

1 Internet Printing Protocol WG
2 INTERNET-DRAFT
3 draft-ietf-ipp-implementers-guide-v11-04.txt
4 [Obsoletes RFC 2639]
5 [Target category: informational]
6 Expires: January 17, 2002

T. Hastings
C. Manros
P. Zehler
Xerox Corporation
C. Kugler
IBM Printing Systems Co
H. Holst
i-data Printing Systems
September 28, 2001

11 Internet Printing Protocol/1.1: Implementer's Guide

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

13 Status of this Memo

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

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

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

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

23 Abstract

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

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

33

33 **TABLE OF CONTENTS**

34	1	Introduction.....	5
35	1.1	Conformance language.....	5
36	1.2	Other terminology.....	6
37	1.3	Issues Raised from Interoperability Testing Events.....	6
38	2	IPP Objects.....	6
39	3	IPP Operations.....	8
40	3.1	Common Semantics.....	8
41	3.1.1	Summary of Operation Attributes	8
42	3.1.2	Suggested Operation Processing Steps for IPP Objects	14
43	3.1.2.1	Suggested Operation Processing Steps for all Operations	15
44	3.1.2.1.1	Validate version number.....	16
45	3.1.2.1.2	Validate operation identifier	16
46	3.1.2.1.3	Validate the request identifier.....	17
47	3.1.2.1.4	Validate attribute group and attribute presence and order	17
48	3.1.2.1.4.1	Validate the presence and order of attribute groups	17
49	3.1.2.1.4.2	Ignore unknown attribute groups in the expected position.....	17
50	3.1.2.1.4.3	Validate the presence of a single occurrence of required Operation attributes	18
51	3.1.2.1.5	Validate the values of the REQUIRED Operation attributes	24
52	3.1.2.1.6	Validate the values of the OPTIONAL Operation attributes	28
53	3.1.2.2	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add Documents	31
54	3.1.2.2.1	Default "ipp-attribute-fidelity" if not supplied.....	31
55	3.1.2.2.2	Check that the Printer object is accepting jobs.....	32
56	3.1.2.2.3	Validate the values of the Job Template attributes	32
57	3.1.2.3	Algorithm for job validation.....	32
58	3.1.2.3.1	Check for conflicting Job Template attributes values	37
59	3.1.2.3.2	Decide whether to REJECT the request	37
60	3.1.2.3.3	For the Validate-Job operation, RETURN one of the success status codes	39
61	3.1.2.3.4	Create the Job object with attributes to support.....	39
62	3.1.2.3.5	Return one of the success status codes	40
63	3.1.2.3.6	Accept appended Document Content	41
64	3.1.2.3.7	Scheduling and Starting to Process the Job.....	41
65	3.1.2.3.8	Completing the Job	41
66	3.1.2.3.9	Destroying the Job after completion	41
67	3.1.2.3.10	Interaction with "ipp-attribute-fidelity"	42
68	3.1.2.3.11	Character set code conversion support	42
69	3.1.2.3.12	What charset to return when an unsupported charset is requested (Issue 1.19)?.....	43
70	3.1.2.3.13	Natural Language Override (NLO).....	44
71	3.1.3	Status codes returned by operation	45
72	3.1.3.1	Printer Operations.....	45
73	3.1.3.1.1	Print-Job.....	45
74	3.1.3.1.2	Print-URI.....	47
75	3.1.3.1.3	Validate-Job.....	47
76	3.1.3.1.4	Create-Job	47
77	3.1.3.1.5	Get-Printer-Attributes	47
78	3.1.3.1.6	Get-Jobs	48
79	3.1.3.1.7	Pause-Printer	49
80	3.1.3.1.8	Resume-Printer	49
81	3.1.3.1.8.1	What about Printers unable to change state due to an error condition?.....	50

