

1 Network Working Group
2 Request for Comments: 3196
3 Obsoletes RFC 2639
4 Category: Informational

T. Hastings
C. Manros
P. Zehler
Xerox Corporation
C. Kugler
IBM Printing Systems Co
H. Holst
i-data Printing Systems
November 2001

11 Internet Printing Protocol/1.1: Implementer's Guide

12 Status of this Memo

13 This memo provides information for the Internet community. It does not specify an Internet standard of
14 any kind. Distribution of this memo is unlimited.

15 Copyright Notice

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

17 Abstract

18 This document is one of a set of documents, which together describe all aspects of a new Internet
19 Printing Protocol (IPP). IPP is an application level protocol that can be used for distributed printing
20 using Internet tools and technologies. This document contains information that supplements the IPP
21 Model and Semantics [RFC2911] and the IPP Transport and Encoding [RFC2910] documents. It is
22 intended to help implementers understand IPP/1.1, as well as IPP/1.0 [RFC2565, RFC2566], and some
23 of the considerations that may assist them in the design of their client and/or IPP object
24 implementations. For example, a typical order of processing requests is given, including error checking.
25 Motivation for some of the specification decisions is also included.

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

27

27 **TABLE OF CONTENTS**

28	1	Introduction.....	5
29	1.1	Conformance language.....	5
30	1.2	Other terminology.....	6
31	1.3	Issues Raised from Interoperability Testing Events.....	6
32	2	IPP Objects.....	6
33	3	IPP Operations.....	8
34	3.1	Common Semantics.....	8
35	3.1.1	Summary of Operation Attributes	9
36	3.1.2	Suggested Operation Processing Steps for IPP Objects	14
37	3.1.2.1	Suggested Operation Processing Steps for all Operations	15
38	3.1.2.1.1	Validate version number.....	16
39	3.1.2.1.2	Validate operation identifier	16
40	3.1.2.1.3	Validate the request identifier.....	17
41	3.1.2.1.4	Validate attribute group and attribute presence and order	17
42	3.1.2.1.4.1	Validate the presence and order of attribute groups.....	17
43	3.1.2.1.4.2	Ignore unknown attribute groups in the expected position.....	17
44	3.1.2.1.4.3	Validate the presence of a single occurrence of required Operation attributes	18
45	3.1.2.1.5	Validate the values of the REQUIRED Operation attributes	24
46	3.1.2.1.6	Validate the values of the OPTIONAL Operation attributes	28
47	3.1.2.2	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add Documents	31
49	3.1.2.2.1	Default "ipp-attribute-fidelity" if not supplied.....	31
50	3.1.2.2.2	Check that the Printer object is accepting jobs.....	32
51	3.1.2.2.3	Validate the values of the Job Template attributes	32
52	3.1.2.3	Algorithm for job validation.....	32
53	3.1.2.3.1	Check for conflicting Job Template attributes values	37
54	3.1.2.3.2	Decide whether to REJECT the request	37
55	3.1.2.3.3	For the Validate-Job operation, RETURN one of the success status codes	39
56	3.1.2.3.4	Create the Job object with attributes to support.....	39
57	3.1.2.3.5	Return one of the success status codes	40
58	3.1.2.3.6	Accept appended Document Content	41
59	3.1.2.3.7	Scheduling and Starting to Process the Job.....	41
60	3.1.2.3.8	Completing the Job	41
61	3.1.2.3.9	Destroying the Job after completion	41
62	3.1.2.3.10	Interaction with "ipp-attribute-fidelity".....	42
63	3.1.2.3.11	Character set code conversion support	42
64	3.1.2.3.12	What charset to return when an unsupported charset is requested (Issue 1.19)?.....	43
65	3.1.2.3.13	Natural Language Override (NLO).....	43
66	3.1.3	Status codes returned by operation	44
67	3.1.3.1	Printer Operations.....	44
68	3.1.3.1.1	Print-Job.....	45
69	3.1.3.1.2	Print-URI.....	46
70	3.1.3.1.3	Validate-Job.....	47
71	3.1.3.1.4	Create-Job	47
72	3.1.3.1.5	Get-Printer-Attributes	47
73	3.1.3.1.6	Get-Jobs	48
74	3.1.3.1.7	Pause-Printer	48
75	3.1.3.1.8	Resume-Printer	49
76	3.1.3.1.8.1	What about Printers unable to change state due to an error condition?	49

77	3.1.3.1.8.2	How is "printer-state" handled on Resume-Printer?.....	50
78	3.1.3.1.9	Purge-Printer	50
79	3.1.3.2	Job Operations.....	50
80	3.1.3.2.1	Send-Document	50
81	3.1.3.2.2	Send-URI	51
82	3.1.3.2.3	Cancel-Job.....	51
83	3.1.3.2.4	Get-Job-Attributes	52
84	3.1.3.2.5	Hold-Job.....	53
85	3.1.3.2.6	Release-Job.....	53
86	3.1.3.2.7	Restart-Job	53
87	3.1.3.2.7.1	Can documents be added to a restarted job?.....	54
88	3.1.4	Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)	54
89	3.1.5	Sending empty attribute groups.....	54
90	3.2	Printer Operations	55
91	3.2.1	Print-Job operation	55
92	3.2.1.1	Flow controlling the data portion of a Print-Job request (Issue 1.22).....	55
93	3.2.1.2	Returning job-state in Print-Job response (Issue 1.30)	55
94	3.2.2	Get-Printer-Attributes operation	56
95	3.2.3	Get-Jobs operation.....	56
96	3.2.3.1	Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?.....	56
97	3.2.3.2	Why is there a "limit" attribute in the Get-Jobs operation?.....	56
98	3.2.4	Create-Job operation.....	57
99	3.3	Job Operations	57
100	3.3.1	Validate-Job	57
101	3.3.2	Restart-Job	57
102	4	Object Attributes.....	58
103	4.1	Attribute Syntax's.....	58
104	4.1.1	The 'none' value for empty sets (Issue 1.37)	58
105	4.1.2	Multi-valued attributes (Issue 1.31).....	58
106	4.1.3	Case Sensitivity in URIs (issue 1.6).....	58
107	4.1.4	Maximum length for xxxWithLanguage and xxxWithoutLanguage	59
108	4.2	Job Template Attributes	59
109	4.2.1	multiple-document-handling(type2 keyword)	59
110	4.2.1.1	Support of multiple document jobs.....	59
111	4.3	Job Description Attributes.....	60
112	4.3.1	Getting the date and time of day.....	60
113	4.4	Printer Description Attributes.....	60
114	4.4.1	queued-job-count (integer(0:MAX))	60
115	4.4.1.1	Why is "queued-job-count" RECOMMENDED (Issue 1.14)?.....	60
116	4.4.1.2	Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?.....	60
117	4.4.2	printer-current-time (dateTime).....	61
118	4.4.3	Printer-uri.....	61
119	4.5	Empty Jobs	61
120	5	Directory Considerations.....	62
121	5.1	General Directory Schema Considerations	62
122	5.2	IPP Printer with a DNS name.....	62
123	6	Security Considerations.....	62
124	6.1	Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)	62
125			62

126	7	Encoding and Transport	63
127	7.1	General Headers.....	64
128	7.2	Request Headers	65
129	7.3	Response Headers.....	66
130	7.4	Entity Headers	67
131	7.5	Optional support for HTTP/1.0.....	67
132	7.6	HTTP/1.1 Chunking.....	68
133	7.6.1	Disabling IPP Server Response Chunking	68
134	7.6.2	Warning About the Support of Chunked Requests	68
135	8	References	68
136	9	Authors' Address.....	70
137	10	Description of the Base IPP Documents	73
138	11	Full Copyright Statement	74
139			
140		TABLES	
141			
142		Table 1 - Summary of Printer operation attributes that sender MUST supply	9
143		Table 2 - Summary of Printer operation attributes that sender MAY supply.....	10
144		Table 3 - Summary of Job operation attributes that sender MUST supply.....	11
145		Table 4 - Summary of Job operation attributes that sender MAY supply.....	12
146		Table 5 - Printer operation response attributes.....	13
147		Table 6 - Examples of validating IPP version.....	16
148		Table 7 - Rules for validating single values X against Z.....	33
149			
150			

150
151
152

153 **1 Introduction**

154 The IPP Implementer's Guide (IIG) (this document) contains information that supplements the IPP
155 Model and Semantics [RFC2911] and the IPP Transport and Encoding [RFC2910] documents. This
156 document is just one of a suite of documents that fully define IPP. The base set of IPP documents
157 includes:

158 Design Goals for an Internet Printing Protocol [RFC2567]
159 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
160 Internet Printing Protocol/1.1: Model and Semantics [RFC2911]
161 Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
162 Internet Printing Protocol/1.1: Implementer's Guide (this document)
163 Mapping between LPD and IPP Protocols [RFC2569]

164
165
166

See section 10 for a description of these base IPP documents. Anyone reading these documents for the first time is strongly encouraged to read the IPP documents in the above order.

