

1 INTERNET-DRAFT
2 draft-ietf-ipp-implementers-guide-v11-021.txt
3 Obsoletes RFC 2639

T. Hastings
Xerox Corporation
C. Manros
Xerox Corporation
C. Kugler
IBM Printing Systems Co
H. Holst
i-data Printing Systems
P. Zehler
Xerox Corporation
January 25, 2001~~May 30, 2000~~

Internet Printing Protocol/1.1: Implementer's Guide

Copyright (C) The Internet Society (2001~~0~~). All Rights Reserved.

17 Status of this Memo

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

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

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

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

26 Abstract

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

35 This document obsoletes RFC 2639 which was the Implementer's Guide for IPP/1.0.

36

36 The full set of IPP documents includes:

37 Design Goals for an Internet Printing Protocol [RFC2567]

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

39 Internet Printing Protocol/1.1: Model and Semantics [~~IPP-MOD~~[RFC2911](#)]

40 Internet Printing Protocol/1.1: Encoding and Transport [~~IPP-PRO~~[RFC2910](#)]

41 Mapping between LPD and IPP Protocols [RFC2569]

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

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

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

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

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

59

60

60 **TABLE OF CONTENTS**

61	1	Introduction.....	6
62	1.1	Conformance language	6
63	1.2	Other terminology.....	6
64	1.3	Issues Raised from Interoperability Testing Events.....	7
65	2	IPP Objects	7
66	3	IPP Operations	8
67	3.1	Common Semantics.....	8
68	3.1.1	Summary of Operation Attributes.....	8
69	3.1.2	Suggested Operation Processing Steps for IPP Objects.....	14
70	3.1.2.1	Suggested Operation Processing Steps for all Operations.....	15
71	3.1.2.1.1	Validate version number.....	16
72	3.1.2.1.2	Validate operation identifier.....	16
73	3.1.2.1.3	Validate the request identifier.....	17
74	3.1.2.1.4	Validate attribute group and attribute presence and order.....	17
75	3.1.2.1.4.1	Validate the presence and order of attribute groups	17
76	3.1.2.1.4.2	Ignore unknown attribute groups in the expected position.....	17
77	3.1.2.1.4.3	Validate the presence of a single occurrence of required Operation attributes	18
78	3.1.2.1.5	Validate the values of the REQUIRED Operation attributes	25
79	3.1.2.1.6	Validate the values of the OPTIONAL Operation attributes.....	28
80	3.1.2.2	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add Documents	31
81	3.1.2.2.1	Default "ipp-attribute-fidelity" if not supplied	31
82	3.1.2.2.2	Check that the Printer object is accepting jobs.....	32
83	3.1.2.2.3	Validate the values of the Job Template attributes	32
84	3.1.2.3	Algorithm for job validation.....	32
85	3.1.2.3.1	Check for conflicting Job Template attributes values.....	37
86	3.1.2.3.2	Decide whether to REJECT the request.....	37
87	3.1.2.3.3	For the Validate-Job operation, RETURN one of the success status codes	39
88	3.1.2.3.4	Create the Job object with attributes to support	39
89	3.1.2.3.5	Return one of the success status codes	40
90	3.1.2.3.6	Accept appended Document Content	41
91	3.1.2.3.7	Scheduling and Starting to Process the Job	41
92	3.1.2.3.8	Completing the Job.....	41
93	3.1.2.3.9	Destroying the Job after completion.....	41
94	3.1.2.3.10	Interaction with "ipp-attribute-fidelity".....	42
95	3.1.2.3.11	Character set code conversion support.....	42
96	3.1.2.3.12	What charset to return when an unsupported charset is requested (Issue 1.19)?.....	43
97	3.1.2.3.13	Natural Language Override (NLO)	43
98	3.1.3	Status codes returned by operation.....	44
99	3.1.3.1	Printer Operations.....	44
100			

101	3.1.3.1.1	Print-Job	45
102	3.1.3.1.2	Print-URI	46
103	3.1.3.1.3	Validate-Job.....	47
104	3.1.3.1.4	Create-Job	47
105	3.1.3.1.5	Get-Printer-Attributes	47
106	3.1.3.1.6	Get-Jobs	48
107	3.1.3.1.7	Pause-Printer.....	48
108	3.1.3.1.8	Resume-Printer.....	49
109	3.1.3.1.8.1	What about Printers unable to change state due to an error condition?.....	49
110	3.1.3.1.8.2	How is "printer-state" handled on Resume-Printer?.....	50
111	3.1.3.1.9	Purge-Printer	50
112	3.1.3.2	Job Operations	51
113	3.1.3.2.1	Send-Document.....	51
114	3.1.3.2.2	Send-URI.....	51
115	3.1.3.2.3	Cancel-Job.....	52
116	3.1.3.2.4	Get-Job-Attributes.....	52
117	3.1.3.2.5	Hold-Job.....	53
118	3.1.3.2.6	Release-Job.....	54
119	3.1.3.2.7	Restart-Job.....	54
120	3.1.3.2.7.1	Can documents be added to a restarted job?.....	54
121	3.1.4	Returning unsupported attributes in Get-Xxxx responses (Issue 1.18).....	54
122	3.1.5	Sending empty attribute groups.....	55
123	3.2	Printer Operations	55
124	3.2.1	Print-Job operation.....	55
125	3.2.1.1	Flow controlling the data portion of a Print-Job request (Issue 1.22)	55
126	3.2.1.2	Returning job-state in Print-Job response (Issue 1.30)	55
127	3.2.2	Get-Printer-Attributes operation.....	56
128	3.2.3	Get-Jobs operation.....	56
129	3.2.3.1	Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?.....	56
130	3.2.3.2	Why is there a "limit" attribute in the Get-Jobs operation?	57
131	3.2.4	Create-Job operation.....	57
132	3.3	Job Operations.....	57
133	3.3.1	Validate-Job.....	58
134	3.3.2	Restart-Job.....	58
135	4	Object Attributes.....	58
136	4.1	Attribute Syntax's.....	58
137	4.1.1	The 'none' value for empty sets (Issue 1.37).....	58
138	4.1.2	Multi-valued attributes (Issue 1.31)	58
139	4.1.3	Case Sensitivity in URIs (issue 1.6)	59
140	4.1.4	Maximum length for xxxWithLanguage and xxxWithoutLanguage	59
141	4.2	Job Template Attributes.....	60
142	4.2.1	multiple-document-handling(type2 keyword)	60
143	4.2.1.1	Support of multiple document jobs	60
144	4.3	Job Description Attributes	60
145	4.3.1	Getting the date and time of day	60
146	4.4	Printer Description Attributes.....	61
147	4.4.1	queued-job-count (integer(0:MAX))	61
148	4.4.1.1	Why is "queued-job-count" RECOMMENDED (Issue 1.14)?.....	61
149	4.4.1.2	Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?.....	61
150	4.4.2	printer-current-time (dateTime)	61
151	4.4.3	Printer-uri.....	62
152	4.5	Empty Jobs	62

153 5 Directory Considerations 62

154 5.1 General Directory Schema Considerations 62

155 5.2 IPP Printer with a DNS name 63

156 6 Security Considerations 63

157 6.1 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)... 63

158 7 Encoding and Transport..... 64

159 7.1 General Headers 65

160 7.2 Request Headers..... 65

161 7.3 Response Headers 66

162 7.4 Entity Headers..... 66

163 7.5 Optional support for HTTP/1.0..... 67

164 7.6 HTTP/1.1 Chunking 67

165 7.6.1 Disabling IPP Server Response Chunking..... 67

166 7.6.2 Warning About the Support of Chunked Requests..... 68

167 8 References 68

168 9 Authors' Address 70

169 10 Full Copyright Statement 70

170

171 **TABLES**

172

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

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

175 Table 3 - Summary of Job operation attributes that sender **MUST** supply 11

176 Table 4 - Summary of Job operation attributes that sender **MAY** supply 12

177 Table 5 - Printer operation response attributes..... 13

178 Table 6 - Examples of validating IPP version..... 16

179 Table 7 - Rules for validating single values X against Z..... 33

180

181

181
182
183

184 1 Introduction

185 The IPP Implementer's Guide (IIG) (this document) contains information that supplements the IPP Model and
186 Semantics [~~IPP-MOD~~RFC2911] and the IPP Transport and Encoding [~~IPP-PRO~~RFC2910] documents. As
187 such this information is not part of the formal specifications. Instead information is presented to help
188 implementers understand the specification, including some of the motivation for decisions taken by the
189 committee in developing the specification. Some of the implementation considerations are intended to help
190 implementers design their client and/or IPP object implementations. If there are any contradictions between
191 this document and [~~IPP-MOD~~RFC2911] or [~~IPP-PRO~~RFC2910], those documents take precedence over
192 this document.

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

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

197 1.1 Conformance language

198 Usually, this document does not contain the terminology MUST, MUST NOT, MAY, NEED NOT,
199 SHOULD, SHOULD NOT, REQUIRED, and OPTIONAL. However, when those terms do appear in this
200 document, their intent is to repeat what the [~~IPP-MOD~~RFC2911] and [~~IPP-PRO~~RFC2910] documents
201 require and allow, rather than specifying additional conformance requirements. These terms are defined in
202 section ~~12.43~~ on conformance terminology in [~~IPP-MOD~~RFC2911], most of which is taken from RFC 2119
203 [RFC2119].