83	3.1.3.1.8.2	How is "printer-state" handled on Resume-Printer?	50
84	3.1.3.1.9	Purge-Printer	50
85	3.1.3.2	Job Operations.....	51
86	3.1.3.2.1	Send-Document	51
87	3.1.3.2.2	Send-URI	52
88	3.1.3.2.3	Cancel-Job.....	52
89	3.1.3.2.4	Get-Job-Attributes	53
90	3.1.3.2.5	Hold-Job.....	53
91	3.1.3.2.6	Release-Job.....	54
92	3.1.3.2.7	Restart-Job	54
93	3.1.3.2.7.1	Can documents be added to a restarted job?.....	54
94	3.1.4	Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)	55
95	3.1.5	Sending empty attribute groups.....	55
96	3.2	Printer Operations	55
97	3.2.1	Print-Job operation	55
98	3.2.1.1	Flow controlling the data portion of a Print-Job request (Issue 1.22).....	55
99	3.2.1.2	Returning job-state in Print-Job response (Issue 1.30)	56
100	3.2.2	Get-Printer-Attributes operation	56
101	3.2.3	Get-Jobs operation.....	57
102	3.2.3.1	Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?.....	57
103	3.2.3.2	Why is there a "limit" attribute in the Get-Jobs operation?.....	57
104	3.2.4	Create-Job operation.....	57
105	3.3	Job Operations	58
106	3.3.1	Validate-Job	58
107	3.3.2	Restart-Job	58
108	4	Object Attributes.....	58
109	4.1	Attribute Syntax's.....	58
110	4.1.1	The 'none' value for empty sets (Issue 1.37)	58
111	4.1.2	Multi-valued attributes (Issue 1.31).....	59
112	4.1.3	Case Sensitivity in URIs (issue 1.6).....	59
113	4.1.4	Maximum length for xxxWithLanguage and xxxWithoutLanguage.....	60
114	4.2	Job Template Attributes	60
115	4.2.1	multiple-document-handling(type2 keyword)	60
116	4.2.1.1	Support of multiple document jobs.....	60
117	4.3	Job Description Attributes.....	60
118	4.3.1	Getting the date and time of day.....	60
119	4.4	Printer Description Attributes.....	61
120	4.4.1	printer-state-reasons (1setOf type2 keyword).....	61
121	4.4.1.1	Is a suffix needed for the "printer-state-reasons" 'none' value (Issue 3.6)?.....	61
122	4.4.2	queued-job-count (integer(0:MAX))	61
123	4.4.2.1	Why is "queued-job-count" RECOMMENDED (Issue 1.14)?.....	61
124	4.4.2.2	Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?.....	61
125	4.4.3	printer-current-time (dateTime).....	61
126	4.4.4	Printer-uri.....	62
127	4.5	Empty Jobs	62
128	5	Directory Considerations.....	63
129	5.1	General Directory Schema Considerations	63
130	5.2	IPP Printer with a DNS name	63
131	6	Security Considerations.....	63

132	6.1	Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)	
133		63	
134	7	Encoding and Transport	64
135	7.1	General Headers.....	65
136	7.2	Request Headers	66
137	7.3	Response Headers.....	67
138	7.4	Entity Headers	67
139	7.5	Optional support for HTTP/1.0.....	68
140	7.6	HTTP/1.1 Chunking.....	68
141	7.6.1	Disabling IPP Server Response Chunking	68
142	7.6.2	Warning About the Support of Chunked Requests	69
143	7.7	HTTP "continue" interim response	69
144	7.8	How can an IPP client Provoke authentication challenges from IPP Printers	69
145	8	References (Informational)	73
146	9	Authors' Address.....	75
147	10	Description of the Base IPP Documents	78
148	11	Full Copyright Statement	79
149			
150	TABLES		
151			
152		Table 1 - Summary of Printer operation attributes that sender MUST supply	9
153		Table 2 - Summary of Printer operation attributes that sender MAY supply.....	10
154		Table 3 - Summary of Job operation attributes that sender MUST supply.....	11
155		Table 4 - Summary of Job operation attributes that sender MAY supply.....	12
156		Table 5 - Printer operation response attributes.....	13
157		Table 6 - Examples of validating IPP version.....	16
158		Table 7 - Rules for validating single values X against Z.....	33
159			
160			

