHOULD use the REQUIRED 'utf-
1403 8' charset to return a status message for the following error status codes (see section 13 in [IPP-MOD]):
1404 'client-error-bad-request', 'client-error-charset-not-supported', 'server-error-internal-error', 'server-error-
1405 operation-not-supported', and 'server-error-version-not-supported'. In this case, it MUST set the value of
1406 the "attributes-charset" operation attribute to 'utf-8' in the error response.

1407 For the following error status codes, no job is created and no "job-id" or "job-uri" is returned:

1408 client-error-bad-request: The request syntax does not conform to the specification.

1409 client-error-forbidden: The request is being refused for authorization or authentication reasons. The
1410 implementation security policy is to not reveal whether the failure is one of authentication or
1411 authorization.

1412 client-error-not-authenticated: Either the request requires authentication information to be supplied or
1413 the authentication information is not sufficient for authorization.

1414 client-error-not-authorized: The requester is not authorized to perform the request on the target object.

1415 client-error-not-possible: The request cannot be carried out because of the state of the system. See also
1416 'server-error-not-accepting-jobs' status code, which MUST take precedence if the Printer object's
1417 "printer-accepting-jobs" attribute is 'false'.

1418 client-error-timeout: not applicable.

1419 client-error-not-found: the target object does not exist.

1420 client-error-gone: the target object no longer exists and no forwarding address is known.

1421 client-error-request-entity-too-large: the size of the request and/or print data exceeds the capacity of the
1422 IPP Printer to process it.

1423 client-error-request-value-too-long: the size of request variable length attribute values, such as 'text' and
1424 'name' attribute syntax's, exceed the maximum length specified in [IPP-MOD] for the attribute and
1425 MUST be returned in the Unsupported Attributes Group.

1426 client-error-document-format-not-supported: the document format supplied is not supported. The
1427 "document-format" attribute with the unsupported value MUST be returned in the Unsupported
1428 Attributes Group. This error SHOULD take precedence over any other 'xxx-not-supported' error,
1429 except 'client-error-charset-not-supported'.

1430 client-error-attributes-or-values-not-supported: one or more supplied attributes, attribute syntax's, or
1431 values are not supported and the client supplied the "ipp-attributes-fidelity" operation attribute with
1432 a 'true' value. They MUST be returned in the Unsupported Attributes Group as explained below.

1433 client-error-uri-scheme-not-supported: not applicable.

1434 client-error-charset-not-supported: the charset supplied in the "attributes-charset" operation attribute is
1435 not supported. The Printer's "configured-charset" MUST be returned in the response as the value of
1436 the "attributes-charset" operation attribute and used for any 'text' and 'name' attributes returned in the
1437 error response. This error SHOULD take precedence over any other error, unless the request syntax
1438 is so bad that the client's supplied "attributes-charset" cannot be determined.

1439 client-error-conflicting-attributes: one or more supplied attribute values conflicted with each other and
1440 the client supplied the "ipp-attributes-fidelity" operation attribute with a 'true' value. They MUST
1441 be returned in the Unsupported Attributes Group as explained below.

1442 server-error-internal-error: an unexpected condition prevents the request from being fulfilled.
1443 server-error-operation-not-supported: not applicable (since Print-Job is REQUIRED).
1444 server-error-service-unavailable: the service is temporarily overloaded.
1445 server-error-version-not-supported: the version in the request is not supported. The "closest" version
1446 number supported MUST be returned in the response.
1447 server-error-device-error: a device error occurred while receiving or spooling the request or document
1448 data or the IPP Printer object can only accept one job at a time.
1449 server-error-temporary-error: a temporary error such as a buffer full write error, a memory overflow, or
1450 a disk full condition occurred while receiving the request and/or the document data.
1451 server-error-not-accepting-jobs: the Printer object's "printer-is-not-accepting-jobs" attribute is 'false'.
1452 server-error-busy: the Printer is too busy processing jobs to accept another job at this time.
1453 server-error-job-canceled: the job has been canceled by an operator or the system while the client was
1454 transmitting the document data.

1455 3.1.3.1.2 Print-URI

1456 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Print-
1457 URI with the following specializations and differences. See Section 14 for a more complete description of
1458 each status code.

1459 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation
1460 attribute is not supported and is returned in the Unsupported Attributes group.
1461 server-error-operation-not-supported: the Print-URI operation is not supported.
1462

1463 3.1.3.1.3 Validate-Job

1464 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to
1465 Validate-Job. See Section 13 in [IPP-MOD] for a more complete description of each status code.

1466 3.1.3.1.4 Create-Job

1467 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Create-
1468 Job with the following specializations and differences. See Section 13 in [IPP-MOD] for a more complete
1469 description of each status code.

1470 server-error-operation-not-supported: the Create-Job operation is not supported.
1471 client-error-multiple-document-jobs-not-supported: while the Create-Job and Send-Document
1472 operations are supported, this implementation doesn't support more than one document with data.

1473 3.1.3.1.5 Get-Printer-Attributes

1474 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the
1475 Get-Printer-Attributes operation with the following specialization's and differences. See Section 13 in
1476 [IPP-MOD] for a more complete description of each status code.

1477 For the following success status codes, the requested attributes are returned in Group 3 in the response:
1478 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1479 attributes were unsupported.
1480 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1481 operation attribute MAY, but NEED NOT, be returned with the unsupported values.
1482 successful-ok-conflicting-attributes: same as Print-Job.

1483 For the error status codes, Group 3 is returned containing no attributes or is not returned at all:
1484 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.
1485 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.
1486 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1487 MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.
1488 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.
1489 server-error-operation-not-supported: not applicable (since Get-Printer-Attributes is REQUIRED).
1490 server-error-device-error: same as Print-Job, except that no document data is involved.
1491 server-error-temporary-error: same as Print-Job, except that no document data is involved.
1492 server-error-not-accepting-jobs: not applicable..
1493 server-error-busy: same as Print-Job, except the IPP object is too busy to accept even query requests.
1494 server-error-job-canceled: not applicable..

1495 3.1.3.1.6 Get-Jobs

1496 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the
1497 Get-Jobs operation with the following specialization's and differences. See Section 13 in [IPP-MOD] for a
1498 more complete description of each status code.

1499 For the following success status codes, the requested attributes are returned in Group 3 in the response:
1500 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1501 attributes were unsupported.
1502 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1503 operation attribute MAY, but NEED NOT, be returned with the unsupported values.
1504 successful-ok-conflicting-attributes: same as Print-Job.