204 Implementers should read section ~~12.43~~ (APPENDIX A) in [~~IPP-MOD~~RFC2911] in order to understand
205 these capitalized words. The words MUST, MUST NOT, and REQUIRED indicate what implementations
206 are required to support in a client or IPP object in order to be conformant to [~~IPP-MOD~~RFC2911] and
207 [~~IPP-PRO~~RFC2910]. MAY, NEED NOT, and OPTIONAL indicate was is merely allowed as an
208 implementer option. The verbs SHOULD and SHOULD NOT indicate suggested behavior, but which is not
209 required or disallowed, respectively, in order to conform to the specification.

210 1.2 Other terminology

211 This document uses other terms, such as "attributes", "operation", and "Printer" as defined in [RFC2911]
212 section 12. In addition, ~~T~~the term "sender" refers to the client that sends a request or an IPP object that
213 returns a response. The term "receiver" refers to the IPP object that receives a request and to a client that
214 receives a response.

215 1.3 Issues Raised from Interoperability ~~Bake-Offs~~ Testing Events

216 The IPP WG has conducted ~~two-three~~ open ~~i~~Interoperability ~~Testing Events~~ "~~Bake-Offs~~". The first ~~bake-one~~
217 ~~off~~ was held in September 1998, ~~and Bake-Off2~~ the second one was held in March 1999, and the third one
218 was held in October 2000. See the summary reports in:

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

220 The issues raised from the first Interoperability Testing Event ~~bake-off~~ are numbered 1.n in this document and
221 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~~ RFC2566] and the "IPP/1.0 Encoding and Transport" [~~IPP-PRO~~ RFC2565] documents. However,
225 some of the discussion is left here in the Implementer's Guide to help understanding.

226 The issues raised from the second Interoperability Testing Event ~~Bake-Off2~~ are numbered 2.n in this document
227 have been incorporated into "IPP/1.1 Model and Semantics" [RFC2911] and the "IPP/1.1 Encoding and
228 Transport" [RFC2910] documents. However, some of the discussion is left here in the Implementer's Guide
229 to help understanding.

230 and are described in:

231 <ftp://ftp.pwg.org/pub/pwg/ipp/issues/issues-raised-at-bake-off2.pdf> The issues raised from the third
232 Interoperability Testing Event are numbered 3.n in this document and are described in:

233 <ftp://ftp.pwg.org/pub/pwg/ipp/Issues/Issues-raised-at-Bake-Off3.pdf>

234 <ftp://ftp.pwg.org/pub/pwg/ipp/Issues/Issues-raised-at-Bake-Off3.doc>

235 <ftp://ftp.pwg.org/pub/pwg/ipp/Issues/Issues-raised-at-Bake-Off3.txt>

236 2 IPP Objects

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

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

244

244 **3 IPP Operations**

245 This section corresponds to Section 3 "IPP Operations" in the IPP/1.1 Model and Semantics document [[IPP-](#)
246 [MODRFC2911](#)].

247 **3.1 Common Semantics**

248 This section discusses semantics common to all operations.

249 **3.1.1 Summary of Operation Attributes**

250 Legend for the following table:

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

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

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

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

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

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

258

259

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

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

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

260

261

261

262

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	

263

264

264

265

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.

266

267

267

268

Table 5 - Printer operation response attributes

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

269

270

270

271 **3.1.2 Suggested Operation Processing Steps for IPP Objects**

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

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

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

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

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

298

298 **3.1.2.1 Suggested Operation Processing Steps for all Operations**

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

302	IIG Sect #	Flow	IPP error status codes
303	-----	----	-----
304			
305		v	err
306	3.1.2.1.1	<Validate version>	--> server-error-version-not-
307			supported
308		ok	
309		v	err
310	3.1.2.1.2	<Validate operation>	--> server-error-operation-not-
311			supported
312		ok	
313		v	err
314	3.1.2.1.4.1-	<Validate presence>	--> client-error-bad-request
315	3.1.2.1.4.2	<of attributes>	
316		ok	
317		v	err
318	3.1.2.1.4.3	<Validate presence>	--> client-error-bad-request
319		<of operation attr>	
320		ok	
321		v	err
322	3.1.2.1.5	<Valied values of>	--> client-error-bad-request
323		<operation attrs>	client-error-request-value-
324			too-long
325		<(length, tag, range,>	
326		<multi-value)>	
327		ok	
328		v	err
329	3.1.2.1.5	<Validate values>	--> client-error-bad-request
330		<with supported values>	client-error-charset-not-
331			supported
332		ok	client-error-attributes-or-
333			values-
334			not-supported
335		v	err
336	3.1.2.1.6	<Validate optionally>	--> client-error-bad-request
337		<operation attr>	client-error-natural-language-
338			not-supported
339			client-error-request-value-
340			too-long
341			client-error-attributes-or-
342			values-not-supported

343

344 **3.1.2.1.1 Validate version number**

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

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

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

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

361

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

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

367 **3.1.2.1.2 Validate operation identifier**

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

371 **3.1.2.1.3 Validate the request identifier**

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

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

377 **3.1.2.1.4 Validate attribute group and attribute presence and order**

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

379 **3.1.2.1.4.1 Validate the presence and order of attribute groups**

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

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

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

392 **3.1.2.1.4.2 Ignore unknown attribute groups in the expected position**

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

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

4 0 2 Note: By validating that requests are in the proper form, IPP objects force clients to use the proper form
 4 0 3 which, in turn, increases the chances that customers will be able to use such clients from multiple vendors with
 4 0 4 IPP objects from other vendors.

4 0 5 3.1.2.1.4.3 Validate the presence of a single occurrence of required Operation attributes

4 0 6 Client requests and IPP object responses contain Operation attributes that [~~IPP-MO~~RFC2911] Section 3
 4 0 7 requires to be present. Attributes within a group may be in any order, except for the ordering of target,
 4 0 8 charset, and natural languages attributes. These attributes MUST be first, and MUST be supplied in the
 4 0 9 following order: charset, natural language, and then target. An IPP object verifies that the attributes that
 4 1 0 Section 4 requires to be supplied by the client have been supplied in the request (attributes without an * in the
 4 1 1 following tables). An asterisk (*) indicates groups and Operation attributes that the client may omit in a
 4 1 2 request or an IPP object may omit in a response.

4 1 3 If an IPP object receives a request with required attributes missing or repeated from a group or in the wrong
 4 1 4 position, the behavior of the IPP object is IMPLEMENTATION DEPENDENT. Some of the possible
 4 1 5 implementations are:

- 4 1 6 REJECTS the request and RETURNS the 'client-error-bad-request' status code
- 4 1 7 accepts the request and uses the first occurrence of the attribute no matter where it is
- 4 1 8 accepts the request and uses the last occurrence of the attribute no matter where it is
- 4 1 9 accept the request and assume some default value for the missing attribute

4 2 0 Therefore, client MUST send conforming requests, if they want to receive the same behavior from all IPP
 4 2 1 object implementations. For example, it is an error for the "attributes-charset" or "attributes-natural-language"
 4 2 2 attribute to be omitted in any operation request, or for an Operation attribute to be supplied in a Job Template
 4 2 3 group or a Job Template attribute to be supplied in an Operation Attribute group in a create request. It is also
 4 2 4 an error to supply the "attributes-charset" attribute twice.

4 2 5 Since these kinds of attribute errors are most likely to be detected by a client developer rather than by a
 4 2 6 customer, the IPP object NEED NOT return an indication of which attribute was in error in either the
 4 2 7 Unsupported Attributes group or the Status Message. Also, the IPP object NEED NOT find all attribute
 4 2 8 errors before returning this error.

4 2 9 The following tables list all the attributes for all the operations by attribute group in each request and each
 4 3 0 response. The order of the groups is the order that the client supplies the groups as specified in [~~IPP-~~
 4 3 1 ~~MO~~RFC2911] Section 3. The order of the attributes within a group is arbitrary, except as noted for some
 4 3 2 of the special operation attributes (charset, natural language, and target). The tables below use the following
 4 3 3 notation:

- 4 3 4 R indicates a REQUIRED attribute or operation that an IPP object MUST support
- 4 3 5 O indicates an OPTIONAL attribute or operation that an IPP object NEED NOT support

436 * indicates that a client MAY omit the attribute in a request and that an IPP object MAY omit the
437 attribute in a response. The absence of an * means that a client MUST supply the
438 attribute in a request and an IPP object MUST supply the attribute in a response.
439 + indicates that this is not a IPP/1.0 operation, but is only a part of IPP/1.1 and future versions of
440 IPP.