160
161
162

163 **1 Introduction**

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

168 Design Goals for an Internet Printing Protocol [RFC2567]
169 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
170 Internet Printing Protocol/1.1: Model and Semantics [RFC2911]
171 Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
172 Internet Printing Protocol/1.1: Implementer's Guide (this document)
173 Mapping between LPD and IPP Protocols [RFC2569]

174

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

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

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

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

188 **1.1 Conformance language**

189 Usually, this document does not contain the terminology MUST, MUST NOT, MAY, NEED NOT,
190 SHOULD, SHOULD NOT, REQUIRED, and OPTIONAL. However, when those terms do appear in
191 this document, their intent is to repeat what the [RFC2911] and [RFC2910] documents require and
192 allow, rather than specifying additional conformance requirements. These terms are defined in section
193 13 on conformance terminology in [RFC2911], most of which is taken from RFC 2119 [RFC2119].

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

200 1.2 Other terminology

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

203 1.3 Issues Raised from Interoperability Testing Events

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

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

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

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

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

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

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

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

221 2 IPP Objects

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

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

229

229 **3 IPP Operations**

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

232 **3.1 Common Semantics**

233 This section discusses semantics common to all operations.

234 **3.1.1 Summary of Operation Attributes**

235 Legend for the following table:

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

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

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

242

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

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

243

244

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

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

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

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

245

246

246

247

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

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

248

249

249

250

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

Operation Attributes	Job Operations						
	Requests						Responses
	Send-Document (O)	Send-URI (O)	Cancel-Job (R)	Get-Job-Attributes (R)	Hold-Job, Restart-Job (O+)	Release-Job (O+)	All Operations
Operation attributes--OPTIONAL to be supplied by the sender:							
status-message							O
detailed-status-message							O
document-access-error							O**
compression	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							

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

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

251

252

252

253

Table 5 - Printer operation response attributes

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

254

255

255

256 3.1.2 Suggested Operation Processing Steps for IPP Objects

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

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

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

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

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

284

284 3.1.2.1 Suggested Operation Processing Steps for all Operations

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

288	IIG Sect #	Flow	IPP error status codes
289	-----	----	-----
290			
291		v	err
292	3.1.2.1.1	<Validate version>	--> server-error-version-not-
293			supported
294		ok	
295		v	err
296	3.1.2.1.2	<Validate operation>	--> server-error-operation-not-
297			supported
298		ok	
299		v	err
300	3.1.2.1.4.1-	<Validate presence>	--> client-error-bad-request
301	3.1.2.1.4.2	<of attributes>	
302		ok	
303		v	err
304	3.1.2.1.4.3	<Validate presence>	--> client-error-bad-request
305		<of operation attr>	
306		ok	
307		v	err
308	3.1.2.1.5	<Validate values of>	--> client-error-bad-request
309		<operation attrs>	client-error-request-value-
310			too-long
311		<(length, tag, range,>	
312		<multi-value)>	
313		ok	
314		v	err
315	3.1.2.1.5	<Validate values>	--> client-error-bad-request
316		<with supported values>	client-error-charset-not-
317			supported
318		ok	client-error-attributes-or-
319			values-
320			not-supported
321		v	err
322	3.1.2.1.6	<Validate optionally>	--> client-error-bad-request
323		<operation attr>	client-error-natural-language-
324			not-supported
325			client-error-request-value-
326			too-long
327			client-error-attributes-or-
328			values-not-supported
329			

330 3.1.2.1.1 Validate version number

331 Every request and every response contains the "version-number" attribute. The value of this attribute is
 332 the major and minor version number of the syntax and semantics that the client and IPP object is using,
 333 respectively. The "version-number" attribute remains in a fixed position across all future versions so
 334 that all clients and IPP object that support future versions can determine which version is being used.
 335 The IPP object checks to see if the major version number supplied in the request is supported. If not,
 336 the Printer object REJECTS the request and RETURNS the 'server-error-version-not-supported' status
 337 code in the response. The IPP object returns in the "version-number" response attribute the major and
 338 minor version for the error response. Thus the client can learn at least one major and minor version that
 339 the IPP object supports. The IPP object is encouraged to return the closest version number to the one
 340 supplied by the client.