1505 For any error status codes, Group 3 is returned containing no attributes or is not returned at all. The
1506 following brief error status code descriptions contain unique information for use with Get-Jobs operation.
1507 See section 14 for the other error status codes that apply uniformly to all operations:

1508 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.
1509 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.
1510 client-error-document-format-not-supported: not applicable.
1511 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1512 MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.
1513 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.
1514 server-error-operation-not-supported: not applicable (since Get-Jobs is REQUIRED).
1515 server-error-device-error: same as Print-Job, except that no document data is involved.

- 1516 server-error-temporary-error: same as Print-Job, except that no document data is involved.
1517 server-error-not-accepting-jobs: not applicable.
1518 server-error-job-canceled: not applicable.
- 1519 3.1.3.1.7 Pause-Printer
- 1520 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Pause-
1521 Printer with the following specializations and differences. See Section 13 in [IPP-MOD] for a more
1522 complete description of each status code.
- 1523 For the following success status codes, the Printer object is being stopped from scheduling jobs on all its
1524 devices.
- 1525 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
1526 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
1527 successful-ok-conflicting-attributes: same as Print-Job.
1528
- 1529 For any of the error status codes, the Printer object has not been stopped from scheduling jobs on all its
1530 devices.
- 1531 client-error-not-possible: not applicable.
1532 client-error-not-found: the target Printer object does not exist.
1533 client-error-gone: the target Printer object no longer exists and no forwarding address is known.
1534 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
1535 client-error-document-format-not-supported: not applicable.
1536 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-
1537 jobs" attribute is not involved.
1538 server-error-operation-not-supported: the Pause-Printer operation is not supported.
1539 server-error-device-error: not applicable.
1540 server-error-temporary-error: same as Print-Job, except no document data is involved.
1541 server-error-not-accepting-jobs: not applicable.
1542 server-error-job-canceled: not applicable.
- 1543 3.1.3.1.8 Resume-Printer
- 1544 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
1545 specialization's described for Pause-Printer are applicable to Resume-Printer. See Section 13 in [IPP-
1546 MOD] for a more complete description of each status code.
- 1547 For the following success status codes, the Printer object resumes scheduling jobs on all its devices.
- 1548 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
1549 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
1550 successful-ok-conflicting-attributes: same as Print-Job.
- 1551 For any of the error status codes, the Printer object does not resume scheduling jobs.

1552 server-error-operation-not-supported: the Resume-Printer operation is not supported.

1553

1554 3.1.3.1.8.1 What about Printers unable to change state due to an error condition?

1555 If, in case, the IPP printer is unable to change its state due to some problem with the actual printer device
1556 (say, it is shut down or there is a media-jam as indicated in [ipp-mod]), what should be the result of the
1557 "Resume-printer" operation? Should it still change the 'printer-state-reasons' and return success or should it
1558 fail ?

1559 The 'resume-printer' operation must clear the 'paused' or 'moving-to-paused' 'printer-state-message'. The
1560 operation must return a 'successful-ok' status code.

1561 3.1.3.1.8.2 How is 'printer-state' handled on Resume-Printer?

1562

1563 If "Resume-Printer" succeeds, what should be the value of 'Printer-state' and who should take care of the
1564 'Printer-state' later on ?

1565 The "Resume-Printer" operation may change the "printer-state-reasons" value.

1566 The "printer-state" will change to one of three states:

- 1567 1) 'idle' - no additional jobs and no error conditions present
1568 2) 'processing' - job available and no error conditions present
1569 3) current state (i.e. no change) an error condition is present (e.g. media jam)

1570 In the third case the 'printer-state-reason' will be cleared by automata when it detects the error condition no
1571 longer exists. The 'printer-state' will move to 'idle' or 'processing' when conditions permit. (i.e. no more
1572 error conditions)

1573 3.1.3.1.9 Purge-Printer

1574 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
1575 specialization's described for Pause-Printer are applicable to Purge-Printer. See Section 13 in [IPP-MOD]
1576 for a more complete description of each status code.

1577 For the following success status codes, the Printer object purges all it's jobs.

- 1578 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
1579 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
1580 successful-ok-conflicting-attributes: same as Print-Job.

1581 For any of the error status codes, the Printer object does not purge any jobs.

1582 server-error-operation-not-supported: the Purge-Printer operation is not supported.

1583 3.1.3.2 Job Operations

1584 3.1.3.2.1 Send-Document

1585 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the
1586 Get-Printer-Attributes operation with the following specialization's and differences. See Section 13 in
1587 [IPP-MOD] for a more complete description of each status code.

1588 For the following success status codes, the document has been added to the specified Job object and the
1589 job's "number-of-documents" attribute has been incremented:

1590 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

1591 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

1592 successful-ok-conflicting-attributes: same as Print-Job.

1593 For the error status codes, no document has been added to the Job object and the job's "number-of-
1594 documents" attribute has not been incremented:

1595 client-error-not-possible: Same as Print-Job, except that the Printer's "printer-is-accepting-jobs"
1596 attribute is not involved, so that the client is able to finish submitting a [multi-document job that was
1597 created with a Create-Job operation](#) after this attribute has been set to 'true'. Another condition is
1598 that the state of the job precludes Send-Document, i.e., the job has already been closed out by the
1599 client. However, if the IPP Printer closed out the job due to timeout, the 'client-error-timeout' error
1600 status SHOULD be returned instead.

1601 client-error-timeout: This request was sent after the Printer closed the job, because it has not received a
1602 Send-Document or Send-URI operation within the Printer's "multiple-operation-time-out" period .

1603 client-error-request-entity-too-large: same as Print-Job.

1604 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attributes-fidelity" operation
1605 attribute is not involved..

1606 server-error-operation-not-supported: the Send-Document request is not supported.

1607 server-error-not-accepting-jobs: not applicable.

1608 server-error-job-canceled: the job has been canceled by an operator or the system while the client was
1609 transmitting the data.

1610 3.1.3.2.2 Send-URI

1611 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
1612 specialization's described for Send-Document are applicable to Send-URI. See Section 13 in [IPP-MOD]
1613 for a more complete description of each status code.

1614 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation
1615 attribute is not supported and the "document-uri" attribute MUST be returned in the Unsupported
1616 Attributes group.

1617 server-error-operation-not-supported: the Send-URI operation is not supported.
1618

1619 3.1.3.2.3 Cancel-Job

1620 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Cancel-
1621 Job with the following specializations and differences. See Section 13 in [IPP-MOD] for a more complete
1622 description of each status code.