441

442 Operation Requests

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

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

446

447 Print-Job Request (R):

448 Group 1: Operation Attributes (R)

449 attributes-charset (R)

450 attributes-natural-language (R)

451 printer-uri (R)

452 requesting-user-name (R*)

453 job-name (R*)

454 ipp-attribute-fidelity (R*)

455 document-name (R*)

456 document-format (R*)

457 document-natural-language (O*)

458 compression (O*)

459 job-k-octets (O*)

460 job-impressions (O*)

461 job-media-sheets (O*)

462 Group 2: Job Template Attributes (R*)

463 <Job Template attributes> (O*)

464 (see [[IPP-MODRFC2911](#)] Section 4.2)

465 Group 3: Document Content (R)

466 <document content>

467

468 Validate-Job Request (R):

469 Group 1: Operation Attributes (R)

470 attributes-charset (R)

471 attributes-natural-language (R)

472 printer-uri (R)

473 requesting-user-name (R*)

474 job-name (R*)

475 ipp-attribute-fidelity (R*)

476 document-name (R*)

477 document-format (R*)

478 document-natural-language (O*)

```
479         compression (O*)
480         job-k-octets (O*)
481         job-impressions (O*)
482         job-media-sheets (O*)
483     Group 2: Job Template Attributes (R*)
484         <Job Template attributes> (O*)
485         (see [IPP-MODRFC2911] Section 4.2)
486
487 Print-URI Request (O):
488     Group 1: Operation Attributes (R)
489         attributes-charset (R)
490         attributes-natural-language (R)
491         printer-uri (R)
492         document-uri (R)
493         requesting-user-name (R*)
494         job-name (R*)
495         ipp-attribute-fidelity (R*)
496         document-name (R*)
497         document-format (R*)
498         document-natural-language (O*)
499         compression (O*)
500         job-k-octets (O*)
501         job-impressions (O*)
502         job-media-sheets (O*)
503     Group 2: Job Template Attributes (R*)
504         <Job Template attributes> (O*) (see
505         (see [IPP-MODRFC2911] Section 4.2)
506
507 Create-Job Request (O):
508     Group 1: Operation Attributes (R)
509         attributes-charset (R)
510         attributes-natural-language (R)
511         printer-uri (R)
512         requesting-user-name (R*)
513         job-name (R*)
514         ipp-attribute-fidelity (R*)
515         job-k-octets (O*)
516         job-impressions (O*)
517         job-media-sheets (O*)
518     Group 2: Job Template Attributes (R*)
519         <Job Template attributes> (O*) (see
520         (see [IPP-MODRFC2911] Section 4.2)
521
522 Get-Printer-Attributes Request (R):
523     Group 1: Operation Attributes (R)
524         attributes-charset (R)
525         attributes-natural-language (R)
526         printer-uri (R)
```

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

575 requesting-user-name (R*)
576 message (O*)
577
578 Get-Job-Attributes Request (R):
579 Group 1: Operation Attributes (R)
580 attributes-charset (R)
581 attributes-natural-language (R)
582 (printer-uri & job-id) | job-uri (R)
583 requesting-user-name (R*)
584 requested-attributes (R*)
585
586 Pause-Printer Request (O+):
587 Resume-Printer Request (O+):
588 Purge-Printer Request (O+):
589 Group 1: Operation Attributes (R)
590 attributes-charset (R)
591 attributes-natural-language (R)
592 printer-uri (R)
593 requesting-user-name (R*)
594
595 Hold-Job Request (O+):
596 Restart-Job Request (O+):
597 Group 1: Operation Attributes (R)
598 attributes-charset (R)
599 attributes-natural-language (R)
600 (printer-uri & job-id) | job-uri (R)
601 requesting-user-name (R*)
602 job-hold-until (R*)
603 message (O*)

604

605 Operation Responses

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

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

609

610 Print-Job Response (R):
611 Create-Job Response (O):
612 Send-Document Response (O):
613 Group 1: Operation Attributes (R)
614 attributes-charset (R)
615 attributes-natural-language (R)
616 status-message (O*)
617 detailed-status-message (O*)
618 Group 2: Unsupported Attributes (R*) (see Note 3)
619 <unsupported attributes> (R*)

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

668 <requested attributes> (R*)
669
670 Get-Jobs Response (R):
671 Group 1: Operation Attributes (R)
672 attributes-charset (R)
673 attributes-natural-language (R)
674 status-message (O*)
675 detailed-status-message (O*)
676 Group 2: Unsupported Attributes (R*) (see Note 4)
677 <unsupported attributes> (R*)
678 Group 3: Job Object Attributes (R*) (see Note 2, 5)
679 <requested attributes> (R*)
680
681 Get-Job-Attributes Response (R):
682 Group 1: Operation Attributes (R)
683 attributes-charset (R)
684 attributes-natural-language (R)
685 status-message (O*)
686 detailed-status-message (O*)
687 Group 2: Unsupported Attributes (R*) (see Note 4)
688 <unsupported attributes> (R*)
689 Group 3: Job Object Attributes (R*) (see Note 2)
690 <requested attributes> (R*)
691
692 Pause-Printer Response (O+):
693 Resume-Printer Response (O+):
694 Purge-Printer Response (O+):
695 Group 1: Operation Attributes (R)
696 attributes-charset (R)
697 attributes-natural-language (R)
698 status-message (O*)
699 detailed-status-message (O*)
700 Group 2: Unsupported Attributes (R*) (see Note 4)
701 <unsupported attributes> (R*)
702

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

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

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

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

711 3.1.2.1.5 Validate the values of the REQUIRED Operation attributes

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

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

- 716 a) that the length of each Operation attribute value is correct for the attribute syntax tag supplied
 717 by the client according to [~~IPP-MO~~RFC2911] Section 4.1,
- 718 b) that the attribute syntax tag is correct for that Operation attribute according to [~~IPP-~~
 719 ~~MO~~RFC2911] Section 3,
- 720 c) that the value is in the range specified for that Operation attribute according to [~~IPP-~~
 721 ~~MO~~RFC2911] Section 3,
- 722 d) that multiple values are supplied by the client only for operation attributes that are multi-valued,
 723 i.e., that are 1setOf X according to [~~IPP-MO~~RFC2911] Section 3.

724

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

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

737 -----

738 attributes-charset (charset)

739 IF NOT a single non-empty 'charset' value, REJECT/RETURN 'client-error-bad-request'.
 740 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-long'.
 741 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-
 742 charset-not-supported".
 743

744 attributes-natural-language(naturalLanguage)

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

751

752 requesting-user-name

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

757

758 job-name(name)

759 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
760 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
761 long'.
762 IF NOT supplied by the client, the Printer object creates a name from the document-name or
763 document-uri.

764

765 document-name (name)

766 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
767 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
768 long'.
769

770 ipp-attribute-fidelity (boolean)

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

776 document-format (mimeMediaType)

777 IF NOT a single non-empty 'mimeMediaType' value, REJECT/RETURN 'client-error-bad-request'.
778 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
779 long'.
780 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN 'client-
781 error-document-format-not-supported'

782 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's "document-
783 format-default" attribute.

784
785 document-uri (uri)

786 IF NOT a single non-empty 'uri' value, REJECT/RETURN 'client-error-bad-request'.
787 IF the value length is greater than 1023 octets, REJECT/RETURN 'client-error-request-value-too-
788 long'.
789 IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.
790 If the client-supplied URI scheme is not supported, i.e. the value is not in the Printer object's
791 referenced-uri-scheme-supported" attribute, the Printer object MUST reject the request and
792 return the 'client-error-uri-scheme-not-supported' status code. The Printer object MAY
793 check to see if the document exists and is accessible. If the document is not found or is not
794 accessible, REJECT/RETURN 'client-error-not found'.

795 last-document (boolean)

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

799
800 job-id (integer(1:MAX))

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

805
806 requested-attributes (1setOf keyword)

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

813
814 which-jobs (type2 keyword)

815 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
816 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
817 long'.
818 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to the
819 Unsupported Attributes response group and REJECT/RETURN 'client-error-attributes-or-
820 values-not-supported'.

821 Note: a Printer still supports the 'completed' value even if it keeps no completed/canceled/aborted
822 jobs: by returning no jobs when so queried.

823 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.

824

825 my-jobs (boolean)

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

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

829 IF NOT supplied by the client, the IPP object assumes the 'false' value.

830

831 limit (integer(1:MAX))

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

834 IF NOT supplied by the client, the IPP object returns all jobs, no matter how many.

835

836 -----

837

838 **3.1.2.1.6 Validate the values of the OPTIONAL Operation attributes**

839 OPTIONAL Operation attributes are those that an IPP object MAY ~~or MAY NOT~~ support. An IPP object
840 validates the values of the OPTIONAL attributes supplied by the client. The IPP object performs the same
841 syntactic validation checks for each OPTIONAL attribute value as in Section 3.1.2.1.5. As in Section
842 3.1.2.1.5, if any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' or
843 the 'client-error-request-value-too-long' status code.

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

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

851 -----

852 document-natural-language (naturalLanguage)

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

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

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

857

858 compression (type3 keyword)

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

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

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

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

868

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

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

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

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

875

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

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

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

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

882

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

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

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

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

889

890 message (text(127))

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

892 IF the value length is greater than 127 octets,

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

894

895 unknown or unsupported attribute

896

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

897

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

898

ELSE copy the attribute and value to the Unsupported Attributes response group and change the attribute

899

value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.

900

901

Note: Future Operation attributes may be added to the protocol specification that may occur anywhere in the

902

specified group. When the operation is otherwise successful, the IPP object returns the 'successful-ok-

903

ignored-or-substituted-attributes' status code. Ignoring unsupported Operation attributes in all operations is

904

analogous to the handling of unsupported Job Template attributes in the create and Validate-Job operations

905

when the client supplies the "ipp-attribute-fidelity" Operation attribute with the 'false' value. This last rule is so

906

that we can add OPTIONAL Operation attributes to future versions of IPP so that older clients can inter-

907

work with new IPP objects and newer clients can inter-work with older IPP objects. (If the new attribute

908

cannot be ignored without performing unexpectedly, the major version number would have been increased in

909

the protocol document and in the request). This rule for Operation attributes is independent of the value of the

910

"ipp-attribute-fidelity" attribute. For example, if an IPP object doesn't support the OPTIONAL "job-k-

911

octets" attribute', the IPP object treats "job-k-octets" as an unknown attribute and only checks the length for

912

the 'integer' attribute syntax supplied by the client. If it is not four octets, the IPP object REJECTS the request

913

and RETURNS the 'client-error-bad-request' status code, else the IPP object copies the attribute to the

914

Unsupported Attribute response group, setting the value to the "out-of-band" 'unsupported' value, but

915

otherwise ignores the attribute.