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

348 **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

349

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

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

355 3.1.2.1.2 Validate operation identifier

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

359 **3.1.2.1.3 Validate the request identifier**

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

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

366 **3.1.2.1.4 Validate attribute group and attribute presence and order**

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

368 **3.1.2.1.4.1 Validate the presence and order of attribute groups**

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

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

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

381 **3.1.2.1.4.2 Ignore unknown attribute groups in the expected position**

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

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

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

394 **3.1.2.1.4.3 Validate the presence of a single occurrence of required Operation attributes**

395 Client requests and IPP object responses contain Operation attributes that [RFC2911] Section 3
396 requires to be present. Attributes within the Operation attributes group (Group 1) in a request may be
397 in any order, except for the ordering of target, charset, and natural languages attributes. These
398 attributes **MUST** be first, and **MUST** be supplied in the following order: charset, natural language, and
399 then target. An IPP object verifies that the attributes that Section 4 requires to be supplied by the client
400 have been supplied in the request (attributes without an * in the following tables). An asterisk (*)
401 indicates groups and Operation attributes that the client may omit in a request or an IPP object may
402 omit in a response.

403 If an IPP object receives a request with required attributes missing, repeated, or in the wrong position in
404 the Operation Attributes group (Group 1), the behavior of the IPP object is **IMPLEMENTATION**
405 **DEPENDENT**. Some of the possible implementations are:

- 406 REJECTS the request and RETURNS the 'client-error-bad-request' status code
- 407 accepts the request and uses the first occurrence of the attribute no matter where it is
- 408 accepts the request and uses the last occurrence of the attribute no matter where it is
- 409 accept the request and assume some default value for the missing attribute

410 Therefore, client **MUST** send conforming requests, if they want to receive the same behavior from all
411 IPP object implementations. For example, it is an error for the "attributes-charset" or "attributes-
412 natural-language" attribute to be omitted in any operation request to supply the "attributes-charset"
413 attribute twice.

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

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

- 423 R indicates a **REQUIRED** attribute or operation that an IPP object **MUST** support
- 424 O indicates an **OPTIONAL** attribute or operation that an IPP object **NEED NOT** support

- 425 * indicates that a client MAY omit the attribute in a request and that an IPP object MAY omit
426 the attribute in a response. The absence of an * means that a client MUST supply
427 the attribute in a request and an IPP object MUST supply the attribute in a
428 response.
429 + indicates that this is not a IPP/1.0 operation, but is only a part of IPP/1.1 and future versions
430 of IPP.

431

432 Operation Requests

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

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

436

437 Print-Job Request (R):

438 Group 1: Operation Attributes (R)
439 attributes-charset (R)
440 attributes-natural-language (R)
441 printer-uri (R)
442 requesting-user-name (R*)
443 job-name (R*)
444 ipp-attribute-fidelity (R*)
445 document-name (R*)
446 document-format (R*)
447 document-natural-language (O*)
448 compression (R*)
449 job-k-octets (O*)
450 job-impressions (O*)
451 job-media-sheets (O*)
452 Group 2: Job Template Attributes (R*)
453 <Job Template attributes> (O*)
454 (see [RFC2911] Section 4.2)
455 Group 3: Document Content (R)
456 <document content>

457

458 Validate-Job Request (R):

459 Group 1: Operation Attributes (R)
460 attributes-charset (R)
461 attributes-natural-language (R)
462 printer-uri (R)
463 requesting-user-name (R*)
464 job-name (R*)
465 ipp-attribute-fidelity (R*)
466 document-name (R*)
467 document-format (R*)
468 document-natural-language (O*)
469 compression (R*)
470 job-k-octets (O*)