167 As such the information in this document is not part of the formal specification of IPP/1.1. Instead
168 information is presented to help implementers understand IPP/1.1, as well as IPP/1.0 [RFC2565,
169 RFC2566], including some of the motivation for decisions taken by the committee in developing the
170 specification. Some of the implementation considerations are intended to help implementers design their
171 client and/or IPP object implementations. If there are any contradictions between this document and
172 [RFC2911] or [RFC2910], those documents take precedence over this document.

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

174 Note: In order to help the reader of the IIG and the IPP Model and Semantics document, the sections
175 in this document parallel the corresponding sections in the Model document and are numbered the same
176 for ease of cross reference. The sections that correspond to the IPP Transport and Encoding are
177 correspondingly offset.

178 **1.1 Conformance language**

179 Usually, this document does not contain the terminology MUST, MUST NOT, MAY, NEED NOT,
180 SHOULD, SHOULD NOT, REQUIRED, and OPTIONAL. However, when those terms do appear in
181 this document, their intent is to repeat what the [RFC2911] and [RFC2910] documents require and
182 allow, rather than specifying additional conformance requirements. These terms are defined in section
183 12 on conformance terminology in [RFC2911], most of which is taken from RFC 2119 [RFC2119].

184 Implementers should read section 12 (APPENDIX A) in [RFC2911] in order to understand these
185 capitalized words. The words MUST, MUST NOT, and REQUIRED indicate what implementations
186 are required to support in a client or IPP object in order to be conformant to [RFC2911] and
187 [RFC2910]. MAY, NEED NOT, and OPTIONAL indicate was is merely allowed as an implementer
188 option. The verbs SHOULD and SHOULD NOT indicate suggested behavior, but which is not
189 required or disallowed, respectively, in order to conform to the specification.

190 **1.2 Other terminology**

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

195 **1.3 Issues Raised from Interoperability Testing Events**

196 The IPP WG has conducted three open Interoperability Testing Events. The first one was held in
197 September 1998, the second one was held in March 1999, and the third one was held in October 2000.
198 See the summary reports in:

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

200 The issues raised from the first Interoperability Testing Event are numbered 1.n in this document and
201 have been incorporated into "IPP/1.0 Model and Semantics" [RFC2566] and the "IPP/1.0 Encoding and
202 Transport" [RFC2565] documents. However, some of the discussion is left here in the Implementer's
203 Guide to help understanding.

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

208 The issues raised from the third Interoperability Testing Event are numbered 3.n in this document and
209 are described in:

210 <ftp://ftp.pwg.org/pub/pwg/ipp/Issues/Issues-raised-at-Bake-Off3.pdf>
211 <ftp://ftp.pwg.org/pub/pwg/ipp/Issues/Issues-raised-at-Bake-Off3.doc>
212 <ftp://ftp.pwg.org/pub/pwg/ipp/Issues/Issues-raised-at-Bake-Off3.txt>

213 **2 IPP Objects**

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

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

221

221 **3 IPP Operations**

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

224 **3.1 Common Semantics**

225 This section discusses semantics common to all operations.

226 **3.1.1 Summary of Operation Attributes**227 **Table 1 - Summary of Printer operation attributes that sender MUST supply**

Operation Attributes	Printer Operations						
	Requests						Response s
	PJ, VJ (R)	PU (O)	CJ (O)	GPA (R)	GJ (R)	PP, RP, PP (O+)	All Operatio ns
Operation parameters--REQUIRED to be supplied by the sender:							
operation-id	R	R	R	R	R	R	
status-code							R
request-id	R	R	R	R	R	R	R
version-number	R	R	R	R	R	R	R
Operation attributes--REQUIRED to be supplied by the sender:							
attributes-charset	R	R	R	R	R	R	R
attributes-natural- language	R	R	R	R	R	R	R
document-uri		R					
job-id*							
job-uri*							
last-document							
printer-uri	R	R	R	R	R	R	
Operation attributes--RECOMMENDED to be supplied by the sender:							
job-name	R	R	R				
requesting-user-name	R	R	R	R	R	R	

228

229

Legend:

230

PJ, VJ: Print-Job, Validate-Job

231

PU: Print-URI

232

CJ: Create-Job

233

GPA: Get-Printer-Attributes

234

GJ: Get-Jobs

235

PP, RP, PP: Pause-Printer, Resume-Printer, Purge-Printer

236

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

237

238

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

239

240

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

241

242

243

244

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

Operation Attributes	Printer Operations							Responses
	Requests							
	PJ, (R)	VJ	PU (O)	CJ (O)	GPA (R)	GJ (R)	PP, RP, PP (O+)	All Operati ons
Operation attributes--OPTIONAL to be supplied by the sender:								
status-message								O
detailed-status-message								O
document-access-error								O**
compression	R		R					
document-format	R		R		R			
document-name	O		O					
document-natural-language	O		O					
ipp-attribute-fidelity	R		R	R				
job-impressions	O		O	O				
job-k-octets	O		O	O				
job-media-sheets	O		O	O				
limit						R		
message								
my-jobs						R		
requested-attributes					R	R		
which-jobs						R		

245

Legend:

246

PJ, VJ: Print-Job, Validate-Job

247

PU: Print-URI

248

CJ: Create-Job

249

GPA: Get-Printer-Attributes

250

GJ: Get-Jobs

251

PP, RP, PP: Pause-Printer, Resume-Printer, Purge-Printer

252

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

254

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

255

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

258

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

259

** "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
	SD (O)	SU (O)	CJ (R)	GJA (R)	HJ, RJ, RJ (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

Legend:

264

SD: Send-Document

265

SU: Send-URI

266

CJ: Cancel-Job

267

GJA: Get-Job-Attributes

268

HJ, RJ, RJ: Hold-Job, Release-Job, Restart-Job

269

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

271

272

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

273

274

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

275

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

276

277

278

279

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

Operation Attributes	Job Operations						
	Requests						Responses
	SD (O)	SU (O)	CJ (R)	GJA (R)	HJ, RJ, RJ (O+)	SD (O)	All Operations
Operation attributes--OPTIONAL to be supplied by the sender:							
status-message							O
detailed-status-message							O
document-access-error							O**
compression	R	R					
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							

Legend:

- 280 SD: Send-Document
- 281 SU: Send-URI
- 282 CJ: Cancel-Job
- 283 GJA: Get-Job-Attributes
- 284 HJ, RJ, RJ: Hold-Job, Release-Job, Restart-Job
- 285 R indicates a REQUIRED operation that MUST be supported by the IPP object (Printer or Job). For
- 286 attributes, R indicates that the attribute MUST be supported by the IPP object that supports the
- 287 associated operation.
- 288 O indicates an OPTIONAL operation or attribute that MAY be supported by the IPP object (Printer
- 289 or Job).
- 290 + indicates that this is not an IPP/1.0 feature, but is only a part of IPP/1.1 and future versions of IPP.
- 291 * "job-id" is REQUIRED only if used together with "printer-uri" to identify the target job; otherwise,
- 292 "job-uri" is REQUIRED.
- 293 ** "document-access-error" applies to the Send-URI operation only

294

294

295

Table 5 - Printer operation response attributes

Operation Attributes	Printer Operations						
	Response						
	PJ (R) SD (O)	VJ (R)	PU (O) SU (O)	CJ (O)	GPA (R)	GJ (R)	PP, RP, PP (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				

296

297

Legend:

298

PJ, SJ: Print-Job, Send-Document

299

VJ: Validate-Job

300

PU, SU: Print-URI, Send-URI

301

CJ: Create-Job

302

GPA: Get-Printer-Attributes

303

GJ: Get-Jobs

304

PP, RP, PP: Pause-Printer, Resume-Printer, Purge-Printer

305

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

306

307

308

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

309

310

311

311

3.1.2 Suggested Operation Processing Steps for IPP Objects

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

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

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

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

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

340

340 **3.1.2.1 Suggested Operation Processing Steps for all Operations**

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

344	IIG Sect #	Flow	IPP error status codes
345	-----	----	-----
346			
347		v	err
348	3.1.2.1.1	<Validate version>	--> server-error-version-not-
349			supported
350		ok	
351		v	err
352	3.1.2.1.2	<Validate operation>	--> server-error-operation-not-
353			supported
354		ok	
355		v	err
356	3.1.2.1.4.1-	<Validate presence>	--> client-error-bad-request
357	3.1.2.1.4.2	<of attributes>	
358		ok	
359		v	err
360	3.1.2.1.4.3	<Validate presence>	--> client-error-bad-request
361		<of operation attr>	
362		ok	
363		v	err
364	3.1.2.1.5	<Validate values of>	--> client-error-bad-request
365		<operation attrs>	client-error-request-value-
366			too-long
367		<(length, tag, range,>	
368		<multi-value)>	
369		ok	
370		v	err
371	3.1.2.1.5	<Validate values>	--> client-error-bad-request
372		<with supported values>	client-error-charset-not-
373			supported
374		ok	client-error-attributes-or-
375			values-
376			not-supported
377		v	err
378	3.1.2.1.6	<Validate optionally>	--> client-error-bad-request
379		<operation attr>	client-error-natural-language-
380			not-supported
381			client-error-request-value-
382			too-long
383			client-error-attributes-or-
384			values-not-supported
385			

386 3.1.2.1.1 Validate version number

387 Every request and every response contains the "version-number" attribute. The value of this attribute is
 388 the major and minor version number of the syntax and semantics that the client and IPP object is using,
 389 respectively. The "version-number" attribute remains in a fixed position across all future versions so
 390 that all clients and IPP object that support future versions can determine which version is being used.
 391 The IPP object checks to see if the major version number supplied in the request is supported. If not,
 392 the Printer object REJECTS the request and RETURNS the 'server-error-version-not-supported' status
 393 code in the response. The IPP object returns in the "version-number" response attribute the major and
 394 minor version for the error response. Thus the client can learn at least one major and minor version that
 395 the IPP object supports. The IPP object is encouraged to return the closest version number to the one
 396 supplied by the client.

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

404 **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

405

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

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

411 3.1.2.1.2 Validate operation identifier

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

415 3.1.2.1.3 Validate the request identifier

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

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

422 3.1.2.1.4 Validate attribute group and attribute presence and order

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

424 3.1.2.1.4.1 Validate the presence and order of attribute groups

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

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

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

437 3.1.2.1.4.2 Ignore unknown attribute groups in the expected position

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

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

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

450 **3.1.2.1.4.3 Validate the presence of a single occurrence of required Operation attributes**

451 Client requests and IPP object responses contain Operation attributes that [RFC2911] Section 3
 452 requires to be present. Attributes within a group may be in any order, except for the ordering of target,
 453 charset, and natural languages attributes. These attributes **MUST** be first, and **MUST** be supplied in the
 454 following order: charset, natural language, and then target. An IPP object verifies that the attributes that
 455 Section 4 requires to be supplied by the client have been supplied in the request (attributes without an *
 456 in the following tables). An asterisk (*) indicates groups and Operation attributes that the client may
 457 omit in a request or an IPP object may omit in a response.

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

461 REJECTS the request and RETURNS the 'client-error-bad-request' status code

462 accepts the request and uses the first occurrence of the attribute no matter where it is

463 accepts the request and uses the last occurrence of the attribute no matter where it is

464 accept the request and assume some default value for the missing attribute

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

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

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

479 R indicates a **REQUIRED** attribute or operation that an IPP object **MUST** support

480 O indicates an **OPTIONAL** attribute or operation that an IPP object **NEED NOT** support

481 * indicates that a client MAY omit the attribute in a request and that an IPP object MAY omit
482 the attribute in a response. The absence of an * means that a client MUST supply the
483 attribute in a request and an IPP object MUST supply the attribute in a response.
484 + indicates that this is not a IPP/1.0 operation, but is only a part of IPP/1.1 and future versions
485 of IPP.

486

487 Operation Requests

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

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

491

492 Print-Job Request (R):

493 Group 1: Operation Attributes (R)
494 attributes-charset (R)
495 attributes-natural-language (R)
496 printer-uri (R)
497 requesting-user-name (R*)
498 job-name (R*)
499 ipp-attribute-fidelity (R*)
500 document-name (R*)
501 document-format (R*)
502 document-natural-language (O*)
503 compression (R*)
504 job-k-octets (O*)
505 job-impressions (O*)
506 job-media-sheets (O*)
507 Group 2: Job Template Attributes (R*)
508 <Job Template attributes> (O*)
509 (see [RFC2911] Section 4.2)
510 Group 3: Document Content (R)
511 <document content>

512

513 Validate-Job Request (R):

514 Group 1: Operation Attributes (R)
515 attributes-charset (R)
516 attributes-natural-language (R)
517 printer-uri (R)
518 requesting-user-name (R*)
519 job-name (R*)
520 ipp-attribute-fidelity (R*)
521 document-name (R*)
522 document-format (R*)
523 document-natural-language (O*)
524 compression (R*)
525 job-k-octets (O*)
526 job-impressions (O*)

527 job-media-sheets (O*)
528 Group 2: Job Template Attributes (R*)
529 <Job Template attributes> (O*)
530 (see [RFC2911] Section 4.2)
531
532 Print-URI Request (O):
533 Group 1: Operation Attributes (R)
534 attributes-charset (R)
535 attributes-natural-language (R)
536 printer-uri (R)
537 document-uri (R)
538 requesting-user-name (R*)
539 job-name (R*)
540 ipp-attribute-fidelity (R*)
541 document-name (R*)
542 document-format (R*)
543 document-natural-language (O*)
544 compression (R*)
545 job-k-octets (O*)
546 job-impressions (O*)
547 job-media-sheets (O*)
548 Group 2: Job Template Attributes (R*)
549 <Job Template attributes> (O*) (see
550 (see [RFC2911] Section 4.2)
551
552 Create-Job Request (O):
553 Group 1: Operation Attributes (R)
554 attributes-charset (R)
555 attributes-natural-language (R)
556 printer-uri (R)
557 requesting-user-name (R*)
558 job-name (R*)
559 ipp-attribute-fidelity (R*)
560 job-k-octets (O*)
561 job-impressions (O*)
562 job-media-sheets (O*)
563 Group 2: Job Template Attributes (R*)
564 <Job Template attributes> (O*) (see
565 (see [RFC2911] Section 4.2)
566
567 Get-Printer-Attributes Request (R):
568 Group 1: Operation Attributes (R)
569 attributes-charset (R)
570 attributes-natural-language (R)
571 printer-uri (R)
572 requesting-user-name (R*)
573 requested-attributes (R*)
574 document-format (R*)
575
576 Get-Jobs Request (R):

```
577     Group 1: Operation Attributes (R)
578         attributes-charset (R)
579         attributes-natural-language (R)
580         printer-uri (R)
581         requesting-user-name (R*)
582         limit (R*)
583         requested-attributes (R*)
584         which-jobs (R*)
585         my-jobs (R*)
586
587     Send-Document Request (O):
588         Group 1: Operation Attributes (R)
589             attributes-charset (R)
590             attributes-natural-language (R)
591             (printer-uri & job-id) | job-uri (R)
592             last-document (R)
593             requesting-user-name (R*)
594             document-name (R*)
595             document-format (R*)
596             document-natural-language (O*)
597             compression (R*)
598         Group 2: Document Content (R*)
599             <document content>
600
601     Send-URI Request (O):
602         Group 1: Operation Attributes (R)
603             attributes-charset (R)
604             attributes-natural-language (R)
605             (printer-uri & job-id) | job-uri (R)
606             last-document (R)
607             document-uri (R)
608             requesting-user-name (R*)
609             document-name (R*)
610             document-format (R*)
611             document-natural-language (O*)
612             compression (R*)
613
614     Cancel-Job Request (R):
615     Release-Job Request (O+):
616         Group 1: Operation Attributes (R)
617             attributes-charset (R)
618             attributes-natural-language (R)
619             (printer-uri & job-id) | job-uri (R)
620             requesting-user-name (R*)
621             message (O*)
622
623     Get-Job-Attributes Request (R):
624         Group 1: Operation Attributes (R)
625             attributes-charset (R)
626             attributes-natural-language (R)
```