916

916

916 **3.1.2.2 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add**
 917 **Documents**

918 This section in combination with the previous section recommends
 919 the processing steps for the Print-Job, Validate-Job, Print-URI,
 920 Create-Job, Send-Document, and Send-URI operations that IPP
 921 objects SHOULD use. These are the operations that create jobs,
 922 validate a Print-Job request, and add documents to a job.

923 IIG Sect #	Flow	IPP error status codes
924 -----	----	-----
926		
927	v	No
928 3.1.2.2.1	<ipp-attribute-fidelity>	-----+
929	<supplied?>	
930	Yes	
931		ipp-attribute-fidelity = no
932		<-----+
933	v	No
934 3.1.2.2.2	<Printer is>	--> server-error-not-accepting-
935 jobs		
936	<accepting jobs?>	
937	Yes	
938	v	err
939 3.1.2.3	<Validate values of>	--> client-error-bad-request
940 <Job template attributes>		client-error-request-value-
941 too-		
942		long
943	<(length, tag, range,>	
944	<multi-value)>	
945	ok	
946	v	err
947 3.1.2.3	<Validate values with>	--> client-error-bad-request
948 <supported values>		client-error-attributes-or-
949		values-not-supported
950	v	err
951 3.1.2.3.1	<Any conflicting>	--> client-error-conflicting-
952 <Job Template attr values>		attributes
953		client-error-attributes-or-
954		values-not-supported
955	v	

956 **3.1.2.2.1 Default "ipp-attribute-fidelity" if not supplied**

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

959 3.1.2.2 Check that the Printer object is accepting jobs

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

962 3.1.2.3 Validate the values of the Job Template attributes

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

- 966 a) that the length of each value is correct for the attribute syntax tag supplied by the client
967 according to [~~IPP-MOD~~[RFC2911](#)] Section 4.1.
- 968 b) that the attribute syntax tag is correct for that attribute according to [~~IPP-MOD~~[RFC2911](#)]
969 Sections 4.2 to 4.4.
- 970 c) that multiple values are supplied only for multi-valued attributes, i.e., that are 1setOf X
971 according to [~~IPP-MOD~~[RFC2911](#)] Sections 4.2 to 4.4.

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

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

- 981 1. reject the operation and return the 'client-error-bad-request' error status code
- 982 2. accept the operation and use the first occurrence of the attribute
- 983 3. accept the operation and use the last occurrence of the attribute

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

986 3.1.2.3 Algorithm for job validation

987 The process of validating a Job-Template attribute "xxx" against a Printer attribute "xxx-supported" can use
988 the following validation algorithm (see section 3.2.1.2 in [~~ipp-mod~~[RFC2911](#)]).

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

- 991 1. If U is multi-valued, validate each value X of U by performing the algorithm in Table 7 with each value
 992 X. Each validation is separate from the standpoint of returning unsupported values. Example: If U is
 993 "finishings" that the client supplies with 'staple', 'bind' values, then X takes on the successive values:
 994 'staple', then 'bind'
- 995 2. If V is multi-valued, validate X against each Z of V by performing the algorithm in Table 7 with each
 996 value Z. If a value Z validates, the validation for the attribute value X succeeds. If it fails, the algorithm
 997 is applied to the next value Z of V. If there are no more values Z of V, validation fails. Example" If V is
 998 "sides-supported" with values: 'one-sided', 'two-sided-long', and 'two-sided-short', then Z takes on the
 999 successive values: 'one-sided', 'two-sided-long', and 'two-sided-short'. If the client supplies "sides"
 000 with 'two-sided-long', the first comparison fails ('one-sided' is not equal to 'two-sided-long'), the
 001 second comparison succeeds ('two-sided-long' is equal to 'two-sided-long'), and the third comparison
 002 ('two-sided-short' with 'two-sided-long') is not even performed.
- 003 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.

004 **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.

005

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

013 If some Job Template attributes are supported for some document formats and not for others or the values are
 014 different for different document formats, the IPP object SHOULD take that into account in this validation
 015 using the value of the "document-format" supplied by the client (or defaulted to the value of the Printer's
 016 "document-format-default" attribute, if not supplied by the client). For example, if "number-up" is supported
 017 for the 'text/plain' document format, but not for the 'application/postscript' document format, the check
 018 SHOULD (though it NEED NOT) depend on the value of the "document-format" operation attribute. See
 019 "document-format" in [[IPP-MODRFC2911](#)] section 3.2.1.1 and 3.2.5.1.

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

023 -----

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

025 IF NOT a single 'integer' value with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
026 request'.

027 IF NOT supplied by the client, use the value of the Printer object's "job-priority-default" attribute at job
028 submission time.

029 IF NOT in the range 1 to 100, inclusive, copy the attribute and the unsupported value to the Unsupported
030 Attributes response group.

031 Map the value to the nearest supported value in the range 1:100 as specified by the number of discrete
032 values indicated by the value of the Printer's "job-priority-supported" attribute. See the formula in
033 [[IPP-MODRFC2911](#)] Section 4.2.1.

034

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

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

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

038 IF NOT supplied by the client, use the value of the Printer object's "job-hold-until" attribute at job
039 submission time.

040 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the unsupported
041 value to the Unsupported Attributes response group.

042

043 job-sheets (type3 keyword | name)

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

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

046 IF NOT in the Printer object's "job-sheets-supported" attribute, copy the attribute and the unsupported
047 value to the Unsupported Attributes response group.

048

049 multiple-document-handling (type2 keyword)

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

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

052 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute and the
053 unsupported value to the Unsupported Attributes response group.

054

055 copies (integer(1:MAX))

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

057 REJECT/RETURN 'client-error-bad-request'.
058 IF NOT in range of the Printer object's "copies-supported" attribute
059 copy the attribute and the unsupported value to the Unsupported Attributes response group.
060
061 finishings (1setOf type2 enum)

062 IF NOT an 'enum' value(s) each with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
063 request'.
064 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the unsupported
065 value(s), but not any supported values, to the Unsupported Attributes response group.
066
067 page-ranges (1setOf rangeOfInteger(1:MAX))

068 IF NOT a 'rangeOfInteger' value(s) each with a length equal to 8 octets, REJECT/RETURN 'client-error-
069 bad-request'.
070 IF first value is greater than second value in any range, the ranges are not in ascending order, or ranges
071 overlap, REJECT/RETURN 'client-error-bad-request'.
072 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to the
073 Unsupported Attributes response group and set the value to the "out-of-band" 'unsupported' value.
074
075 sides (type2 keyword)

076 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
077 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
078 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported value to
079 the Unsupported Attributes response group.
080
081 number-up (integer(1:MAX))

082 IF NOT a single 'integer' value with a length equal to 4 octets,
083 REJECT/RETURN 'client-error-bad-request'.
084 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"
085 attribute, copy the attribute and value to the Unsupported Attribute response group.
086
087 orientation-requested (type2 enum)

088 IF NOT a single 'enum' value with a length equal to 4 octets,
089 REJECT/RETURN 'client-error-bad-request'.
090 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the
091 unsupported value to the Unsupported Attributes response group.
092
093 media (type3 keyword | name)

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

095 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
 096 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported value to
 097 the Unsupported Attributes response group.

098

099 printer-resolution (resolution)

100 IF NOT a single 'resolution' value with a length equal to 9 octets,
 101 REJECT/RETURN 'client-error-bad-request'.

102 IF NOT in the Printer object's "printer-resolution-supported" attribute, copy the attribute and the
 103 unsupported value to the Unsupported Attributes response group.

104

105 print-quality (type2 enum)

106 IF NOT a single 'enum' value with a length equal to 4 octets,
 107 REJECT/RETURN 'client-error-bad-request'.

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

110

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

112 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute syntax,
 113 REJECT/RETURN 'client-error-bad-request' if the length of the attribute syntax is fixed or 'client-error-
 114 request-value-too-long' if the length of the attribute syntax is variable.

115 ELSE copy the attribute and value to the Unsupported Attributes response group and change the attribute
 116 value to the "out-of-band" 'unsupported' value. Any remaining Job Template Attributes are either
 117 unknown or unsupported Job Template attributes and are validated algorithmically according to
 118 their attribute syntax for proper length (see below).