471 job-impressions (O*)
472 job-media-sheets (O*)
473 Group 2: Job Template Attributes (R*)
474 <Job Template attributes> (O*)
475 (see [RFC2911] Section 4.2)
476
477 Print-URI Request (O):
478 Group 1: Operation Attributes (R)
479 attributes-charset (R)
480 attributes-natural-language (R)
481 printer-uri (R)
482 document-uri (R)
483 requesting-user-name (R*)
484 job-name (R*)
485 ipp-attribute-fidelity (R*)
486 document-name (R*)
487 document-format (R*)
488 document-natural-language (O*)
489 compression (R*)
490 job-k-octets (O*)
491 job-impressions (O*)
492 job-media-sheets (O*)
493 Group 2: Job Template Attributes (R*)
494 <Job Template attributes> (O*) (see
495 (see [RFC2911] Section 4.2)
496
497 Create-Job Request (O):
498 Group 1: Operation Attributes (R)
499 attributes-charset (R)
500 attributes-natural-language (R)
501 printer-uri (R)
502 requesting-user-name (R*)
503 job-name (R*)
504 ipp-attribute-fidelity (R*)
505 job-k-octets (O*)
506 job-impressions (O*)
507 job-media-sheets (O*)
508 Group 2: Job Template Attributes (R*)
509 <Job Template attributes> (O*) (see
510 (see [RFC2911] Section 4.2)
511
512 Get-Printer-Attributes Request (R):
513 Group 1: Operation Attributes (R)
514 attributes-charset (R)
515 attributes-natural-language (R)
516 printer-uri (R)
517 requesting-user-name (R*)
518 requested-attributes (R*)
519 document-format (R*)
520

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

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

595 Operation Responses

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

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

599
600 Print-Job Response (R):
601 Create-Job Response (O):
602 Send-Document Response (O):
603 Group 1: Operation Attributes (R)
604 attributes-charset (R)
605 attributes-natural-language (R)
606 status-message (O*)
607 detailed-status-message (O*)
608 Group 2: Unsupported Attributes (R*) (see Note 3)
609 <unsupported attributes> (R*)
610 Group 3: Job Object Attributes (R*) (see Note 2)
611 job-uri (R)
612 job-id (R)
613 job-state (R)
614 job-state-reasons (O* | R+)
615 job-state-message (O*)
616 number-of-intervening-jobs (O*)
617

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

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

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

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

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

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

701 **3.1.2.1.5 Validate the values of the REQUIRED Operation attributes**

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

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

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

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

714

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

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

727

728

attributes-charset (charset)

729

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

730

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

731

732

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

733

734

735

attributes-natural-language(naturalLanguage)

736

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

737

738

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

739

740

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

741

742

743

744

745

requesting-user-name

746 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
747 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
748 too-long'.
749 IF the IPP object can obtain a better-authenticated name, use it instead.
750

751 job-name(name)

752 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
753 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
754 too-long'.
755 IF NOT supplied by the client, the Printer object creates a name from the document-name or
756 document-uri.
757

758 document-name (name)

759 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
760 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
761 too-long'.
762

763 ipp-attribute-fidelity (boolean)

764 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-
765 error-bad-request'.
766 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-
767 long'.
768 IF NOT supplied by the client, the IPP object assumes the value 'false'.
769

770 document-format (mimeType)

771 IF NOT a single non-empty 'mimeType' value, REJECT/RETURN 'client-error-bad-
772 request'.
773 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
774 too-long'.
775 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN
776 'client-error-document-format-not-supported'.
777 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's
778 "document-format-default" attribute.
779

780 document-uri (uri)