1623 For the following success status codes, the Job object is being canceled or has been canceled:

1624 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

1625 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

1626 successful-ok-conflicting-attributes: same as Print-Job.

1627

1628 For any of the error status codes, the Job object has not been canceled or was previously canceled.

1629 client-error-not-possible: The request cannot be carried out because of the state of the Job object
1630 ('completed', 'canceled', or 'aborted') or the state of the system.

1631 client-error-not-found: the target Printer and/or Job object does not exist.

1632 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding address is
1633 known.

1634 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.

1635 client-error-document-format-not-supported: not applicable.

1636 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1637 and values MUST be ignored.

1638 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-
1639 jobs" attribute is not involved.

1640 server-error-operation-not-supported: not applicable (Cancel-Job is REQUIRED).

1641 server-error-device-error: same as Print-Job, except no document data is involved.

1642 server-error-temporary-error: same as Print-Job, except no document data is involved.

1643 server-error-not-accepting-jobs: not applicable..

1644 server-error-job-canceled: not applicable.

1645 3.1.3.2.4 Get-Job-Attributes

1646 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Get-
1647 Job-Attributes with the following specializations and differences. See Section 13 in [IPP-MOD] for a more
1648 complete description of each status code.

1649 For the following success status codes, the requested attributes are returned in Group 3 in the response:

1650 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1651 attributes were unsupported.

1652 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1653 operation attribute MAY, but NEED NOT, be returned with the unsupported values.

1654 successful-ok-conflicting-attributes: same as Print-Job.

1655 For the error status codes, Group 3 is returned containing no attributes or is not returned at all.

1656 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.
1657 client-error-document-format-not-supported: not applicable.
1658 client-error-attributes-or-values-not-supported: not applicable.
1659 client-error-uri-scheme-not-supported: not applicable.
1660 client-error-conflicting-attributes: not applicable
1661 server-error-operation-not-supported: not applicable (since Get-Job-Attributes is REQUIRED).
1662 server-error-device-error: same as Print-Job, except no document data is involved.
1663 server-error-temporary-error: sane as Print-Job, except no document data is involved..
1664 server-error-not-accepting-jobs: not applicable.
1665 server-error-job-canceled: not applicable.

1666 3.1.3.2.5 Hold-Job

1667 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Hold-
1668 Job with the following specializations and differences. See Section 13 in [IPP-MOD] for a more complete
1669 description of each status code.

1670 For the following success status codes, the Job object is being held or has been held:

1671 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
1672 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
1673 successful-ok-conflicting-attributes: same as Print-Job.
1674

1675 For any of the error status codes, the Job object has not been held or was previously held.

1676 client-error-not-possible: The request cannot be carried out because of the state of the Job object
1677 ('completed', 'canceled', or 'aborted') or the state of the system.
1678 client-error-not-found: the target Printer and/or Job object does not exist.
1679 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding address is
1680 known.
1681 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
1682 client-error-document-format-not-supported: not applicable.
1683 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-
1684 jobs" attribute is not involved.
1685 server-error-operation-not-supported: the Hold-Job operation is not supported.
1686 server-error-device-error: not applicable.
1687 server-error-temporary-error: same as Print-Job, except no document data is involved.
1688 server-error-not-accepting-jobs: not applicable.
1689 server-error-job-canceled: not applicable.

1690 3.1.3.2.6 Release-Job

1691 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
1692 specialization's described for Hold-Job are applicable to Release-Job. See Section 13 in [IPP-MOD] for a
1693 more complete description of each status code.

1694 server-error-operation-not-supported: the Release-Job operation is not supported.

1695 3.1.3.2.7 Restart-Job

1696 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
1697 specialization's described for Hold-Job are applicable to Restart-Job. See Section 13 in [IPP-MOD] for a
1698 more complete description of each status code.

1699 server-error-operation-not-supported: the Restart-Job operation is not supported.
1700

1701 3.1.3.2.7.1 Can documents be added to a restarted job?

1702 Assume I give a Create-Job request along with a set of 5 documents . All the documents get printed and the
1703 job state is moved to completed . I issue a Restart-Job request on the job. Now the issue is that, if I try to
1704 add new documents to the restarted job-, will the [IppIPP](#) Server permit me to do so or return "client-error-
1705 not-possible " and again print those 5 jobs?

1706 A job can not move to the 'completed' state until all the documents have been processed. The 'last-
1707 document' flag indicates when the last document for a job is being sent from the client. This is the semantic
1708 equivalent of closing a job. No documents may be added once a job is closed. Section 3.3.7 of the IPP/1.1
1709 model states "The job is moved to the 'pending' job state and restarts the beginning on the same IPP Printer
1710 object with the same attribute values." 'number-of-documents' is a job attribute.

1713 3.1.4 Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)

1714 In the Get-Printer-Attributes, Get-Jobs, or Get-Job-Attributes responses, the client cannot depend on getting
1715 unsupported attributes returned in the Unsupported Attributes group that the client requested, but are not
1716 supported by the IPP object. However, such unsupported requested attributes will not be returned in the
1717 Job Attributes or Printer Attributes group (since they are unsupported). Furthermore, the IPP object is
1718 REQUIRED to return the 'successful-ok-ignored-or-substituted-attributes' status code, so that the client
1719 knows that not all that was requested has been returned.

1720 3.1.5 Sending empty attribute groups

1721 The [IPP-MOD] and [IPP-PRO] specifications RECOMMEND that a sender not send an empty attribute
1722 group in a request or a response. However, they REQUIRE a receiver to accept an empty attribute group as
1723 equivalent to the omission of that group. So a client SHOULD omit the Job Template Attributes group
1724 entirely in a create operation that is not supplying any Job Template attributes. Similarly, an IPP object
1725 SHOULD omit an empty Unsupported Attributes group if there are no unsupported attributes to be returned
1726 in a response.

1727 The [IPP-PRO] specification REQUIRES a receiver to be able to receive either an empty attribute group or
1728 an omitted attribute group and treat them equivalently. The term "receiver" means an IPP object for a

1729 request and a client for a response. The term "sender" means a client for a request and an IPP object for a
1730 response.

1731 There is an exception to the rule for Get-Jobs when there are no attributes to be returned. [IPP-PRO]
1732 contains the following paragraph:

1733 The syntax allows an xxx-attributes-tag to be present when the xxx-attribute-sequence that follows is
1734 empty. The syntax is defined this way to allow for the response of Get-Jobs where no attributes are returned
1735 for some job-objects. Although it is RECOMMENDED that the sender not send an xxx-attributes-tag if
1736 there are no attributes (except in the Get-Jobs response just mentioned), the receiver MUST be able to
1737 decode such syntax.