627 (printer-uri & job-id) | job-uri (R)
628 requesting-user-name (R*)
629 requested-attributes (R*)
630
631 Pause-Printer Request (O+):
632 Resume-Printer Request (O+):
633 Purge-Printer Request (O+):
634 Group 1: Operation Attributes (R)
635 attributes-charset (R)
636 attributes-natural-language (R)
637 printer-uri (R)
638 requesting-user-name (R*)
639
640 Hold-Job Request (O+):
641 Restart-Job Request (O+):
642 Group 1: Operation Attributes (R)
643 attributes-charset (R)
644 attributes-natural-language (R)
645 (printer-uri & job-id) | job-uri (R)
646 requesting-user-name (R*)
647 job-hold-until (R*)
648 message (O*)
649
650 Operation Responses

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

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

654
655 Print-Job Response (R):
656 Create-Job Response (O):
657 Send-Document Response (O):
658 Group 1: Operation Attributes (R)
659 attributes-charset (R)
660 attributes-natural-language (R)
661 status-message (O*)
662 detailed-status-message (O*)
663 Group 2: Unsupported Attributes (R*) (see Note 3)
664 <unsupported attributes> (R*)
665 Group 3: Job Object Attributes(R*) (see Note 2)
666 job-uri (R)
667 job-id (R)
668 job-state (R)
669 job-state-reasons (O* | R+)
670 job-state-message (O*)
671 number-of-intervening-jobs (O*)
672
673 Validate-Job Response (R):

674 Cancel-Job Response (R):
675 Hold-Job Response (O+):
676 Release-Job Response (O+):
677 Restart-Job Response (O+):
678 Group 1: Operation Attributes (R)
679 attributes-charset (R)
680 attributes-natural-language (R)
681 status-message (O*)
682 detailed-status-message (O*)
683 Group 2: Unsupported Attributes (R*) (see Note 3)
684 <unsupported attributes> (R*)
685
686 Print-URI Response (O):
687 Send-URI Response (O):
688 Group 1: Operation Attributes (R)
689 attributes-charset (R)
690 attributes-natural-language (R)
691 status-message (O*)
692 detailed-status-message (O*)
693 document-access-error (O*)
694 Group 2: Unsupported Attributes (R*) (see Note 3)
695 <unsupported attributes> (R*)
696 Group 3: Job Object Attributes (R*) (see Note 2)
697 job-uri (R)
698 job-id (R)
699 job-state (R)
700 job-state-reasons (O* | R+)
701 job-state-message (O*)
702 number-of-intervening-jobs (O*)
703
704 Get-Printer-Attributes Response (R):
705 Group 1: Operation Attributes (R)
706 attributes-charset (R)
707 attributes-natural-language (R)
708 status-message (O*)
709 detailed-status-message (O*)
710 Group 2: Unsupported Attributes (R*) (see Note 4)
711 <unsupported attributes> (R*)
712 Group 3: Printer Object Attributes (R*) (see Note 2)
713 <requested attributes> (R*)
714
715 Get-Jobs Response (R):
716 Group 1: Operation Attributes (R)
717 attributes-charset (R)
718 attributes-natural-language (R)
719 status-message (O*)
720 detailed-status-message (O*)
721 Group 2: Unsupported Attributes (R*) (see Note 4)
722 <unsupported attributes> (R*)
723 Group 3: Job Object Attributes (R*) (see Note 2, 5)

724 <requested attributes> (R*)
 725
 726 Get-Job-Attributes Response (R):
 727 Group 1: Operation Attributes (R)
 728 attributes-charset (R)
 729 attributes-natural-language (R)
 730 status-message (O*)
 731 detailed-status-message (O*)
 732 Group 2: Unsupported Attributes (R*) (see Note 4)
 733 <unsupported attributes> (R*)
 734 Group 3: Job Object Attributes (R*) (see Note 2)
 735 <requested attributes> (R*)
 736
 737 Pause-Printer Response (O+):
 738 Resume-Printer Response (O+):
 739 Purge-Printer Response (O+):
 740 Group 1: Operation Attributes (R)
 741 attributes-charset (R)
 742 attributes-natural-language (R)
 743 status-message (O*)
 744 detailed-status-message (O*)
 745 Group 2: Unsupported Attributes (R*) (see Note 4)
 746 <unsupported attributes> (R*)
 747

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

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

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

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

756 3.1.2.1.5 **Validate the values of the REQUIRED Operation attributes**

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

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

- 761 a) that the length of each Operation attribute value is correct for the attribute syntax tag
- 762 supplied by the client according to [RFC2911] Section 4.1,

- 763 b) that the attribute syntax tag is correct for that Operation attribute according to
764 [RFC2911] Section 3,
- 765 c) that the value is in the range specified for that Operation attribute according to
766 [RFC2911] Section 3,
- 767 d) that multiple values are supplied by the client only for operation attributes that are multi-
768 valued, i.e., that are 1setOf X according to [RFC2911] Section 3.

769

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

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

782

783

attributes-charset (charset)

- 784 IF NOT a single non-empty 'charset' value, REJECT/RETURN 'client-error-bad-request'.
785 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-
786 long'.
787 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-
788 charset-not-supported".

789

790

attributes-natural-language(naturalLanguage)

- 791 IF NOT a single non-empty 'naturalLanguage' value, REJECT/RETURN 'client-error-bad-
792 request'.
793 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-
794 long'.
795 ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-
796 language-supported" attribute. If the supplied value is not a member of the Printer
797 object's "generated-natural-language-supported" attribute, use the Printer object's
798 "natural-language-configured" value.

799

800

requesting-user-name

801 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
802 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
803 too-long'.
804 IF the IPP object can obtain a better-authenticated name, use it instead.
805
806 job-name(name)

807 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
808 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
809 too-long'.
810 IF NOT supplied by the client, the Printer object creates a name from the document-name or
811 document-uri.
812
813 document-name (name)

814 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
815 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
816 too-long'.
817
818 ipp-attribute-fidelity (boolean)

819 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-
820 error-bad-request'.
821 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-
822 long'.
823 IF NOT supplied by the client, the IPP object assumes the value 'false'.
824
825 document-format (mimeType)

826 IF NOT a single non-empty 'mimeType' value, REJECT/RETURN 'client-error-bad-
827 request'.
828 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
829 too-long'.
830 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN
831 'client-error-document-format-not-supported'.
832 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's
833 "document-format-default" attribute.
834
835 document-uri (uri)

836 IF NOT a single non-empty 'uri' value, REJECT/RETURN 'client-error-bad-request'.
837 IF the value length is greater than 1023 octets, REJECT/RETURN 'client-error-request-value-
838 too-long'.
839 IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.

840 If the client-supplied URI scheme is not supported, i.e. the value is not in the Printer object's
841 referenced-uri-scheme-supported" attribute, the Printer object MUST reject the request
842 and return the 'client-error-uri-scheme-not-supported' status code. The Printer object
843 MAY check to see if the document exists and is accessible. If the document is not found
844 or is not accessible, REJECT/RETURN 'client-error-not found'.
845 last-document (boolean)
846 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-
847 error-bad-request'.
848 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-
849 long'
850
851 job-id (integer(1:MAX))
852 IF NOT an single 'integer' value equal to 4 octets AND in the range 1 to MAX,
853 REJECT/RETURN 'client-error-bad-request'.
854 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-
855 error-gone' status code, if keep track of recently deleted jobs.
856
857 requested-attributes (1setOf keyword)
858 IF NOT one or more 'keyword' values, REJECT/RETURN 'client-error-bad-request'.
859 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
860 too-long'.
861 Ignore unsupported values, which are the keyword names of unsupported attributes. Don't
862 bother to copy such requested (unsupported) attributes to the Unsupported Attribute
863 response group since the response will not return them.
864
865 which-jobs (type2 keyword)
866 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
867 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
868 too-long'.
869 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to
870 the Unsupported Attributes response group and REJECT/RETURN 'client-error-
871 attributes-or-values-not-supported'.
872 Note: a Printer still supports the 'completed' value even if it keeps no
873 completed/canceled/aborted jobs: by returning no jobs when so queried.
874 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.
875
876 my-jobs (boolean)
877 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-
878 error-bad-request'.
879 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-
880 long'
881 IF NOT supplied by the client, the IPP object assumes the 'false' value.