119

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

126	Name	Octet length check for read-write attributes
127	-----	-----
128	'textWithLanguage	<= 1023 AND 'naturalLanguage' <= 63
129	'textWithoutLanguage'	<= 1023
130	'nameWithLanguage'	<= 255 AND 'naturalLanguage' <= 63
131	'nameWithoutLanguage'	<= 255
132	'keyword'	<= 255
133	'enum'	= 4
134	'uri'	<= 1023

```
135      'uriScheme'           <= 63
136      'charset'            <= 63
137      'naturalLanguage'    <= 63
138      'mimeType'           <= 255
139      'octetString'        <= 1023
140      'boolean'            = 1
141      'integer'            = 4
142      'rangeOfInteger'     = 8
143      'dateTime'           = 11
144      'resolution'         = 9
145      'lsetOf X'
```

146

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

152 **3.1.2.3.1 Check for conflicting Job Template attributes values**

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

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

165 **3.1.2.3.2 Decide whether to REJECT the request**

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

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

172

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

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

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

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

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

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

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

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

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

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

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

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

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

212 **3.1.2.3.3 For the Validate-Job operation, RETURN one of the success status codes**

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

- 214 1. the "successful-ok" status code, if there are no unsupported or conflicting Job Template attributes or
215 values.
- 216 2. the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or values.
- 217 3. the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template
218 attributes or values.
219

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

224 **3.1.2.3.4 Create the Job object with attributes to support**

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

- 226 1. creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and initializes
227 all of the job's other supported Job Description attributes.
- 228 2. removes all unsupported attributes from the Job object.
- 229 3. for each unsupported value, removes either the unsupported value or substitutes the unsupported
230 attribute value with some supported value. If an attribute has no values after removing unsupported
231 values from it, the attribute is removed from the Job object (so that the normal default behavior at job
232 processing time will take place for that attribute).
- 233 4. for each conflicting value, removes either the conflicting value or substitutes the conflicting attribute
234 value with some other supported value. If an attribute has no values after removing conflicting values
235 from it, the attribute is removed from the Job object (so that the normal default behavior at job
236 processing time will take place for that attribute).
237

238 If there were no attributes or values flagged as unsupported, or the value of 'ipp-attribute-fidelity' was 'false',
239 the Printer object is able to accept the create request and create a new Job object. If the "ipp-attribute-
240 fidelity" attribute is set to 'true', the Job Template attributes that populate the new Job object are necessarily all
241 the Job Template attributes supplied in the create request. If the "ipp-attribute-fidelity" attribute is set to 'false',
242 the Job Template attributes that populate the new Job object are all the client supplied Job Template attributes
243 that are supported or that have value substitution. Thus, some of the requested Job Template attributes **may**
244 **will** not appear in the Job object because the Printer object did not support those attributes. The attributes
245 that populate the Job object are persistently stored with the Job object for that Job. A Get-Job-Attributes
246 operation on that Job object will return only those attributes that are persistently stored with the Job object.

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

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

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

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

268 3.1.2.3.5 Return one of the success status codes

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

- 270 1. the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes or
271 values.
- 272 2. the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template attribute
273 or values.

274 3. the 'successful-ok-ignored-or-substituted-attributes' status code, if there are only unsupported Job
275 Template attributes or values.

276

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

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

283 **3.1.2.3.6 Accept appended Document Content**

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

286 **3.1.2.3.7 Scheduling and Starting to Process the Job**

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

292 **3.1.2.3.8 Completing the Job**

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

297 **3.1.2.3.9 Destroying the Job after completion**

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

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

3.1.2.3.10 Interaction with "ipp-attribute-fidelity"

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

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

3.1.2.3.11 Character set code conversion support

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

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

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

According to the [IPP-ModRFC2911](#) document (section 3.1.4.2), the value of the "attributes-charset" for the response of the second request must be "us-ascii" since that is the charset specified in the request. The "job-name" value, however, is in "utf-8" format. Should the request be rejected even though both "utf-8" and "us-ascii" charsets are supported by the server? or should the "job-name" value be converted to "us-ascii" and return "successful-ok-conflicting-attributes" (0x0002) as the status code?

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

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

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

345 **3.1.2.3.12 What charset to return when an unsupported charset is requested (Issue 1.19)?**

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

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

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

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

356 Furthermore, [~~ipp-mod~~[RFC2911](#)] section 14.1.4.14 client-error-charset-not-supported (0x040D) was
357 clarified in November 1998 as follows:

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

361 **3.1.2.3.13 Natural Language Override (NLO)**

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

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

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

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

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

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

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

401 **3.1.3 Status codes returned by operation**

402 This section corresponds to [~~IPP-MOD~~RFC2911] section 3.1.6 "Operation Response Status Codes and
403 Status Messages". This section lists all status codes once in the first operation (Print-Job). Then it lists the
404 status codes that are different or specialized for subsequent operations under each operation.

405 **3.1.3.1 Printer Operations**

406 **3.1.3.1.1 Print-Job**

407 The Printer object MUST return one of the following "status-code" values for the indicated reason. Whether
408 all of the document data has been accepted or not before returning the success or error response depends on
409 implementation. See Section 13 in [[IPP-MODRFC2911](#)] for a more complete description of each status
410 code.

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

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

414 successful-ok-ignored-or-substituted-attributes: some supplied (1) attributes were ignored or (2)
415 unsupported attribute syntaxes or values were substituted with supported values or were ignored.

416 Unsupported attributes, attribute syntax's, or values MUST be returned in the Unsupported Attributes
417 group of the response.

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

422

423 [[ipp-modRFC2911](#)] section 3.1.6 Operation Status Codes and Messages states:

424 If the Printer object supports the "status-message" operation attribute, it SHOULD use the REQUIRED
425 'utf-8' charset to return a status message for the following error status codes (see section 13 in [[IPP-](#)
426 [MODRFC2911](#)]): 'client-error-bad-request', 'client-error-charset-not-supported', 'server-error-internal-
427 error', 'server-error-operation-not-supported', and 'server-error-version-not-supported'. In this case, it
428 MUST set the value of the "attributes-charset" operation attribute to 'utf-8' in the error response.

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

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

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

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

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

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

440 client-error-timeout: not applicable.

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

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

4 4 3 client-error-request-entity-too-large: the size of the request and/or print data exceeds the capacity of
4 4 4 the IPP Printer to process it.

4 4 5 client-error-request-value-too-long: the size of request variable length attribute values, such as 'text'
4 4 6 and 'name' attribute syntax's, exceed the maximum length specified in [[IPP-MODRFC2911](#)] for
4 4 7 the attribute and MUST be returned in the Unsupported Attributes Group.

4 4 8 client-error-document-format-not-supported: the document format supplied is not supported. The
4 4 9 "document-format" attribute with the unsupported value MUST be returned in the Unsupported
4 5 0 Attributes Group. This error SHOULD take precedence over any other 'xxx-not-supported'
4 5 1 error, except 'client-error-charset-not-supported'.

4 5 2 client-error-attributes-or-values-not-supported: one or more supplied attributes, attribute syntax's, or
4 5 3 values are not supported and the client supplied the "ipp-attributes-fidelity" operation attribute with
4 5 4 a 'true' value. They MUST be returned in the Unsupported Attributes Group as explained below.

4 5 5 client-error-uri-scheme-not-supported: not applicable.

4 5 6 client-error-charset-not-supported: the charset supplied in the "attributes-charset" operation attribute is
4 5 7 not supported. The Printer's "configured-charset" MUST be returned in the response as the value
4 5 8 of the "attributes-charset" operation attribute and used for any 'text' and 'name' attributes returned
4 5 9 in the error response. This error SHOULD take precedence over any other error, unless the
4 6 0 request syntax is so bad that the client's supplied "attributes-charset" cannot be determined.

4 6 1 client-error-conflicting-attributes: one or more supplied attribute values conflicted with each other and
4 6 2 the client supplied the "ipp-attributes-fidelity" operation attribute with a 'true' value. They MUST
4 6 3 be returned in the Unsupported Attributes Group as explained below.

4 6 4 server-error-internal-error: an unexpected condition prevents the request from being fulfilled.

4 6 5 server-error-operation-not-supported: not applicable (since Print-Job is REQUIRED).

4 6 6 server-error-service-unavailable: the service is temporarily overloaded.

4 6 7 server-error-version-not-supported: the version in the request is not supported. The "closest" version
4 6 8 number supported MUST be returned in the response.

4 6 9 server-error-device-error: a device error occurred while receiving or spooling the request or
4 7 0 document data or the IPP Printer object can only accept one job at a time.

4 7 1 server-error-temporary-error: a temporary error such as a buffer full write error, a memory overflow,
4 7 2 or a disk full condition occurred while receiving the request and/or the document data.

4 7 3 server-error-not-accepting-jobs: the Printer object's "printer-is-not-accepting-jobs" attribute is 'false'.

4 7 4 server-error-busy: the Printer is too busy processing jobs to accept another job at this time.

4 7 5 server-error-job-canceled: the job has been canceled by an operator or the system while the client
4 7 6 was transmitting the document data.

4 7 7 3.1.3.1.2 Print-URI

4 7 8 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Print-
4 7 9 URI with the following specializations and differences. See Section 14 for a more complete description of
4 8 0 each status code.

4 8 1 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation attribute
4 8 2 is not supported and is returned in the Unsupported Attributes group.

483 server-error-operation-not-supported: the Print-URI operation is not supported.
484

485 3.1.3.1.3 Validate-Job

486 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to
487 Validate-Job. See Section 13 in [~~IPP-MO~~[RFC2911](#)] for a more complete description of each status code.

488 3.1.3.1.4 Create-Job

489 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Create-
490 Job with the following specializations and differences. See Section 13 in [~~IPP-MO~~[RFC2911](#)] for a more
491 complete description of each status code.

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

495 3.1.3.1.5 Get-Printer-Attributes

496 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the Get-
497 Printer-Attributes operation with the following specialization's and differences. See Section 13 in [~~IPP-~~
498 [RFC2911](#)] for a more complete description of each status code.

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

500 successful-ok: no ~~request-operation~~ attributes ~~or values~~ were substituted or ignored (same as Print-Job)
501 and no requested attributes were unsupported.

502 successful-ok-ignored-or-substituted-attributes: ~~same as Print-Job, except for this status code.~~ The
503 "requested-attributes" operation attribute MAY, but NEED NOT, be returned with the unsupported
504 values.