1738 3.2 Printer Operations

1739 3.2.1 Print-Job operation

1740 3.2.1.1 Flow controlling the data portion of a Print-Job request (Issue 1.22)

1741 A paused printer, or one that is stopped due to paper out or jam or spool space full or buffer space full, may
1742 flow control the data of a Print-Job operation (at the TCP/IP layer), so that the client is not able to send all
1743 the document data. Consequently, the Printer will not return a response until the condition is changed.

1744 The Printer should not return a Print-Job response with an error code in any of these conditions, since either
1745 the printer will be resumed and/or the condition will be freed either by human intervention or as jobs print.

1746 In writing test scripts to test IPP Printers, the script must also be written not to expect a response, if the
1747 printer has been paused, until the printer is resumed, in order to work with all possible implementations.

1748 3.2.1.2 Returning job-state in Print-Job response (Issue 1.30)

1749 An IPP client submits a small job via Print-Job. By the time the IPP printer/print server is putting together
1750 a response to the operation, the job has finished printing and been removed as an object from the print
1751 system. What should the job-state be in the response?

1752 The Model suggests that the Printer return a response before it even accepts the document content. The Job
1753 Object Attributes are returned only if the IPP object returns one of the success status codes. Then the job-
1754 state would always be "pending" or "pending-held".

1755 This issue comes up for the implementation of an IPP Printer object as a server that forwards jobs to
1756 devices that do not provide job status back to the server. If the server is reasonably certain that the job
1757 completed successfully, then it should return the job-state as 'completed'. Also the server can keep the job
1758 in its "job history" long after the job is no longer in the device. Then a user could query the server and see
1759 that the job was in the 'completed' state and completed as specified by the jobs "time-at-completed" time,
1760 which would be the same as the server submitted the job to the device.

1761 An alternative is for the server to respond to the client before or while sending the job to the device, instead
1762 of waiting until the server has finished sending the job to the device. In this case, the server can return the
1763 job's state as 'pending' with the 'job-outgoing' value in the job's "job-state-reasons" attribute.

1764 If the server doesn't know for sure whether the job completed successfully (or at all), it could return the
1765 (out-of-band) 'unknown' value.

1766 On the other hand, if the server is able to query the device and/or setup some sort of event notification that
1767 the device initiates when the job makes state transitions, then the server can return the current job state in
1768 the Print-Job response and in subsequent queries because the server knows what the job state is in the
1769 device (or can query the device).

1770 All of these alternatives depend on implementation of the server and the device.

1771 3.2.2 Get-Printer-Attributes operation

1772 1 If a Printer supports the "printer-make-and-model" attribute and returns the .INF file model name of
1773 the printer in that attribute, the Microsoft client will automatically install the correct driver (if available).

1774 2 Clients which poll periodically for printer status or queued-job-count should use the "requested-
1775 attributes" operation attribute to limit the scope of the query in order to save Printer and network resources.

1776 3.2.3 Get-Jobs operation

1777 3.2.3.1 Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?

1778 In [IPP-MOD] section 3.2.6.1 'Get-Jobs Request', if the attribute 'my-jobs' is present and set to TRUE,
1779 MUST the 'requesting-user-name' attribute be there too, and if it's not present what should the IPP printer
1780 do?

1781 [IPP-MOD] Section 8.3 describes the various cases of "requesting-user-name" being present or not for any
1782 operation. If the client does not supply a value for "requesting-user-name", the printer MUST assume that
1783 the client is supplying some anonymous name, such as "anonymous".

1784 3.2.3.2 Why is there a "limit" attribute in the Get-Jobs operation?

1785 When using the Get-Jobs operation a client implementer might choose to limit the number of jobs that the
1786 client shows on the first screenful. For example, if its UI can only display 50 jobs, it can defend itself
1787 against a printer that would otherwise return 500 jobs, perhaps taking a long time on a slow dial-up line.
1788 The client can then go and ask for a larger number of jobs in the background, while showing the user the
1789 first 50 jobs. Since the job history is returned in reverse order, namely the most recently completed jobs are
1790 returned first, the user is most likely interested in the first jobs that are returned. Limiting the number of
1791 jobs may be especially useful for a client that is requesting 'completed' jobs from a printer that keeps a long

1792 job history. Clients that don't mind sometimes getting very large responses, can omit the "limit" attribute in
1793 their Get-Jobs requests.

1794 3.2.4 Create-Job operation

1795 A Printer may respond to a Create-Job operation with "job-state" 'pending' or 'pending-held' and "job-state-
1796 reason" 'job-data-insufficient' to indicate that operation has been accepted by the Printer, but the Printer is
1797 expecting additional document data before it can move the job into the 'processing' state. Alternatively, it
1798 may respond with "job-state" 'processing' and "job-state-reason" 'job-incoming' to indicate that the Create-
1799 Job operation has been accepted by the Printer, but the Printer is expecting additional Send-Document
1800 and/or Send-URI operations and/or is accessing/accepting document data. The second alternative is for
1801 non-spooling Printers that don't implement the 'pending' state.

1802 Should the server wait for the "last-document" operation attribute set to 'true' before starting to "process"
1803 the job?

1804 It depends on implementation. Some servers spool the entire job, including all document data, before
1805 starting to process, so such an implementation would wait for the "last-document" before starting to process
1806 the job. If the time-out occurs without the "last-document", then the server takes one of the indicated
1807 actions in section 3.3.1 in the [IPP-MOD] document. Other servers will start to process document data as
1808 soon as they have some. These are the so-called "non-spooling" printers. Currently, there isn't a way for a
1809 client to determine whether the Printer will spool all the data or will start to process (and print) as soon as it
1810 has some data.

1811 3.3 Job Operations

1812 3.3.1 Validate-Job

1813 The Validate-Job operation has been designed so that its implementation may be a part of the Print-Job
1814 operation. Therefore, requiring Validate-Job is not a burden on implementers. Also it is useful for client's
1815 to be able to count on its presence in all conformance implementations, so that the client can determine
1816 before sending a long document, whether the job will be accepted by the IPP Printer or not.

1817 3.3.2 Restart-Job

1818 The Restart-Job operation allows the reprocessing of a completed job. Some jobs store the document data
1819 on the printer. Jobs created using the Print-Job operation are an example. It is required that the printer
1820 retains the job data after the job has moved to a 'completed state' in order for the Restart-Job operation to
1821 succeed.