882

883 limit (integer(1:MAX))

884

IF NOT a single 'integer' value equal to 4 octets AND in the range 1 to MAX,

885

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

886

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

887

888

889

890 **3.1.2.1.6 Validate the values of the OPTIONAL Operation attributes**

891

OPTIONAL Operation attributes are those that an IPP object MAY support. An IPP object validates the values of the OPTIONAL attributes supplied by the client. The IPP object performs the same syntactic validation checks for each OPTIONAL attribute value as in Section 3.1.2.1.5. As in Section 3.1.2.1.5, if any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' or the 'client-error-request-value-too-long' status code.

892

893

894

895

896

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

897

898

899

900

901

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

902

903

904

document-natural-language (naturalLanguage)

905

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

906

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

907

908

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

909

910

911

912

compression (type3 keyword)

913

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

914

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

915

916

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

917

918 Note to IPP/1.0 implementers: Support for the "compression" attribute was optional in IPP/1.0 and
919 was changed to REQUIRED in IPP/1.1. However, an IPP/1.0 object SHOULD at least
920 check for the "compression" attribute being present and reject the create request, if they don't
921 support "compression". Not checking is a bug, since the data will be unintelligible.
922

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

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

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

926 IF NOT in the range of the Printer object's "job-k-octets-supported" attribute, copy the attribute and
927 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN
928 'client-error-attributes-or-values-not-supported'.
929

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

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

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

933 IF NOT in the range of the Printer object's "job-impressions-supported" attribute, copy the attribute
934 and the unsupported value to the Unsupported Attributes response group and
935 REJECT/RETURN 'client-error-attributes-or-values-not-supported'.
936

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

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

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

940 IF NOT in the range of the Printer object's "job-media-sheets-supported" attribute, copy the attribute
941 and the unsupported value to the Unsupported Attributes response group and
942 REJECT/RETURN 'client-error-attributes-or-values-not-supported'.
943

944 message (text(127))

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

946 IF the value length is greater than 127 octets,

947 REJECT/RETURN 'client-error-request-value-too-long'.
948

949 unknown or unsupported attribute

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

952 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
953 attribute value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.
954

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

970

010 **3.1.2.2.2 Check that the Printer object is accepting jobs**

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

013 **3.1.2.2.3 Validate the values of the Job Template attributes**

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

- 017 a) that the length of each value is correct for the attribute syntax tag supplied by the client
018 according to [RFC2911] Section 4.1.
- 019 b) that the attribute syntax tag is correct for that attribute according to [RFC2911]
020 Sections 4.2 to 4.4.
- 021 c) that multiple values are supplied only for multi-valued attributes, i.e., that are 1setOf X
022 according to [RFC2911] Sections 4.2 to 4.4.

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

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

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

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

037 **3.1.2.3 Algorithm for job validation**

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

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

- 042 1. If U is multi-valued, validate each value X of U by performing the algorithm in Table 7 with each
043 value X. Each validation is separate from the standpoint of returning unsupported values.
044 Example: If U is "finishings" that the client supplies with 'staple', 'bind' values, then X takes on
045 the successive values: 'staple', then 'bind'
- 046 2. If V is multi-valued, validate X against each Z of V by performing the algorithm in Table 7 with
047 each value Z. If a value Z validates, the validation for the attribute value X succeeds. If it fails,
048 the algorithm is applied to the next value Z of V. If there are no more values Z of V, validation
049 fails. Example" If V is "sides-supported" with values: 'one-sided', 'two-sided-long', and 'two-
050 sided-short', then Z takes on the successive values: 'one-sided', 'two-sided-long', and 'two-sided-
051 short'. If the client supplies "sides" with 'two-sided-long', the first comparison fails ('one-sided' is
052 not equal to 'two-sided-long'), the second comparison succeeds ('two-sided-long' is equal to 'two-
053 sided-long"), and the third comparison ('two-sided-short' with 'two-sided-long') is not even
054 performed.
- 055 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.

056 **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.

057

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

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

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

075 -----

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

077 IF NOT a single 'integer' value with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
078 request'.
079 IF NOT supplied by the client, use the value of the Printer object's "job-priority-default" attribute at
080 job submission time.
081 IF NOT in the range 1 to 100, inclusive, copy the attribute and the unsupported value to the
082 Unsupported Attributes response group.
083 Map the value to the nearest supported value in the range 1:100 as specified by the number of
084 discrete values indicated by the value of the Printer's "job-priority-supported" attribute. See
085 the formula in [RFC2911] Section 4.2.1.
086

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

088 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
089 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
090 long'.
091 IF NOT supplied by the client, use the value of the Printer object's "job-hold-until" attribute at job
092 submission time.
093 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the
094 unsupported value to the Unsupported Attributes response group.
095

096 job-sheets (type3 keyword | name)

097 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
098 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
099 long'.
100 IF NOT in the Printer object's "job-sheets-supported" attribute, copy the attribute and the
101 unsupported value to the Unsupported Attributes response group.
102

103 multiple-document-handling (type2 keyword)

104 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
105 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
106 long'.
107 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute
108 and the unsupported value to the Unsupported Attributes response group.
109

110 copies (integer(1:MAX))

111 IF NOT a single 'integer' value with a length equal to 4 octets,
112 REJECT/RETURN 'client-error-bad-request'.
113 IF NOT in range of the Printer object's "copies-supported" attribute
114 copy the attribute and the unsupported value to the Unsupported Attributes response group.
115
116 finishings (1setOf type2 enum)

117 IF NOT an 'enum' value(s) each with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
118 request'.
119 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the
120 unsupported value(s), but not any supported values, to the Unsupported Attributes response
121 group.
122
123 page-ranges (1setOf rangeOfInteger(1:MAX))

124 IF NOT a 'rangeOfInteger' value(s) each with a length equal to 8 octets, REJECT/RETURN 'client-
125 error-bad-request'.
126 IF first value is greater than second value in any range, the ranges are not in ascending order, or
127 ranges overlap, REJECT/RETURN 'client-error-bad-request'.
128 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to
129 the Unsupported Attributes response group and set the value to the "out-of-band"
130 'unsupported' value.
131
132 sides (type2 keyword)

133 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
134 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
135 long'.
136 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported
137 value to the Unsupported Attributes response group.
138
139 number-up (integer(1:MAX))

140 IF NOT a single 'integer' value with a length equal to 4 octets,
141 REJECT/RETURN 'client-error-bad-request'.
142 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"
143 attribute, copy the attribute and value to the Unsupported Attribute response group.
144
145 orientation-requested (type2 enum)

146 IF NOT a single 'enum' value with a length equal to 4 octets,
147 REJECT/RETURN 'client-error-bad-request'.
148 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the
149 unsupported value to the Unsupported Attributes response group.
150
151 media (type3 keyword | name)

152 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
 153 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
 154 long'.
 155 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported
 156 value to the Unsupported Attributes response group.
 157
 158 printer-resolution (resolution)

159 IF NOT a single 'resolution' value with a length equal to 9 octets,
 160 REJECT/RETURN 'client-error-bad-request'.
 161 IF NOT in the Printer object's "printer-resolution-supported" attribute, copy the attribute and the
 162 unsupported value to the Unsupported Attributes response group.
 163
 164 print-quality (type2 enum)

165 IF NOT a single 'enum' value with a length equal to 4 octets,
 166 REJECT/RETURN 'client-error-bad-request'.
 167 IF NOT in the Printer object's "print-quality-supported" attribute, copy the attribute and the
 168 unsupported value to the Unsupported Attributes response group.
 169
 170 unknown or unsupported attribute (i.e., there is no corresponding Printer object "xxx-supported"
 171 attribute)