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

785 If the client-supplied URI scheme is not supported, i.e. the value is not in the Printer object's
786 referenced-uri-scheme-supported" attribute, the Printer object MUST reject the request
787 and return the 'client-error-uri-scheme-not-supported' status code. The Printer object
788 MAY check to see if the document exists and is accessible. If the document is not found
789 or is not accessible, REJECT/RETURN 'client-error-not found'.
790 last-document (boolean)
791 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-
792 error-bad-request'.
793 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-
794 long'
795
796 job-id (integer(1:MAX))
797 IF NOT an single 'integer' value equal to 4 octets AND in the range 1 to MAX,
798 REJECT/RETURN 'client-error-bad-request'.
799 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-
800 error-gone' status code, if keep track of recently deleted jobs.
801
802 requested-attributes (1setOf keyword)
803 IF NOT one or more 'keyword' values, REJECT/RETURN 'client-error-bad-request'.
804 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
805 too-long'.
806 Ignore unsupported values, which are the keyword names of unsupported attributes. Don't
807 bother to copy such requested (unsupported) attributes to the Unsupported Attribute
808 response group since the response will not return them.
809
810 which-jobs (type2 keyword)
811 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
812 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
813 too-long'.
814 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to
815 the Unsupported Attributes response group and REJECT/RETURN 'client-error-
816 attributes-or-values-not-supported'.
817 Note: a Printer still supports the 'completed' value even if it keeps no
818 completed/canceled/aborted jobs: by returning no jobs when so queried.
819 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.
820
821 my-jobs (boolean)
822 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-
823 error-bad-request'.
824 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-
825 long'.
826 IF NOT supplied by the client, the IPP object assumes the 'false' value.

827

828 limit (integer(1:MAX))

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

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

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

832

833 -----

834

835 **3.1.2.1.6 Validate the values of the OPTIONAL Operation attributes**

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

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

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

848 -----

849 document-natural-language (naturalLanguage)

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

851 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-
852 long'.853 IF NOT a value that the Printer object supports in document formats, (no corresponding "xxx-
854 supported" Printer attribute), REJECT/RETURN 'client-error-natural-language-not-
855 supported'.

856

857 compression (type3 keyword)

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

859 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
860 long'.861 IF NOT in the Printer object's "compression-supported" attribute, REJECT/RETURN 'client-error-
862 compression-not-supported'.

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

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

869 IF NOT a single 'integer' value equal to 4 octets,
870 REJECT/RETURN 'client-error-bad-request'.
871 IF NOT in the range of the Printer object's "job-k-octets-supported" attribute, copy the attribute and
872 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN
873 'client-error-attributes-or-values-not-supported'.
874

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

876 IF NOT a single 'integer' value equal to 4 octets,
877 REJECT/RETURN 'client-error-bad-request'.
878 IF NOT in the range of the Printer object's "job-impressions-supported" attribute, copy the attribute
879 and the unsupported value to the Unsupported Attributes response group and
880 REJECT/RETURN 'client-error-attributes-or-values-not-supported'.
881

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

883 IF NOT a single 'integer' value equal to 4 octets,
884 REJECT/RETURN 'client-error-bad-request'.
885 IF NOT in the range of the Printer object's "job-media-sheets-supported" attribute, copy the attribute
886 and the unsupported value to the Unsupported Attributes response group and
887 REJECT/RETURN 'client-error-attributes-or-values-not-supported'.
888

889 message (text(127))

890 IF NOT a single 'text' value, REJECT/RETURN 'client-error-bad-request'.
891 IF the value length is greater than 127 octets,
892 REJECT/RETURN 'client-error-request-value-too-long'.
893

894 unknown or unsupported attribute

895 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute
896 syntax, REJECT/RETURN 'client-error-request-value-too-long'.
897 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
898 attribute value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.
899

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

915

915 **3.1.2.2 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add**
 916 **Documents**

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

921	IIG Sect #	Flow	IPP error status codes
922	-----	----	-----
923			
924			
925		v	No
926	3.1.2.2.1	<ipp-attribute-fidelity>	-----+-----
927		<supplied?>	
928		Yes	
929			
930			
931			
932			
933			
934			
935			
936			
937			
938			
939			
940			
941			
942			
943			
944			
945			
946			
947			
948			
949			
950			
951			
		v	
	3.1.2.2.2	<Printer is	--> server-error-not-accepting-jobs
		<accepting jobs?>	
		Yes	
		v	err
	3.1.2.3	<Validate values of>	--> client-error-bad-request
		<Job template attributes>	client-error-request-value-too-
			long
		<(length, tag, range,>	
		<multi-value)>	
		ok	
		v	err
	3.1.2.3	<Validate values with>	--> client-error-bad-request
		<supported values>	client-error-attributes-or-
			values-not-supported
		v	err
	3.1.2.3.1	<Any conflicting>	--> client-error-conflicting-
			attributes
		<Job Template attr values>	client-error-attributes-or-
			values-not-supported

952 **3.1.2.2.1 Default "ipp-attribute-fidelity" if not supplied**

953 The Printer object checks to see if the client supplied an "ipp-attribute-fidelity" Operation attribute in
 954 the Operation Attribute group (group 1) in the request. If the attribute is not supplied by the client, the
 955 IPP object assumes that the value is 'false'.