1822 Some jobs contain only a reference to the job data. A job created using the Print-URI is an example of such
1823 a job. When the Restart-Job operation is issued the job is reprocessed. The job data MUST be retrieved
1824 again to print the job.

1825 It is possible that a job fails while attempting to access the print data. When such a job is the target of a
1826 Restart-Job the Printer SHALL attempt to retrieve the job data again.

1827 4 Object Attributes

1828 4.1 Attribute Syntax's

1829 4.1.1 The 'none' value for empty sets (Issue 1.37)

1830 [IPP-MOD] states that the 'none' value should be used as the value of a 1setOf when the set is empty. In
1831 most cases, sets that are potentially empty contain keywords so the keyword 'none' is used, but for the 3
1832 finishings attributes, the values are enums and thus the empty set is represented by the enum 3. Currently
1833 there are no other attributes with 1setOf values, which can be empty and can contain values that are not
1834 keywords. This exception requires special code and is a potential place for bugs. It would have been better
1835 if we had chosen an out-of-band value, either "no-value" or some new value, such as 'none'. Since we
1836 didn't, implementations have to deal with the different representations of 'none', depending on the attribute
1837 syntax.

1838 4.1.2 Multi-valued attributes (Issue 1.31)

1839 What is the attribute syntax for a multi-valued attribute? Since some attributes support values in more than
1840 one data type, such as "media", "job-hold-until", and "job-sheets", IPP semantics associate the attribute
1841 syntax with each value, not with the attribute as a whole. The protocol associates the attribute syntax tag
1842 with each value. Don't be fooled, just because the attribute syntax tag comes before the attribute keyword.
1843 All attribute values after the first have a zero length attribute keyword as the indication of a subsequent
1844 value of the same attribute.

1845 4.1.3 Case Sensitivity in URIs (issue 1.6)

1846 IPP client and server implementations must be aware of the diverse uppercase/lowercase nature of URIs.
1847 RFC 2396 defines URL schemes and Host names as case insensitive but reminds us that the rest of the URL
1848 may well demonstrate case sensitivity. When creating URL's for fields where the choice is completely
1849 arbitrary, it is probably best to select lower case. However, this cannot be guaranteed and implementations
1850 MUST NOT rely on any fields being case-sensitive or case-insensitive in the URL beyond the URL scheme
1851 and host name fields.

1852 The reason that the IPP specification does not make any restrictions on URIs, is so that implementations of
1853 IPP may use off-the-shelf components that conform to the standards that define URIs, such as RFC 2396
1854 and the HTTP/1.1 specifications [RFC2616]. See these specifications for rules of matching, comparison,
1855 and case-sensitivity.

1856 It is also recommended that System Administrators and implementations avoid creating URLs for different
1857 printers that differ only in their case. For example, don't have Printer1 and printer1 as two different IPP
1858 Printers.

1859 Example of equivalent URI's

1860 `http://abc.com:80/~smith/home.html`

1861 `http://ABC.com/%7Esmith/home.html`

1862 `http://ABC.com:/%7esmith/home.html`

1863 Example of equivalent URI's using the IPP scheme

1864 `ipp://abc.com:631/~smith/home.html`

1865 `ipp://ABC.com/%7Esmith/home.html`

1866 `http://ABC.com:631/%7esmith/home.html`

1867 The HTTP/1.1 specification [RFC2616] contains more details on comparing URLs.

1868 4.1.4 Maximum length for xxxWithLanguage and xxxWithoutLanguage

1869 The 'textWithLanguage' and 'nameWithLanguage' are compound syntaxes that have two components. The
1870 first component is the 'language' component that can contain up to 63 octets. The second component is the
1871 'text' or 'name' component. The maximum length of these are 1023 octets and 255 octets respectively.
1872 The definition of attributes with either syntax may further restrict the length. (e.g. printer-name
1873 (name(127)))

1874 The length of the 'language' component has no effect on the allowable length of 'text' in
1875 'textWithLanguage' or the length of 'name' in 'nameWithLanguage'

1876 4.2 Job Template Attributes

1877 4.2.1 multiple-document-handling(type2 keyword)

1878 4.2.1.1 Support of multiple document jobs

1879 **ISSUE:**—IPP/1.0 is silent on which of the four effects an implementation would perform if it supports
1880 Create-Job, but does not support "multiple-document-handling" or multiple documents per job. IPP/1.1
1881 was changed so that a Printer could support Create-Job without having to support multiple document jobs.
1882 The "multiple-document-jobs-supported" (boolean) Printer description attribute was added to IPP/1.1 along
1883 with the 'server-error-multiple-document-jobs-not-supported' status code for a Printer to indicate whether or
1884 not it supports multiple document jobs, when it supports the Create-Job operation. Also IPP/1.1 was
1885 clarified that the Printer MUST support the "multiple-document-handling" (type2 keyword) Job Template
1886 attribute with at least one value if the Printer supports multiple documents per job.

~~A fix to IPP/1.0 would be to require implementing all four values of "multiple document handling" if Create-Job is supported at all. Or at least 'single document new sheet' and 'separate documents uncollated copies'. In any case, an implementation that supports Create-Job SHOULD also support "multiple document handling". Support for all four values is RECOMMENDED, but at least the 'single document new sheet' and 'separate documents uncollated copies' values, along with the "multiple document handling default" indicating the default behavior and "multiple document handling supported" values. If an implementation spools the data, it should also support the 'separate documents collated copies' value as well.~~

4.3 Job Description Attributes

The time-at-creation, time-at-processing, and time-at-completed attributes may be returned in integer time ticks or absolute dateTime syntax. There are various ways for a Printer to get the time of day. Some suggestions:

1. A Printer can get time from an NTP timeserver if there's one reachable on the network. See RFC 1305. Also DHCP option 32 in RFC 2132 returns the IP address of the NTP server.
2. Get the date and time at startup from a human operator
3. Have an operator set the date and time using a web administrative interface
4. Get the date and time from incoming HTTP requests, though the problems of spoofing need to be considered. Perhaps comparing several HTTP requests could reduce the chances of spoofing.
5. Internal date time clock battery driven.
6. Query "<http://tycho.usno.navy.mil/cgi-bin/timer.pl>"

4.4 Printer Description Attributes

4.4.1 queued-job-count

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

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

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

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

1917 The "queued-job-count" is not a good measure of how busy the printer is when there are held jobs. A future
1918 registration could be to add a "held-job-count" (or an "active-job-count") Printer Description attribute if
1919 experience shows that such an attribute (combination) is needed to quickly indicate how busy a printer
1920 really is.