172 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute
 173 syntax,
 174 REJECT/RETURN 'client-error-bad-request' if the length of the attribute syntax is fixed or 'client-
 175 error-request-value-too-long' if the length of the attribute syntax is variable.
 176 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
 177 attribute value to the "out-of-band" 'unsupported' value. Any remaining Job Template
 178 Attributes are either unknown or unsupported Job Template attributes and are validated
 179 algorithmically according to their attribute syntax for proper length (see below).
 180 -----

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

188 Name	188 Octet length check for read-write attributes	
189 -----	189 -----	
190 'textWithLanguage	<= 1023 AND	'naturalLanguage' <= 63
191 'textWithoutLanguage'	<= 1023	
192 'nameWithLanguage'	<= 255 AND	'naturalLanguage' <= 63
193 'nameWithoutLanguage'	<= 255	

```

194     'keyword'           <= 255
195     'enum'             = 4
196     'uri'              <= 1023
197     'uriScheme'       <= 63
198     'charset'         <= 63
199     'naturalLanguage' <= 63
200     'mimeType'         <= 255
201     'octetString'     <= 1023
202     'boolean'         = 1
203     'integer'         = 4
204     'rangeOfInteger'  = 8
205     'dateTime'       = 11
206     'resolution'     = 9
207     'lsetOf X'
208

```

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

214 3.1.2.3.1 Check for conflicting Job Template attributes values

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

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

228 3.1.2.3.2 Decide whether to REJECT the request

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

- 232 1. 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes
 233 supplied by the client.

234 2. 'client-error-attributes-or-values-not-supported' status code, otherwise.

235

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

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

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

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

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

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

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

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

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

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

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

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

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

276 **3.1.2.3.3 For the Validate-Job operation, RETURN one of the success status codes**

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

- 278 1. the "successful-ok" status code, if there are no unsupported or conflicting Job Template
279 attributes or values.
- 280 2. the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or
281 values.
- 282 3. the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template
283 attributes or values.

284

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

289 **3.1.2.3.4 Create the Job object with attributes to support**

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

- 291 1. creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and
292 initializes all of the job's other supported Job Description attributes.
- 293 2. removes all unsupported attributes from the Job object.
- 294 3. for each unsupported value, removes either the unsupported value or substitutes the
295 unsupported attribute value with some supported value. If an attribute has no values after
296 removing unsupported values from it, the attribute is removed from the Job object (so that the
297 normal default behavior at job processing time will take place for that attribute).
- 298 4. for each conflicting value, removes either the conflicting value or substitutes the conflicting
299 attribute value with some other supported value. If an attribute has no values after removing
300 conflicting values from it, the attribute is removed from the Job object (so that the normal
301 default behavior at job processing time will take place for that attribute).

302

303 If there were no attributes or values flagged as unsupported, or the value of 'ipp-attribute-fidelity' was
304 'false', the Printer object is able to accept the create request and create a new Job object. If the "ipp-
305 attribute-fidelity" attribute is set to 'true', the Job Template attributes that populate the new Job object
306 are necessarily all the Job Template attributes supplied in the create request. If the "ipp-attribute-
307 fidelity" attribute is set to 'false', the Job Template attributes that populate the new Job object are all the
308 client supplied Job Template attributes that are supported or that have value substitution. Thus, some
309 of the requested Job Template attributes will not appear in the Job object because the Printer object did
310 not support those attributes. The attributes that populate the Job object are persistently stored with the
311 Job object for that Job. A Get-Job-Attributes operation on that Job object will return only those
312 attributes that are persistently stored with the Job object.

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

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

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

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

334 **3.1.2.3.5 Return one of the success status codes**

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

- 336 1. the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes
337 or values.
- 338 2. the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template
339 attribute or values.

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

342

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

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

349 **3.1.2.3.6 Accept appended Document Content**

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

352 **3.1.2.3.7 Scheduling and Starting to Process the Job**

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

358 **3.1.2.3.8 Completing the Job**

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

364 **3.1.2.3.9 Destroying the Job after completion**

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

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

372 **3.1.2.3.10 Interaction with "ipp-attribute-fidelity"**

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

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

384 **3.1.2.3.11 Character set code conversion support**

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

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

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

397 According to the RFC2911 document (section 3.1.4.2), the value of the "attributes-charset" for the
398 response of the second request must be "us-ascii" since that is the charset specified in the request. The
399 "job-name" value, however, is in "utf-8" format. Should the request be rejected even though both "utf-
400 8" and "us-ascii" charsets are supported by the server? or should the "job-name" value be converted to
401 "us-ascii" and return "successful-ok-conflicting-attributes" (0x0002) as the status code?

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

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

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

415 **3.1.2.3.12 What charset to return when an unsupported charset is requested (Issue 1.19)?**

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

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

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

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

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

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

433 **3.1.2.3.13 Natural Language Override (NLO)**

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

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

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

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

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

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

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

477 **3.1.3 Status codes returned by operation**

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

481 **3.1.3.1 Printer Operations**

482 3.1.3.1.1 Print-Job

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

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

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

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

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

498
499 [RFC2911] section 3.1.6 Operation Status Codes and Messages states:

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

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

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

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

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

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

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

518 client-error-timeout: not applicable.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

559 3.1.3.1.2 Print-URI

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

563 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation
564 attribute is not supported and is returned in the Unsupported Attributes group.
565 server-error-operation-not-supported: the Print-URI operation is not supported.
566

567 3.1.3.1.3 Validate-Job

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

570 3.1.3.1.4 Create-Job

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

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

578 3.1.3.1.5 Get-Printer-Attributes

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

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

583 successful-ok: no operation attributes or values were substituted or ignored (same as Print-Job) and
584 no requested attributes were unsupported.
585 successful-ok-ignored-or-substituted-attributes: The "requested-attributes" operation attribute
586 MAY, but NEED NOT, be returned with the unsupported values.
587 successful-ok-conflicting-attributes: same as Print-Job.
588

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

590 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any
591 requests.
592 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.
593 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation
594 attributes and/or values MUST be ignored and an appropriate success code returned (see above).
595 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not
596 involved.
597 server-error-operation-not-supported: not applicable (since Get-Printer-Attributes is REQUIRED).
598 server-error-device-error: same as Print-Job, except that no document data is involved.

599 server-error-temporary-error: same as Print-Job, except that no document data is involved.
600 server-error-not-accepting-jobs: not applicable.
601 server-error-busy: same as Print-Job, except the IPP object is too busy to accept even query
602 requests.
603 server-error-job-canceled: not applicable.

604 **3.1.3.1.6 Get-Jobs**

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

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

609 successful-ok: same as Get-Printer-Attributes (see section 3.1.3.1.5).
610 successful-ok-ignored-or-substituted-attributes: same as Get-Printer-Attributes (see section
611 3.1.3.1.5).
612 successful-ok-conflicting-attributes: same as Get-Printer-Attributes (see section 3.1.3.1.5).
613

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

617 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any
618 requests.
619 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.
620 client-error-document-format-not-supported: not applicable.
621 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation
622 attributes and/or values MUST be ignored and an appropriate success code returned (see
623 above).
624 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not
625 involved.
626 server-error-operation-not-supported: not applicable (since Get-Jobs is REQUIRED).
627 server-error-device-error: same as Print-Job, except that no document data is involved.
628 server-error-temporary-error: same as Print-Job, except that no document data is involved.
629 server-error-not-accepting-jobs: not applicable.
630 server-error-job-canceled: not applicable.

631 **3.1.3.1.7 Pause-Printer**

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

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

637 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
638 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
639 successful-ok-conflicting-attributes: same as Print-Job.

640

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

643 client-error-not-possible: not applicable.
644 client-error-not-found: the target Printer object does not exist.
645 client-error-gone: the target Printer object no longer exists and no forwarding address is known.
646 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
647 client-error-document-format-not-supported: not applicable.
648 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-
649 accepting-jobs" attribute is not involved.
650 server-error-operation-not-supported: the Pause-Printer operation is not supported.
651 server-error-device-error: not applicable.
652 server-error-temporary-error: same as Print-Job, except no document data is involved.
653 server-error-not-accepting-jobs: not applicable.
654 server-error-job-canceled: not applicable.

655 **3.1.3.1.8 Resume-Printer**

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

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

660 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
661 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
662 successful-ok-conflicting-attributes: same as Print-Job.