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

506

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

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

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

510 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
511 ~~and/or values~~ MUST be ignored and ~~an appropriate success code returned (see above)~~'successful-ok-
512 ~~ignored-or-substituted-attributes'~~returned.

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

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

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

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

517 server-error-not-accepting-jobs: not applicable.
518 server-error-busy: same as Print-Job, except the IPP object is too busy to accept even query requests.
519 server-error-job-canceled: not applicable.

520 3.1.3.1.6 Get-Jobs

521 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the Get-
522 Jobs operation with the following specializations and differences. See Section 13 in [[IPP-MODRFC2911](#)]
523 for a more complete description of each status code.

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

525 successful-ok: ~~no request attributes were substituted or ignored (same as Print-Job) and no requested~~
526 ~~attributes were unsupported.~~ same as Get-Printer-Attributes (see section 3.1.3.1.5).
527 successful-ok-ignored-or-substituted-attributes: ~~same as Print-Job, except the "requested-attributes"~~
528 ~~operation attribute MAY, but NEED NOT, be returned with the unsupported values.~~ same as Get-
529 Printer-Attributes (see section 3.1.3.1.5).
530 successful-ok-conflicting-attributes: same as ~~Print-Job~~ Get-Printer-Attributes (see section 3.1.3.1.5).

531

532 For any error status codes, Group 3 is returned containing no attributes or is not returned at all. The following
533 brief error status code descriptions contain unique information for use with Get-Jobs operation. See section
534 14 for the other error status codes that apply uniformly to all operations:

535 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any
536 requests.
537 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.
538 client-error-document-format-not-supported: not applicable.
539 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
540 and/or values MUST be ignored and an appropriate success code returned (see above). ~~successful-~~
541 ~~ok ignored or substituted attributes' returned.~~
542 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.
543 server-error-operation-not-supported: not applicable (since Get-Jobs is REQUIRED).
544 server-error-device-error: same as Print-Job, except that no document data is involved.
545 server-error-temporary-error: same as Print-Job, except that no document data is involved.
546 server-error-not-accepting-jobs: not applicable.
547 server-error-job-canceled: not applicable.

548 3.1.3.1.7 Pause-Printer

549 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Pause-
550 Printer with the following specializations and differences. See Section 13 in [[IPP-MODRFC2911](#)] for a more
551 complete description of each status code.

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

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

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

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

557

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

560 client-error-not-possible: not applicable.

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

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

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

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

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

567 server-error-operation-not-supported: the Pause-Printer operation is not supported.

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

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

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

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

572 **3.1.3.1.8 Resume-Printer**

573 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the specialization's
574 described for Pause-Printer are applicable to Resume-Printer. See Section 13 in [~~IPP-MOD~~[RFC2911](#)] for a
575 more complete description of each status code.

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

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

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

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

580

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

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

583

584 **3.1.3.1.8.1 What about Printers unable to change state due to an error condition?**

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

589 The 'Resume-Printer' operation must clear the 'paused' or 'moving-to-paused' 'printer-state-message'. The
590 operation must return a 'successful-ok' status code.

591 **3.1.3.1.8.2 How is "printer-state" handled on Resume-Printer?**

592

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

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

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

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

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

603 **3.1.3.1.9 Purge-Printer**

604 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the specialization's
605 described for Pause-Printer are applicable to Purge-Printer. See Section 13 in [~~IPP-MOD~~[RFC2911](#)] for a
606 more complete description of each status code.

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

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

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

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

611

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

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

614 3.1.3.2 Job Operations

615 3.1.3.2.1 Send-Document

616 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the Get-
617 Printer-Attributes operation with the following specialization's and differences. See Section 13 in [[IPP-
618 MODRFC2911](#)] for a more complete description of each status code.

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

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

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

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

624

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

627 client-error-not-possible: Same as Print-Job, except that the Printer's "printer-is-accepting-jobs"
628 attribute is not involved, so that the client is able to finish submitting a job that was created with a
629 Create-Job operation after this attribute has been set to 'true'. Another condition is that the state of
630 the job precludes Send-Document, i.e., the job has already been closed out by the client.

631 However, if the IPP Printer closed out the job due to timeout, the 'client-error-timeout' error status
632 SHOULD be returned instead.

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

635

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

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

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

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

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

643 3.1.3.2.2 Send-URI

644 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the specialization's
645 described for Send-Document are applicable to Send-URI. See Section 13 in [[IPP-MODRFC2911](#)] for a
646 more complete description of each status code.

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

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

651

652 3.1.3.2.3 Cancel-Job

653 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Cancel-
654 Job with the following specializations and differences. See Section 13 in [[IPP-MODRFC2911](#)] for a more
655 complete description of each status code.

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

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

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

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

660

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

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

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

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

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

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

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

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

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

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

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

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

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

678 3.1.3.2.4 Get-Job-Attributes

679 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Get-
680 Job-Attributes with the following specializations and differences. See Section 13 in [[IPP-MODRFC2911](#)] for
681 a more complete description of each status code.

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

683 successful-ok: ~~no request attributes were substituted or ignored (same as Print-Job) and no requested~~
684 ~~attributes were unsupported.~~[same as Get-Printer-Attributes \(see section 3.1.3.1.5\).](#)

685 successful-ok-ignored-or-substituted-attributes: same as [Get-Printer-Attributes \(see section](#)
686 [3.1.3.1.5\)](#). ~~Print-Job, except the "requested-attributes" operation attribute MAY, but NEED NOT,~~
687 ~~be returned with the unsupported values.~~
688 successful-ok-conflicting-attributes: same as ~~Print-Job~~ [Get-Printer-Attributes \(see section 3.1.3.1.5\)](#).
689

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

691 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any
692 requests.
693 client-error-document-format-not-supported: not applicable.
694 client-error-attributes-or-values-not-supported: not applicable.
695 client-error-uri-scheme-not-supported: not applicable.
696 [client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes](#)
697 [and/or values MUST be ignored and an appropriate success code returned \(see above\).](#)
698 client-error-conflicting-attributes: not applicable
699 server-error-operation-not-supported: not applicable (since Get-Job-Attributes is REQUIRED).
700 server-error-device-error: same as Print-Job, except no document data is involved.
701 server-error-temporary-error: sane as Print-Job, except no document data is involved..
702 server-error-not-accepting-jobs: not applicable.
703 server-error-job-canceled: not applicable.

704 3.1.3.2.5 Hold-Job

705 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Hold-
706 Job with the following specializations and differences. See Section 13 in [\[IPP-MOÐRFC2911\]](#) for a more
707 complete description of each status code.

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

709 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
710 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
711 successful-ok-conflicting-attributes: same as Print-Job.
712

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

714 client-error-not-possible: The request cannot be carried out because of the state of the Job object
715 ('completed', 'canceled', or 'aborted') or the state of the system.
716 client-error-not-found: the target Printer and/or Job object does not exist.
717 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding address is
718 known.
719 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
720 client-error-document-format-not-supported: not applicable.
721 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-
722 jobs" attribute is not involved.

- 723 server-error-operation-not-supported: the Hold-Job operation is not supported.
724 server-error-device-error: not applicable.
725 server-error-temporary-error: same as Print-Job, except no document data is involved.
726 server-error-not-accepting-jobs: not applicable.
727 server-error-job-canceled: not applicable.

728 3.1.3.2.6 Release-Job

729 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the specialization's
730 described for Hold-Job are applicable to Release-Job. See Section 13 in [[IPP-MODRFC2911](#)] for a more
731 complete description of each status code.

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

733 3.1.3.2.7 Restart-Job

734 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the specialization's
735 described for Hold-Job are applicable to Restart-Job. See Section 13 in [[IPP-MODRFC2911](#)] for a more
736 complete description of each status code.

- 737 server-error-operation-not-supported: the Restart-Job operation is not supported.
738

739 3.1.3.2.7.1 Can documents be added to a restarted job?

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

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

749 3.1.4 Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)

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

756 3.1.5 Sending empty attribute groups

757 The [~~IPP-MOD~~RFC2911] and [~~IPP-PRO~~RFC2910] specifications RECOMMEND that a sender not send
758 an empty attribute group in a request or a response. However, they REQUIRE a receiver to accept an empty
759 attribute group as equivalent to the omission of that group. So a client SHOULD omit the Job Template
760 Attributes group entirely in a create operation that is not supplying any Job Template attributes. Similarly, an
761 IPP object SHOULD omit an empty Unsupported Attributes group if there are no unsupported attributes to
762 be returned in a response.

763 The [~~IPP-PRO~~RFC2910] specification REQUIRES a receiver to be able to receive either an empty attribute
764 group or an omitted attribute group and treat them equivalently. The term "receiver" means an IPP object for
765 a request and a client for a response. The term "sender" means a client for a request and an IPP object for a
766 response.

767 There is an exception to the rule for Get-Jobs when there are no attributes to be returned. [~~IPP-~~
768 ~~PRO~~RFC2910] contains the following paragraph:

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

774 3.2 Printer Operations

775 3.2.1 Print-Job operation

776 3.2.1.1 Flow controlling the data portion of a Print-Job request (Issue 1.22)

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

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

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

784 3.2.1.2 Returning job-state in Print-Job response (Issue 1.30)

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

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

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

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

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

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

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

807 **3.2.2 Get-Printer-Attributes operation**

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

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

812 **3.2.3 Get-Jobs operation**

813 **3.2.3.1 Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?**

814 In [\[IPP-MODRFC2911\]](#) section 3.2.6.1 'Get-Jobs Request', if the attribute 'my-jobs' is present and set to
815 TRUE, MUST the 'requesting-user-name' attribute be there too, and if it's not present what should the IPP
816 printer do?