1921 4.4.2 printer-current-time (dateTime)

1922 A Printer implementation MAY support this attribute by obtaining the date and time by any number of
1923 implementation-dependent means at startup or subsequently. Examples include:

1924 (1) an internal date time clock,

1925 (2) from the operator at startup using the console,

1926 (3) from an operator using an administrative web page,

1927 (4) from HTTP headers supplied in client requests,

1928 (5) use HTTP to query "<http://tycho.usno.navy.mil/cgi-bin/timer.pl>"

1929 (6) from the network, using NTP [RFC1305] or DHCP option 32 [RFC2132] that returns the IP address
1930 of the NTP server.

1931 If an implementation supports this attribute by obtaining the current time from the network (at startup or
1932 later), but the time is not available, then the implementation MUST return the value of this attribute using
1933 the out-of-band 'no-value' meaning not configured. See the beginning of section 4.1.

1934 Since the new "date-and-time-at-xxx" Job Description attributes refer to the "printer-current-time", they
1935 will be covered also.

1936 4.4.3 Printer-uri

1937 Must the operational attribute for printer-uri match one of the values in printer-uri-supported?

1938 A forgiving printer implementation would not reject the operation. But the implementation has its rights to
1939 reject a printer or job operation if the operational attribute printer-uri is not a value of the printer-uri-
1940 supported. The printer may not be improperly configured. The request obviously reached the printer. The
1941 printer could treat the printer-uri as the logical equivalent of a value in the printer-uri-supported. It would
1942 be implementation dependent for which value, and associated security policy, would apply. This does also
1943 apply to a job object specified with a printer-uri and job-id, or with a job-uri. See section 4.1.3 for how to
1944 compare URI's.

1945 4.5 Empty Jobs

1946 The IPP object model does not prohibit a job that contains no documents. Such a job may be created in a
1947 number of ways including a 'create-job' followed by an 'add-document' that contains no data and has the
1948 'last-document' flag set.

1949 An empty job is processed just as any other job. The operation that "closes" an empty job is not rejected
1950 because the job is empty. If no other conditions exist, other than the job is empty, the response to the
1951 operation will indicate success. After the job is scheduled and processed, the job state SHALL be
1952 'completed'

1953 There will be some variation in the value(s) of the 'job-state-reasons' attribute. It is required that if no
1954 conditions, other than the job being empty, exist the 'job-state-reasons' SHALL include the 'completed-
1955 successfully'. If other conditions existed, the 'completed-with-warnings' or 'completed-with-errors' values
1956 may be used."

1957 **5 DNS-Directory Considerations**

1958 5.1 General Directory Schema Considerations

1959 The [ipp-mod] document lists RECOMMENDED and OPTIONAL Printer object attributes for directory
1960 schemas. See [ipp-mod] APPENDIX E: Generic Directory Schema.

1961 The SLP printer template is defined in the "Definition of the Printer Abstract Service Type v2.0" document
1962 [svrloc-printer]. The LDAP printer template is defined in the "Internet Printing Protocol (IPP): LDAP
1963 Schema for Printer Services" document [ldap-printer]. Both documents systematically add "printer-" to any
1964 attribute that doesn't already start with "printer-" in order to keep the printer directory attributes distinct
1965 from other directory attributes. Also, instead of using "printer-uri-supported", "uri-authentication-
1966 supported", and "uri-security-supported", they use a "printer-xri-supported" attribute with special syntax to
1967 contain all of the same information in a single attribute.

1968 5.2 IPP Printer with a DNS name

1969 If the IPP printer has a DNS name should there be at least two values for the printer-uri-supported attribute.
1970 One URL with the fully qualified DNS name the other with the IP address in the URL?

1971 The printer may contain one or the other or both. It's up to the administrator to configure this attribute.

1972 **6 Security Considerations**

1973 This section corresponds to the IPP-MOD Section 8 "Security Considerations.

1974 6.1 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)

1975 The following clarification was added to [IPP-MOD] section 8.5:

8.5 Queries on jobs submitted using non-IPP protocols

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

It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such "foreign jobs", 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 [IPP-PRO].

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

2014 the main application free to do other work. Alternatively, single-threaded applications could use non-
2015 blocking I/O.

2016 Some Printer conditions, such as jam or lack of paper, could cause a client to be blocked indefinitely.
2017 Clients may open additional connections to the Printer to Get-Printer-Attributes, determine the state of the
2018 device, alert a user if the printer is stopped, and let a user decide whether to abort the job transmission or
2019 not.

2020 In the following sections, there are tables of all HTTP headers, which describe their use in an IPP client or
2021 server. The following is an explanation of each column in these tables.

- 2022 • the "header" column contains the name of a header
- 2023 • the "request/client" column indicates whether a client sends the header.
- 2024 • the "request/ server" column indicates whether a server supports the header when received.
- 2025 • the "response/ server" column indicates whether a server sends the header.
- 2026 • the "response /client" column indicates whether a client supports the header when received.
- 2027 • the "values and conditions" column specifies the allowed header values and the conditions for the
2028 header to be present in a request/response.

2029 The table for "request headers" does not have columns for responses, and the table for "response headers"
2030 does not have columns for requests.

2031 The following is an explanation of the values in the "request/client" and "response/ server" columns.

- 2032 • **must:** the client or server MUST send the header,
- 2033 • **must-if:** the client or server MUST send the header when the condition described in the "values and
2034 conditions" column is met,
- 2035 • **may:** the client or server MAY send the header
- 2036 • **not:** the client or server SHOULD NOT send the header. It is not relevant to an IPP implementation.

2037 The following is an explanation of the values in the "response/client" and "request/ server" columns.

- 2038 • **must:** the client or server MUST support the header,
- 2039 • **may:** the client or server MAY support the header
- 2040 • **not:** the client or server SHOULD NOT support the header. It is not relevant to an IPP
2041 implementation.

2042 7.1 General Headers

2043 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

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

2044 7.2 Request Headers

2045 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	

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

2046 7.3 Response Headers

2047 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.

2048 7.4 Entity Headers

2049 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
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	

2050 7.5 Optional support for HTTP/1.0

2051 IPP implementations consist of an HTTP layer and an IPP layer. In the following discussion, the term
 2052 "client" refers to the HTTP client layer and the term "server" refers to the HTTP server layer. The Encoding
 2053 and Transport document [IPP-PRO] requires that HTTP 1.1 MUST be supported by all clients and all
 2054 servers. However, a client and/or a server implementation may choose to also support HTTP 1.0.