663

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

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

666

667 **3.1.3.1.8.1 What about Printers unable to change state due to an error condition?**

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

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

674 **3.1.3.1.8.2** **How is "printer-state" handled on Resume-Printer?**

675

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

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

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

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

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

686 **3.1.3.1.9** **Purge-Printer**

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

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

- 691 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
- 692 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
- 693 successful-ok-conflicting-attributes: same as Print-Job.

694

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

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

697 **3.1.3.2** **Job Operations**

698 **3.1.3.2.1** **Send-Document**

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

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

704 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
705 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
706 successful-ok-conflicting-attributes: same as Print-Job.

707

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

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

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

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

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

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

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

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

726 **3.1.3.2.2 Send-URI**

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

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

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

734

735 **3.1.3.2.3 Cancel-Job**

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

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

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

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

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

743

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

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

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

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

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

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

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

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

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

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

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

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

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

761 **3.1.3.2.4 Get-Job-Attributes**

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

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

766 successful-ok: same as Get-Printer-Attributes (see section 3.1.3.1.5).

767 successful-ok-ignored-or-substituted-attributes: same as Get-Printer-Attributes (see section
768 3.1.3.1.5).

769 successful-ok-conflicting-attributes: same as Get-Printer-Attributes (see section 3.1.3.1.5).

770

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

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

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

775 client-error-attributes-or-values-not-supported: not applicable.

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

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

780 client-error-conflicting-attributes: not applicable

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

782 server-error-device-error: same as Print-Job, except no document data is involved.
783 server-error-temporary-error: sane as Print-Job, except no document data is involved..
784 server-error-not-accepting-jobs: not applicable.
785 server-error-job-canceled: not applicable.

786 **3.1.3.2.5 Hold-Job**

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

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

791 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
792 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
793 successful-ok-conflicting-attributes: same as Print-Job.

794

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

796 client-error-not-possible: The request cannot be carried out because of the state of the Job
797 object ('completed', 'canceled', or 'aborted') or the state of the system.
798 client-error-not-found: the target Printer and/or Job object does not exist.
799 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding
800 address is known.
801 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
802 client-error-document-format-not-supported: not applicable.
803 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-
804 accepting-jobs" attribute is not involved.
805 server-error-operation-not-supported: the Hold-Job operation is not supported.
806 server-error-device-error: not applicable.
807 server-error-temporary-error: same as Print-Job, except no document data is involved.
808 server-error-not-accepting-jobs: not applicable.
809 server-error-job-canceled: not applicable.

810 **3.1.3.2.6 Release-Job**

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

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

815 **3.1.3.2.7 Restart-Job**

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

819 server-error-operation-not-supported: the Restart-Job operation is not supported.
820

821 **3.1.3.2.7.1 Can documents be added to a restarted job?**

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

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

831 **3.1.4 Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)**

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

838 **3.1.5 Sending empty attribute groups**

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

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

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

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

856 3.2 Printer Operations

857 3.2.1 Print-Job operation

858 3.2.1.1 Flow controlling the data portion of a Print-Job request (Issue 1.22)

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

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

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

868 3.2.1.2 Returning job-state in Print-Job response (Issue 1.30)

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

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

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

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

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

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

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

891 **3.2.2 Get-Printer-Attributes operation**

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

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

897 **3.2.3 Get-Jobs operation**

898 **3.2.3.1 Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?**

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

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

905 **3.2.3.2 Why is there a "limit" attribute in the Get-Jobs operation?**

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

915 3.2.4 Create-Job operation

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

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

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

932 3.3 Job Operations

933 3.3.1 Validate-Job

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

938 3.3.2 Restart-Job

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

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

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

948 **4 Object Attributes**

949 **4.1 Attribute Syntax's**

950 **4.1.1 The 'none' value for empty sets (Issue 1.37)**

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

959 **4.1.2 Multi-valued attributes (Issue 1.31)**

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

966 **4.1.3 Case Sensitivity in URIs (issue 1.6)**

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

973 The reason that the IPP specification does not make any restrictions on URIs, is so that implementations
974 of IPP may use off-the-shelf components that conform to the standards that define URIs, such as RFC
975 2396 and the HTTP/1.1 specifications [RFC2616]. See these specifications for rules of matching,
976 comparison, and case-sensitivity.

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

980 Example of equivalent URI's

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

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

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

984 Example of equivalent URI's using the IPP scheme

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

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

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

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

989 **4.1.4 Maximum length for xxxWithLanguage and xxxWithoutLanguage**

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

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

997 **4.2 Job Template Attributes**

998 **4.2.1 multiple-document-handling(type2 keyword)**

999 **4.2.1.1 Support of multiple document jobs**

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

008 4.3 Job Description Attributes

009 4.3.1 Getting the date and time of day

010 The "date-time-at-creation", "date-time-at-processing", and "date-time-at-completed" attributes are
011 returned as dateTime syntax. These attributes are OPTIONAL for a Printer to support. However,
012 there are various ways for a Printer to get the date and time of day. Some suggestions:

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

021 4.4 Printer Description Attributes

022 4.4.1 queued-job-count (integer(0:MAX))

023 4.4.1.1 Why is "queued-job-count" RECOMMENDED (Issue 1.14)?

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

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

030 4.4.1.2 Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?

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

035 **4.4.2 printer-current-time (dateTime)**

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

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

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

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

050 **4.4.3 Printer-uri**

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

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

059 **4.5 Empty Jobs**

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

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

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

071 **5 Directory Considerations**

072 **5.1 General Directory Schema Considerations**

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

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

082 **5.2 IPP Printer with a DNS name**

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

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

086 **6 Security Considerations**

087 The security considerations given in [RFC2911] Section 8 "Security Considerations" all apply to this
088 document. In addition, the following sub-sections describes security consideration that have arisen as a
089 result of implementation testing.

090 **6.1 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)**

091 The following clarification was added to [RFC2911] section 8.5:

092 8.5 Queries on jobs submitted using non-IPP protocols

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

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

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

111 7 Encoding and Transport

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

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

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

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

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

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

- 140 - the "header" column contains the name of a header
- 141 - the "request/client" column indicates whether a client sends the header.
- 142 - the "request/ server" column indicates whether a server supports the header when received.
- 143 - the "response/ server" column indicates whether a server sends the header.
- 144 - the "response /client" column indicates whether a client supports the header when received.
- 145 - the "values and conditions" column specifies the allowed header values and the conditions for the
146 header to be present in a request/response.

147
148 The table for "request headers" does not have columns for responses, and the table for "response
149 headers" does not have columns for requests.

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

- 151 - **must:** the client or server **MUST** send the header,
- 152 - **must-if:** the client or server **MUST** send the header when the condition described in the "values
153 and conditions" column is met,
- 154 - **may:** the client or server **MAY** send the header
- 155 - **not:** the client or server **SHOULD NOT** send the header. It is not relevant to an IPP
156 implementation.

157
158 The following is an explanation of the values in the "response/client" and "request/ server" columns.

- 159 - **must:** the client or server **MUST** support the header,
- 160 - **may:** the client or server **MAY** support the header
- 161 - **not:** the client or server **SHOULD NOT** support the header. It is not relevant to an IPP
162 implementation.

163 7.1 General Headers

164 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	

165 **7.2 Request Headers**

166 The following is a table for the request headers.

Request-Header	Client	Server	Request Values and Conditions
----------------	--------	--------	-------------------------------

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	

167 **7.3 Response Headers**

168 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.