817 [\[IPP-MODRFC2911\]](#) Section 8.3 describes the various cases of "requesting-user-name" being present or not
818 for any operation. If the client does not supply a value for "requesting-user-name", the printer MUST assume
819 that the client is supplying some anonymous name, such as "anonymous".

820 3.2.3.2 Why is there a "limit" attribute in the Get-Jobs operation?

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

829 3.2.4 Create-Job operation

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

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

839 It depends on implementation. Some servers spool the entire job, including all document data, before starting
840 to process, so such an implementation would wait for the "last-document" before starting to process the job. If
841 the time-out occurs without the "last-document", then the server takes one of the indicated actions in section
842 3.3.1 in the [\[IPP-MODRFC2911\]](#) document. Other servers will start to process document data as soon as
843 they have some. These are the so-called "non-spooling" printers. Currently, there isn't a way for a client to
844 determine whether the Printer will spool all the data or will start to process (and print) as soon as it has some
845 data.

846 3.3 Job Operations

847 3.3.1 Validate-Job

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

852 3.3.2 Restart-Job

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

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

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

862 4 Object Attributes

863 4.1 Attribute Syntax's

864 4.1.1 The 'none' value for empty sets (Issue 1.37)

865 [[IPP-MODRFC2911](#)] states that the 'none' value should be used as the value of a 1setOf when the set is
866 empty. In most cases, sets that are potentially empty contain keywords so the keyword 'none' is used, but for
867 the 3 finishings attributes, the values are enums and thus the empty set is represented by the enum 3. Currently
868 there are no other attributes with 1setOf values, which can be empty and can contain values that are not
869 keywords. This exception requires special code and is a potential place for bugs. It would have been better if
870 we had chosen an out-of-band value, either "no-value" or some new value, such as 'none'. Since we didn't,
871 implementations have to deal with the different representations of 'none', depending on the attribute syntax.

872 4.1.2 Multi-valued attributes (Issue 1.31)

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

879 **4.1.3 Case Sensitivity in URIs (issue 1.6)**

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

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

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

893 Example of equivalent URI's

894 http://abc.com:80/~smith/home.html

895 http://ABC.com/%7Esmith/home.html

896 http:/ABC.com:/%7esmith/home.html

897 Example of equivalent URI's using the IPP scheme

898 ipp://abc.com:631/~smith/home.html

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

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

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

902 **4.1.4 Maximum length for xxxWithLanguage and xxxWithoutLanguage**

9 0 3 The 'textWithLanguage' and 'nameWithLanguage' are compound syntaxes that have two components. The
9 0 4 first component is the 'language' component that can contain up to 63 octets. The second component is the
9 0 5 'text' or 'name' component. The maximum length of these are 1023 octets and 255 octets respectively. The
9 0 6 definition of attributes with either syntax may further restrict the length. (e.g. printer-name (name(127)))

9 0 7 The length of the 'language' component has no effect on the allowable length of 'text' in 'textWithLanguage' or
9 0 8 the length of 'name' in 'nameWithLanguage'

9 0 9 **4.2 Job Template Attributes**

9 1 0 **4.2.1 multiple-document-handling(type2 keyword)**

9 1 1 **4.2.1.1 Support of multiple document jobs**

9 1 2 IPP/1.0 is silent on which of the four effects an implementation would perform if it supports Create-Job, but
9 1 3 does not support "multiple-document-handling" or multiple documents per job. IPP/1.1 was changed so that a
9 1 4 Printer could support Create-Job without having to support multiple document jobs. The "multiple-document-
9 1 5 jobs-supported" (boolean) Printer description attribute was added to IPP/1.1 along with the 'server-error-
9 1 6 multiple-document-jobs-not-supported' status code for a Printer to indicate whether or not it supports multiple
9 1 7 document jobs, when it supports the Create-Job operation. Also IPP/1.1 was clarified that the Printer **MUST**
9 1 8 support the "multiple-document-handling" (type2 keyword) Job Template attribute with at least one value if the
9 1 9 Printer supports multiple documents per job.

9 2 0 **4.3 Job Description Attributes**

9 2 1 **4.3.1 Getting the date and time of day**

9 2 2 The "~~date-time-at-creation~~", "~~date-time-at-processing~~", and "~~date-time-at-completed~~" attributes ~~may be are~~
9 2 3 returned ~~in integer time ticks or absolute~~ as dateTime syntax. ~~These attributes are OPTIONAL for a Printer~~
9 2 4 ~~to support. However, there~~ are various ways for a Printer to get the date and time of day. Some
9 2 5 suggestions:

9 2 6 1. A Printer can get time from an NTP timeserver if there's one reachable on the network . See RFC
9 2 7 1305. Also DHCP option 32 in RFC 2132 returns the IP address of the NTP server.

9 2 8 2. Get the date and time at startup from a human operator

9 2 9 3. Have an operator set the date and time using a web administrative interface

9 3 0 4. Get the date and time from incoming HTTP requests, though the problems of spoofing need to be
9 3 1 considered. Perhaps comparing several HTTP requests could reduce the chances of spoofing.

932 5. Internal date time clock battery driven.

933 6. Query "<http://tycho.usno.navy.mil/cgi-bin/timer.pl>"

934 4.4 Printer Description Attributes

935 4.4.1 **queued-job-count** (integer(0:MAX))

936 4.4.1.1 Why is "queued-job-count" RECOMMENDED (Issue 1.14)?

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

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

943 4.4.1.2 Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?

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

948 4.4.2 **printer-current-time** (dateTime)

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

951 1. an internal date time clock,

952 2. from the operator at startup using the console,

953 3. from an operator using an administrative web page,

954 4. from HTTP headers supplied in client requests,

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

956 6. from the network, using NTP [RFC1305] or DHCP option 32 [RFC2132] that returns the IP
957 address of the NTP server.

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

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

963 4.4.3 Printer-uri

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

965 A forgiving printer implementation would not reject the operation. But the implementation has its rights to
966 reject a printer or job operation if the operational attribute printer-uri is not a value of the printer-uri-
967 supported. The printer may-might not be improperly configured. The request obviously reached the printer.
968 The printer could treat the printer-uri as the logical equivalent of a value in the printer-uri-supported. It would
969 be implementation dependent for which value, and associated security policy, would apply. This does also
970 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
971 compare URI's.

972 4.5 Empty Jobs

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

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

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

984 5 Directory Considerations

985 5.1 General Directory Schema Considerations

986 The [~~ipp-mod~~[RFC2911](#)] document lists RECOMMENDED and OPTIONAL Printer object attributes for
987 directory schemas. See [~~ipp-mod~~[RFC2911](#)] APPENDIX E: Generic Directory Schema.

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

995 5.2 IPP Printer with a DNS name

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

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

999 6 Security Considerations

000 This section corresponds to the [IPP-MODRFC2911](#) Section 8 "Security Considerations."

001 6.1 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)

002 The following clarification was added to [[IPP-MODRFC2911](#)] section 8.5:

003 8.5 Queries on jobs submitted using non-IPP protocols

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

010 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such
011 "foreign jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-Attributes
012 and Cancel-Job. Such an implementation also needs to deal with the problem of authentication of such
013 foreign jobs. One approach would be to treat all such foreign jobs as belonging to users other than the
014 user of the IPP client. Another approach would be for the foreign job to belong to 'anonymous'. Only if
015 the IPP client has been authenticated as an operator or administrator of the IPP Printer object, could the
016 foreign jobs be queried by an IPP request. Alternatively, if the security policy were to allow users to query
017 other users' jobs, then the foreign jobs would also be visible to an end-user IPP client using Get-Jobs and
018 Get-Job-Attributes.

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

022 7 Encoding and Transport

023 This section discusses various aspects of IPP/1.1 Encoding and Transport [[IPP-PRO](#)[RFC2910](#)].

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

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

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

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

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

- 048 - the "header" column contains the name of a header
- 049 - the "request/client" column indicates whether a client sends the header.
- 050 - the "request/ server" column indicates whether a server supports the header when received.
- 051 - the "response/ server" column indicates whether a server sends the header.
- 052 - the "response /client" column indicates whether a client supports the header when received.

053 - the "values and conditions" column specifies the allowed header values and the conditions for the
 054 header to be present in a request/response.

055

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

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

- 059 - **must:** the client or server MUST send the header,
- 060 - **must-if:** the client or server MUST send the header when the condition described in the "values and
 061 conditions" column is met,
- 062 - **may:** the client or server MAY send the header
- 063 - **not:** the client or server SHOULD NOT send the header. It is not relevant to an IPP implementation.
 064

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

- 066 - **must:** the client or server MUST support the header,
- 067 - **may:** the client or server MAY support the header
- 068 - **not:** the client or server SHOULD NOT support the header. It is not relevant to an IPP
 069 implementation.

070 7.1 General Headers

071 The following is a table for the general headers.

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

072 7.2 Request Headers

073 The following is a table for the request headers.

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

074 7.3 Response Headers

075 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.

076 7.4 Entity Headers

077 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	

078 7.5 Optional support for HTTP/1.0

079 IPP implementations consist of an HTTP layer and an IPP layer. In the following discussion, the term "client"
 080 refers to the HTTP client layer and the term "server" refers to the HTTP server layer. The Encoding and
 081 Transport document [[IPP-PRO RFC2910](#)] requires that HTTP 1.1 MUST be supported by all clients and all
 082 servers. However, a client and/or a server implementation may choose to also support HTTP 1.0.