2055 • This option means that a server may choose to communicate with a (non-conforming) client that only
 2056 supports HTTP 1.0. In such cases the server should not use any HTTP 1.1 specific parameters or features
 2057 and should respond using HTTP version number 1.0.

2058 • This option also means that a client may choose to communicate with a (non-conforming) server that
 2059 only supports HTTP 1.0. In such cases, if the server responds with an HTTP 'unsupported version number'
 2060 to an HTTP 1.1 request, the client should retry using HTTP version number 1.0.

2061 7.6 HTTP/1.1 Chunking

2062 7.6.1 Disabling IPP Server Response Chunking

2063 Clients **MUST** anticipate that the HTTP/1.1 server may chunk responses and **MUST** accept them in
2064 responses. However, a (non-conforming) HTTP client that is unable to accept chunked responses may
2065 attempt to request an HTTP 1.1 server not to use chunking in its response to an operation by using the
2066 following HTTP header:

2067 TE: identity

2068 This mechanism should not be used by a server to disable a client from chunking a request, since chunking
2069 of document data is an important feature for clients to send long documents.

2070 7.6.2 Warning About the Support of Chunked Requests

2071 This section describes some problems with the use of chunked requests and HTTP/1.1 servers.

2072 The HTTP/1.1 standard [RFC2616] requires that conforming servers support chunked requests for any
2073 method. However, in spite of this requirement, some HTTP/1.1 implementations support chunked
2074 responses in the GET method, but do not support chunked POST method requests. Some HTTP/1.1
2075 implementations that support CGI scripts [CGI] and/or servlets [Servlet] require that the client supply a
2076 Content-Length. These implementations might reject a chunked POST method and return a 411 status code
2077 (Length Required), might attempt to buffer the request and run out of room returning a 413 status code
2078 (Request Entity Too Large), or might successfully accept the chunked request.

2079 Because of this lack of conformance of HTTP servers to the HTTP/1.1 standard, the IPP standard [IPP-
2080 PRO] **REQUIRES** that a conforming IPP Printer object implementation support chunked requests and that
2081 conforming clients accept chunked responses. Therefore, IPP object implementers are warned to seek
2082 HTTP server implementations that support chunked POST requests in order to conform to the IPP standard
2083 and/or use implementation techniques that support chunked POST requests.

2084 **8 References**

2085 [CGI]

2086 CGI/1.1 (<http://www.ietf.org/internet-drafts/draft-coar-cgi-v11-00.txt>).

2087 [IPP-MOD]

2088 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
2089 Semantics", draft-ietf-ipp-model-v11-06.txt, March 1, 2000.

2090 [IPP-PRO]

2091 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and
2092 Transport", draft-ietf-ipp-protocol-v11-05.txt, March 1, 2000.

2093 [[ldap-printer](#)]

- 2094 [Fleming, P., Jones, K., Lewis, H., McDonald, I., "Internet Printing Protocol \(IPP\): LDAP Schema](#)
2095 [for Printer Services", <draft-ietf-ipp-ldap-printer-schema-01.txt>, work in progress, April 27, 2000.](#)
- 2096 [RFC793]
2097 J. Postel, "Transmission Control Protocol", RFC 793.
- 2098 [RFC1123]
2099 Braden, S., "Requirements for Internet Hosts - Application and Support", RFC 1123, October, 1989.
- 2100 [RFC2026]
2101 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 2102 [RFC2119]
2103 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March
2104 1997.
- 2105 [RFC2396]
2106 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic
2107 Syntax", RFC 2396, August 1998.
- 2108 [RFC2565]
2109 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
2110 Semantics", RFC 2566, April 1999.
- 2111 [RFC2566]
2112 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and
2113 Transport", RFC 2565, April 1999.
- 2114 [RFC2567]
2115 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-03.txt, November,
2116 1998.
- 2117 [RFC2568]
2118 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
2119 RFC 2568, April 1999.
- 2120 [RFC2569]
2121 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC
2122 2569, April 1999.
- 2123 [RFC2616]
2124 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext
2125 Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
- 2126 [Servlet]
2127 Servlet Specification Version 2.1 (<http://java.sun.com/products/servlet/2.1/index.html>).

2128 [\[svrloc-printer\]](#)
2129 [St. Pierre, P., Isaacson, S., McDonald, I., "Definition of the Printer Abstract Service Type v2.0",](#)
2130 [<draft-ietf-svrloc-printer-scheme-06.txt>, work in progress, March 8, 2000.](#)

2131 [SSL]
2132 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.

2133 **9 Authors' Address**

2134 Thomas N. Hastings
2135 Xerox Corporation
2136 701 Aviation Blvd.
2137 El Segundo, CA 90245
2138 hastings@cp10.es.xerox.com

2139
2140 Carl-Uno Manros
2141 Xerox Corporation
2142 701 Aviation Blvd.
2143 El Segundo, CA 90245
2144 manros@cp10.es.xerox.com

2145
2146 Carl Kugler
2147 Mail Stop 003G
2148 IBM Printing Systems Co
2149 6300 Diagonal Hwy
2150 Boulder CO 80301
2151 Kugler@us.ibm.com

2152
2153 Henrik Holst
2154 i-data Printing Systems
2155 Vadstrupvej 35-43
2156 2880 Bagsvaerd, Denmark
2157 hh@i-data.com

2158
2159 Peter Zehler
2160 Xerox Corporation
2161 800 Philips Road
2162 Webster, NY 14580
2163
2164 Phone: 716 265-8755
2165 peter.zehler@usa.xerox.com

2166 **10 Notices**

2167 The IETF takes no position regarding the validity or scope of any intellectual property or other rights that
2168 might be claimed to pertain to the implementation or use of the technology described in this document or

2169 the extent to which any license under such rights might or might not be available; neither does it represent
2170 that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to
2171 rights in standards-track and standards-related documentation can be found in BCP-11[BCP-11]. Copies of
2172 claims of rights made available for publication and any assurances of licenses to be made available, or the
2173 result of an attempt made to obtain a general license or permission for the use of such proprietary rights by
2174 implementers or users of this specification can be obtained from the IETF Secretariat.

2175 The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications,
2176 or other proprietary rights, which may cover technology that may be required to practice this standard.
2177 Please address the information to the IETF Executive Director.