169 7.4 Entity Headers

170 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	

171 7.5 Optional support for HTTP/1.0

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

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

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

183 **7.6 HTTP/1.1 Chunking**

184 **7.6.1 Disabling IPP Server Response Chunking**

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

189 TE: identity

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

192 **7.6.2 Warning About the Support of Chunked Requests**

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

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

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

206 **8 References**

207 [CGI]

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

- 209 [IANA-CS]
210 IANA Registry of Coded Character Sets: <ftp://ftp.iana.org/in-notes/iana/assignments/character-sets>
- 211 [ldap-printer]
212 Fleming, P., Jones, K., Lewis, H., McDonald, I., "Internet Printing Protocol (IPP): LDAP Schema
213 for Printer Services", work in progress.
- 214 [RFC793]
215 J. Postel, "Transmission Control Protocol", RFC 793, September 1981.
- 216 [RFC1123]
217 Braden, R., "Requirements for Internet Hosts - Application and Support", RFC 1123, October,
218 1989.
- 219 [RFC2026]
220 Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.
- 221 [RFC2119]
222 Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119,
223 March 1997.
- 224 [RFC2396]
225 Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic
226 Syntax", RFC 2396, August 1998.
- 227 [RFC2565]
228 deBry, R., Hastings, T., Herriot, R., Isaacson, S. and P. Powell, "Internet Printing Protocol/1.0:
229 Model and Semantics", RFC 2566, April 1999.
- 230 [RFC2566]
231 Herriot, R., Butler, S., Moore, P. and R. Turner, "Internet Printing Protocol/1.0: Encoding and
232 Transport", RFC 2565, April 1999.
- 233 [RFC2567]
234 Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.
- 235 [RFC2568]
236 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
237 RFC 2568, April 1999.
- 238 [RFC2569]
239 Herriot, R., Hastings, T., Jacobs, N. and J. Martin, "Mapping between LPD and IPP Protocols",
240 RFC 2569, April 1999.
- 241 [RFC2616]
242 Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee,
243 "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.

- 244 [RFC2910]
245 Herriot, R., Butler, S., Moore, P. and R. Turner, "Internet Printing Protocol/1.0: Encoding and
246 Transport", RFC 2910, September, 2000.
- 247 [RFC2911]
248 deBry, R., Hastings, T., Herriot, R., Isaacson, S. and P. Powell, "Internet Printing Protocol/1.0:
249 Model and Semantics", RFC 2911, September, 2000.
- 250 [Servlet]
251 Servlet Specification Version 2.1 (<http://java.sun.com/products/servlet/2.1/index.html>).
- 252 [svrloc-printer]
253 St. Pierre, P., Isaacson, S., McDonald, I., "Definition of the Printer Abstract Service Type v2.0",
254 <http://www.isi.edu/in-notes/iana/assignments/svrloc-templates/printer.2.0.en> (IANA Registered, May
255 27, 2000).
- 256 [SSL]
257 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.

258 **9 Authors' Address**

- 259
260 Thomas N. Hastings
261 Xerox Corporation
262 701 Aviation Blvd.
263 El Segundo, CA 90245
264 hastings@cp10.es.xerox.com
265
- 266 Carl-Uno Manros
267 Independent Consultant
268 1601 N. Sepulveda Blvd. #505
269 Manhattan Beach, CA 90266
270 carl@manros.com
271
- 272 Carl Kugler
273 Mail Stop 003G
274 IBM Printing Systems Co
275 6300 Diagonal Hwy
276 Boulder CO 80301
277 Kugler@us.ibm.com
278
- 279 Henrik Holst
280 i-data Printing Systems
281 Vadstrupvej 35-43
282 2880 Bagsvaerd, Denmark

283 hh@I-data.com

284

285 Peter Zehler

286 Xerox Corporation

287 800 Philips Road

288 Webster, NY 14580

289 pzehler@crt.xerox.com

290

291 IPP Web Page: <http://www.pwg.org/ipp/>

292 IPP Mailing List: ipp@pwg.org

293

294 To subscribe to the ipp mailing list, send the following email:

295 1) send it to majordomo@pwg.org

296 2) leave the subject line blank

297 3) put the following two lines in the message body:

298 subscribe ipp

299

end

300

301 Implementers of this specification document are encouraged to join the IPP Mailing List in order to
302 participate in any discussions of clarification issues and review of registration proposals for additional
303 attributes and values. In order to reduce spam the mailing list rejects mail from non-subscribers, so you
304 must subscribe to the mailing list in order to send a question or comment to the mailing list.

305

306 Other Participants:

Chuck Adams - Tektronix	Shivaun Albright - HP
Stefan Andersson - Axis	Jeff Barnett - IBM
Ron Bergman - Hitachi Koki Imaging Systems	Dennis Carney - IBM
Keith Carter - IBM	Angelo Caruso - Xerox
Rajesh Chawla - TR Computing Solutions	Nancy Chen - Okidata
Josh Cohen - Microsoft	Jeff Copeland - QMS
Andy Davidson - Tektronix	Roger deBry - IBM
Maulik Desai - Auco	Mabry Dozier - QMS
Lee Farrell - Canon Information Systems	Satoshi Fujitami - Ricoh
Steve Gebert - IBM	Sue Gleeson - Digital
Charles Gordon - Osicom	Brian Grimshaw - Apple
Jerry Hadsell - IBM	Richard Hart - Digital
Tom Hastings - Xerox	Henrik Holst - I-data
Stephen Holmstead	Zhi-Hong Huang - Zenographics
Scott Isaacson - Novell	Babek Jahromi - Microsoft
Swen Johnson - Xerox	David Kellerman - Northlake Software
Robert Kline - TrueSpectra	Charles Kong - Panasonic
Carl Kugler - IBM	Dave Kuntz - Hewlett-Packard
Takami Kurono - Brother	Rick Landau - Digital
Scott Lawrence - Agranot Systems	Greg LeClair - Epson
Dwight Lewis - Lexmark	Harry Lewis - IBM
Tony Liao - Vivid Image	Roy Lomicka - Digital
Pete Loya - HP	Ray Lutz - Cognisys
Mike MacKay - Novell, Inc.	David Manchala - Xerox
Carl-Uno Manros - Xerox	Jay Martin - Underscore
Stan McConnell - Xerox	Larry Masinter - Xerox
Sandra Matts - Hewlett Packard	Peter Michalek - Shinesoft
Ira McDonald - High North Inc.	Mike Moldovan - G3 Nova
Tetsuya Morita - Ricoh	Yuichi Niwa - Ricoh
Pat Nogay - IBM	Ron Norton - Printronics
Hugo Parra, Novell	Bob Pentecost - Hewlett-Packard
Patrick Powell - Astart Technologies	Jeff Rackowitz - Intermec
Eric Random - Peerless	Rob Rhoads - Intel
Xavier Riley - Xerox	Gary Roberts - Ricoh
David Roach - Unisys	Stuart Rowley - Kyocera
Yuji Sasaki - Japan Computer Industry	Richard Schneider - Epson
Kris Schoff - HP	Katsuaki Sekiguchi - Canon
Bob Setterbo - Adobe	Gail Songer - Peerless
Hideki Tanaka - Canon	Devon Taylor - Novell, Inc.
Mike Timperman - Lexmark	Atsushi Uchino - Epson
Shigeru Ueda - Canon	Bob Von Andel - Allegro Software
William Wagner - NetSilicon/DPI	Jim Walker - DAZEL
Chris Wellens - Interworking Labs	Trevor Wells - Hewlett Packard

Craig Whittle - Sharp Labs	Rob Whittle - Novell, Inc.
Jasper Wong - Xionics	Don Wright - Lexmark
Michael Wu - Heidelberg Digital	Rick Yardumian - Xerox
Michael Yeung - Toshiba	Lloyd Young - Lexmark
Atsushi Yuki - Kyocera	Peter Zehler - Xerox
William Zhang- Canon Information Systems	Frank Zhao - Panasonic
Steve Zilles - Adobe	Rob Zirnstein - Canon Information Systems

307

308

309 10 Description of the Base IPP Documents

310 In addition to this document, the base set of IPP documents includes:

311 Design Goals for an Internet Printing Protocol [RFC2567]

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

313 Internet Printing Protocol/1.1: Model and Semantics [RFC2911]

314 Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]

315 Mapping between LPD and IPP Protocols [RFC2569]

316

317 The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed
 318 printing functionality, and it enumerates real-life scenarios that help to clarify the features that need to
 319 be included in a printing protocol for the Internet. It identifies requirements for three types of users:
 320 end users, operators, and administrators. It calls out a subset of end user requirements that are satisfied
 321 in IPP/1.0 [RFC2566, RFC2565]. A few OPTIONAL operator operations have been added to IPP/1.1
 322 [RFC2911, RFC2910].

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

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

331 The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the
 332 abstract operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It also
 333 defines the encoding rules for a new Internet MIME media type called "application/ipp". This document
 334 also defines the rules for transporting a message body over HTTP whose Content-Type is
 335 "application/ipp". This document defines the 'ipp' scheme for identifying IPP printers and jobs.

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

338 **11 Full Copyright Statement**

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

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

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

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

356 **Acknowledgement**

357 Funding for the RFC Editor function is currently provided by the Internet Society.