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

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

089 7.6 HTTP/1.1 Chunking

090 7.6.1 Disabling IPP Server Response Chunking

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

095 TE: identity

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

098 7.6.2 Warning About the Support of Chunked Requests

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

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

107 Because of this lack of conformance of HTTP servers to the HTTP/1.1 standard, the IPP standard [~~IPP-~~
108 ~~PRO~~RFC2910] REQUIRES that a conforming IPP Printer object implementation support chunked requests
109 and that conforming clients accept chunked responses. Therefore, IPP object implementers are warned to
110 seek HTTP server implementations that support chunked POST requests in order to conform to the IPP
111 standard and/or use implementation techniques that support chunked POST requests.

112 8 References

113 [CGI]

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

115 [~~IPP-MOD~~]

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

118 [~~IPP-PRO~~]

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

121 [ldap-printer]

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

124 [RFC793]

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

126 [RFC1123]

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

- 128 [RFC2026]
129 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 130 [RFC2119]
131 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
- 132 [RFC2396]
133 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC
134 2396, August 1998.
- 135 [RFC2565]
136 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
137 Semantics", RFC 2566, April 1999.
- 138 [RFC2566]
139 Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.0: Encoding and Transport",
140 RFC 2565, April 1999.
- 141 [RFC2567]
142 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-03.txt, November, 1998.
- 143 [RFC2568]
144 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol", RFC
145 2568, April 1999.
- 146 [RFC2569]
147 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC 2569,
148 April 1999.
- 149 [RFC2616]
150 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer
151 Protocol - HTTP/1.1", RFC 2616, June 1999.
- 152 [RFC2910]
153 Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.0: Encoding and Transport",
154 RFC 2910, September, 2000.
- 155 [RFC2911]
156 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
157 Semantics", RFC 2911, September, 2000.
- 158 [Servlet]
159 Servlet Specification Version 2.1 (<http://java.sun.com/products/servlet/2.1/index.html>).
- 160 [svrloc-printer]

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

163 [SSL]

164 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.

165 **9 Authors' Address**

166
167 Thomas N. Hastings
168 Xerox Corporation
169 701 Aviation Blvd.
170 El Segundo, CA 90245
171 hastings@cp10.es.xerox.com
172

173 Carl-Uno Manros
174 Xerox Corporation
175 701 Aviation Blvd.
176 El Segundo, CA 90245
177 manros@cp10.es.xerox.com
178

179 Carl Kugler
180 Mail Stop 003G
181 IBM Printing Systems Co
182 6300 Diagonal Hwy
183 Boulder CO 80301
184 Kugler@us.ibm.com
185

186 Henrik Holst
187 i-data Printing Systems
188 Vadstrupvej 35-43
189 2880 Bagsvaerd, Denmark
190 hh@i-data.com
191

192 Peter Zehler
193 Xerox Corporation
194 800 Philips Road
195 Webster, NY 14580
196
197 [Phone: 716-265-8755](tel:716-265-8755) peter.zehler@usa.xerox.com

198 **10 Full Copyright Statement Notices**

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

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

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

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

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

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

226 Acknowledgement

227 ~~Funding for the RFC Editor function is currently provided by the Internet Society. Change History (to be~~
228 ~~removed at time of RFC publishing)~~

229 ~~The change history is in reverse chronological order:~~

230 11.1 Changes from 000509 to 000530

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

232 ~~1. Added section 5.1 on General Directory Considerations which includes references to SLP and LDAP Printer~~
233 ~~schemas and their introduction of the "printer-xri-supported" attribute which combines "printer-uri-supported",~~
234 ~~"uri-security-supported", and "uri-authentication-supported" attributes.~~

235 ~~11.2 Changes from 990927 to 000509~~

236 ~~The following changes were made to the 9/27/99 version to make the 5/09/00 version:~~

237 ~~1. Table 5—Corrected some attributes returned by Send-Document and Send-URI to be the same as Print-Job as~~
238 ~~in [ipp-mod].~~

239 ~~2. Corrected several uses of 'client-error-bad-syntax' to be 'client-error-bad-request' as in the [ipp-mod].~~

240 ~~3. Added section 3.1.3.1.8.1 to clarify what Resume-Printer does if the Printer is unable to resume the output~~
241 ~~device and section 3.1.3.1.8.2 about the "printer-state" for such a condition.~~

242 ~~4. Added section 3.3.2 to indicate that on a Restart-Job that a Printer MUST re-fetch the document data when the~~
243 ~~job was created with Print-URI or Send-URI.~~

244 ~~5. Section 4.1.4—clarified that the length field for 'textWithLanguage' and 'nameWithLanguage' does *not* include the~~
245 ~~language field, so that the same maximum length of the data applies to the WithLanguage as the~~
246 ~~WithoutLanguage types, not counting the language field.~~

247 ~~6. Added section 4.5 about empty jobs, i.e., with no documents. They are processed as any other job, possibly~~
248 ~~producing start and/or end sheets.~~

249 ~~11.3 Changes from 990914 to 990927~~

250 ~~1. Add comments about this document is also IPP/1.0 relevant.~~

251 ~~2. Section 4.1.3: Add more examples of URI's with the port 631 and the ipp scheme.~~

252 ~~3. Section 4.4.3: Move the DNS stuff to the 'how to compare URI's.~~

253 ~~4. Section 4.4.3.2: Swap lines, first tell about the forgiven printer and then what the printer is allowed to do.~~

254 ~~5. Fixed some errors in the Summary Attribute tables 1-5 and broke them into five portrait tables, so that it can be~~
255 ~~made into plain text for INTERNET-DRAFTS.~~

256 ~~11.4 Changes from 990726 to 990914:~~

257 ~~1. Added IPP/1.1 operations and attributes to table 1.~~

258 ~~2. Validate version: Added text and table from issue 32~~

- 259 ~~3.Printer uri supported: Added section 4.4.4~~
- 260 ~~4.Added IPP/1.1 operations to section 3.1.2.1.4.3~~
- 261 ~~5.Added answer to question "Should the server wait for the "last document" operation attribute set to 'true' before~~
262 ~~starting to "process" the job?" in section 3.2.4~~
- 263 ~~6.Changed 'server error uri scheme not supported' to 'client error uri scheme not supported' in section 3.1.2.1.5~~
264 ~~when talking about the 'document uri' attribute.~~
- 265 ~~7.Added 'Suggested Operation Processing Steps' and 'Suggested Additional Processing Steps for Operations that~~
266 ~~Create/Validate Jobs and Add Document' flow chart overview.~~

267 **~~11.5Changes to produce the February 12, 1999 version from the January 8, 1999 version:~~**

- 268 ~~1.Section 2.2.1.5: added check for document not found or accessible in Print URI and Send URI~~
- 269 ~~2.Section 3.6.2: Clarified that the IPP standard requires that servers MUST accept chunked requests and that~~
270 ~~clients MUST accept chunked responses, in spite of the lack of conformance of HTTP servers to the~~
271 ~~HTTP/1.1 requirement to support chunking.~~

272 **~~11.6Changes to produce the January 8, 1999 version from the December 6, 1998 version:~~**

- 273 ~~1.Added section 3.6.2: Warning About the Use of Chunked Requests with CGI Script Implementations~~
- 274 ~~2.Section 2.2.1.2: changed "printer operations supported" to "operations supported".~~
- 275 ~~3.Section 2.2.1.6: changed "job media supported" to "job media sheets supported"~~
- 276 ~~4.Section 2.2.3: separated the validation checks for variable length attributes into two separate tests: one for~~
277 ~~correct attribute syntax and one for correct length.~~
- 278 ~~5.Section 2.2.3: changed "multiple document handling supported" to "printer resolution supported"~~
- 279 ~~6.Section 2.6.1: recommended that an IPP object also support US ASCII charset.~~
- 280 ~~7.Section 3: Clarified that a server is not required to send a response until after it has received the client's~~
281 ~~entire request, but recommend that the server return a response as soon as possible if an error is detected~~
282 ~~while the client is still sending the data, rather than waiting until all of the data is received. Also~~
283 ~~recommended that a client listen for an error response that an IPP server MAY send before it receives all~~
284 ~~the data.~~

285 **~~11.7Changes to produce the December 6, 1998 version from the November 16, 1998 version:~~**

286 ~~Included all of the remaining agreed issues raised before the November 16, 1998 production of the Internet Drafts~~
287 ~~for IPP/1.0 that included adding explanations to the Implementers Guide.~~

288 ~~Changes from 990422 to 990726:~~

289 ~~1. Encoding and Transport: Address issues 4, 5, 20 from Issues raised at Bake-Off2.doc~~

290 ~~2. Decide whether to accept or reject the request: discuss issues 6, 9, 10~~

291 ~~3. Get Printer Attributes: add notes about printer make and model and .INF files; issue 7~~

292 ~~4. Create Job: clarify job incoming vs. data insufficient; issue 13~~

293 ~~5. Get Printer Attributes: polling — issue 16~~

294 ~~6. Job Description Attributes: ways to get time; issue 17~~

295 ~~7. Validate the values of the Job Template Attributes: clarify zero length keywords; issue 22~~

296 ~~8. Validate Optional Operation Attributes: Note about checking for compression in IPP/1.0; issue 28~~

297 ~~9. Validate version number: advantages to backward compatibility; issue 33~~

298 ~~10. Note: examples for issue 2 seem to be covered sufficiently in the new MOD doc.~~