2178 Copyright (C) The Internet Society (1999). All Rights Reserved

2179 This document and translations of it may be copied and furnished to others, and derivative works that
2180 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
2181 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
2182 this paragraph are included on all such copies and derivative works. However, this document itself may not
2183 be modified in any way, such as by removing the copyright notice or references to the Internet Society or
2184 other Internet organizations, except as needed for the purpose of developing Internet standards in which
2185 case the procedures for copyrights defined in the Internet Standards process must be followed, or as
2186 required to translate it into languages other than English.

2187 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its
2188 successors or assigns.

2189 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET
2190 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,
2191 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE
2192 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
2193 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

2194 **11 Change History (to be removed at time of RFC publishing)**

2195 The change history is in *reverse* chronological order:

2196 11.1 Changes from 000509 to 000530

2197 The following changes were made to the 5/09/00 version to make the 5/30/00 version:

- 2198 1. Added section 5.1 on General Directory Considerations which includes references to SLP and LDPA
2199 Printer schemas and their introduction of the "printer-xri-supported" attribute which combines "printer-
2200 uri-supported", "uri-security-supported", and "uri-authentication-supported" attributes.

2201 11.2 Changes from 990927 to 000509

- 2202 The following changes were made to the 9/27/99 version to make the 5/09/00 version:
- 2203 1. Table 5 - Corrected some attributes returned by Send-Document and Send-URI to be the same as Print-
2204 Job as in [ipp-mod].
 - 2205 2. Corrected several uses of 'client-error-bad-syntax' to be 'client-error-bad-request' as in the [ipp-mod].
 - 2206 3. Added section 3.1.3.1.8.1 to clarify what Resume-Printer does if the Printer is unable to resume the
2207 output device and section 3.1.3.1.8.2 about the "printer-state" for such a condition.
 - 2208 4. Added section 3.3.2 to indicate that on a Restart-Job that a Printer MUST re-fetch the document data
2209 when the job was created with Print-URI or Send-URI.
 - 2210 5. Section 4.1.4 - clarified that the length field for 'textWithLanguage' and 'nameWithLanguage' does *not*
2211 include the language field, so that the same maximum length of the data applies to the WithLanguage as
2212 the WithoutLanguage types, not counting the language field.
 - 2213 6. Added section 4.5 about empty jobs, i.e., with no documents. They are processed as any other job,
2214 possibly producing start and/or end sheets.
- 2215 11.3 Changes from 990914 to 990927
- 2216 1. Add comments about this document is also IPP/1.0 relevant.
 - 2217 2. Section 4.1.3: Add more examples of URI's with the port 631 and the ipp scheme.
 - 2218 3. Section 4.4.3: Move the DNS stuff to the 'how to compare URI's.
 - 2219 4. Section 4.4.3.2: Swap lines, first tell about the forgiven printer and then what the printer is allowed to
2220 do.
 - 2221 5. Fixed some errors in the Summary Attribute tables 1-5 and broke them into five portrait tables, so that it
2222 can be made into plain text for INTERNET-DRAFTS.
- 2223 11.4 Changes from 990726 to 990914:
- 2224 1. Added IPP/1.1 operations and attributes to table 1.
 - 2225 2. Validate version: Added text and table from issue 32
 - 2226 3. Printer-uri-supported: Added section 4.4.3
 - 2227 4. Added IPP/1.1 operations to section 3.1.2.1.4.3
 - 2228 5. Added answer to question "Should the server wait for the "last-document" operation attribute set to
2229 'true' before starting to "process" the job?" in section 3.2.4

- 2230 6. Changed 'server-error-uri-scheme-not-supported' to 'client-error-uri-scheme-not-supported' in section
2231 3.1.2.1.5 when talking about the 'document-uri' attribute.
- 2232 7. Added 'Suggested Operation Processing Steps' and 'Suggested Additional Processing Steps for
2233 Operations that Create/Validate Jobs and Add Document' flow-chart overview.
- 2234 11.5 Changes to produce the February 12, 1999 version from the January 8, 1999 version:
- 2235 1. Section 2.2.1.5: added check for document not found or accessible in Print-URI and Send-URI
- 2236 2. Section 3.6.2: Clarified that the IPP standard requires that servers **MUST** accept chunked requests
2237 and that clients **MUST** accept chunked responses, in spite of the lack of conformance of HTTP
2238 servers to the HTTP/1.1 requirement to support chunking.
- 2239 11.6 Changes to produce the January 8, 1999 version from the December 6, 1998 version:
- 2240 1. Added section 3.6.2: Warning About the Use of Chunked Requests with CGI Script
2241 Implementations
- 2242 2. Section 2.2.1.2: changed "printer-operations-supported" to "operations-supported".
- 2243 3. Section 2.2.1.6: changed "job-media-supported" to "job-media-sheets-supported"
- 2244 4. Section 2.2.3: separated the validation checks for variable length attributes into two separate tests:
2245 one for correct attribute syntax and one for correct length.
- 2246 5. Section 2.2.3: changed "multiple-document-handling-supported" to "printer-resolution-supported"
- 2247 6. Section 2.6.1: recommended that an IPP object also support US-ASCII charset.
- 2248 7. Section 3: Clarified that a server is not required to send a response until after it has received the
2249 client's entire request, but recommend that the server return a response as soon as possible if an error
2250 is detected while the client is still sending the data, rather than waiting until all of the data is
2251 received. Also recommended that a client listen for an error response that an IPP server **MAY** send
2252 before it receives all the data.
- 2253 11.7 Changes to produce the December 6, 1998 version from the November 16, 1998 version:
- 2254 Included all of the remaining agreed issues raised before the November 16, 1998 production of the Internet-
2255 Drafts for IPP/1.0 that included adding explanations to the Implementers Guide.
- 2256 Changes from 990422 to 990726:
- 2257 1. Encoding and Transport: Address issues 4, 5, 20 from Issues-raised-at-Bake-Off2.doc

- 2258 2. Decide whether to accept or reject the request: discuss issues 6, 9, 10
- 2259 3. Get-Printer-Attributes: add notes about printer-make-and-model and .INF files; issue 7
- 2260 4. Create-Job: clarify job-incoming vs. data-insufficient; issue 13
- 2261 5. Get-Printer Attributes: polling -- issue 16
- 2262 6. Job Description Attributes: ways to get time; issue 17
- 2263 7. Validate the values of the Job Template Attributes: clarify zero-length keywords; issue 22
- 2264 8. Validate Optional Operation Attributes: Note about checking for compression in IPP/1.0; issue 28
- 2265 9. Validate version number: advantages to backward compatibility; issue 33
- 2266 10. Note: examples for issue 2 seem to be covered sufficiently in the new MOD doc.