956 **3.1.2.2 Check that the Printer object is accepting jobs**

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

959 **3.1.2.3 Validate the values of the Job Template attributes**

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

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

965 b) that the attribute syntax tag is correct for that attribute according to [RFC2911] Sections
966 4.2 to 4.4.

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

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

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

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

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

983 **3.1.2.3 Algorithm for job validation**

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

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

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

002 **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.

003

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

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

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

022 -----

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

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

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

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

030 Map the value to the nearest supported value in the range 1:100 as specified by the number of
031 discrete values indicated by the value of the Printer's "job-priority-supported" attribute. See
032 the formula in [RFC2911] Section 4.2.1.
033

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

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

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

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

040 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the
041 unsupported value to the Unsupported Attributes response group.
042

043 job-sheets (type3 keyword | name)

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

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

047 IF NOT in the Printer object's "job-sheets-supported" attribute, copy the attribute and the
048 unsupported value to the Unsupported Attributes response group.
049

050 multiple-document-handling (type2 keyword)

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

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

054 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute
055 and the unsupported value to the Unsupported Attributes response group.
056

057 copies (integer(1:MAX))

058 IF NOT a single 'integer' value with a length equal to 4 octets,
059 REJECT/RETURN 'client-error-bad-request'.
060 IF NOT in range of the Printer object's "copies-supported" attribute
061 copy the attribute and the unsupported value to the Unsupported Attributes response group.
062

063 **finishings** (1setOf type2 enum)

064 IF NOT an 'enum' value(s) each with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
065 request'.
066 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the
067 unsupported value(s), but not any supported values, to the Unsupported Attributes response
068 group.
069

070 **page-ranges** (1setOf rangeOfInteger(1:MAX))

071 IF NOT a 'rangeOfInteger' value(s) each with a length equal to 8 octets, REJECT/RETURN 'client-
072 error-bad-request'.
073 IF first value is greater than second value in any range, the ranges are not in ascending order, or
074 ranges overlap, REJECT/RETURN 'client-error-bad-request'.
075 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to
076 the Unsupported Attributes response group and set the value to the "out-of-band"
077 'unsupported' value.
078

079 **sides** (type2 keyword)

080 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
081 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
082 long'.
083 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported
084 value to the Unsupported Attributes response group.
085

086 **number-up** (integer(1:MAX))

087 IF NOT a single 'integer' value with a length equal to 4 octets,
088 REJECT/RETURN 'client-error-bad-request'.
089 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"
090 attribute, copy the attribute and value to the Unsupported Attribute response group.
091

092 **orientation-requested** (type2 enum)

093 IF NOT a single 'enum' value with a length equal to 4 octets,
094 REJECT/RETURN 'client-error-bad-request'.
095 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the
096 unsupported value to the Unsupported Attributes response group.
097

098 **media** (type3 keyword | name)

099 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
 100 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
 101 long'.
 102 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported
 103 value to the Unsupported Attributes response group.
 104

105 printer-resolution (resolution)

106 IF NOT a single 'resolution' value with a length equal to 9 octets,
 107 REJECT/RETURN 'client-error-bad-request'.
 108 IF NOT in the Printer object's "printer-resolution-supported" attribute, copy the attribute and the
 109 unsupported value to the Unsupported Attributes response group.
 110

111 print-quality (type2 enum)

112 IF NOT a single 'enum' value with a length equal to 4 octets,
 113 REJECT/RETURN 'client-error-bad-request'.
 114 IF NOT in the Printer object's "print-quality-supported" attribute, copy the attribute and the
 115 unsupported value to the Unsupported Attributes response group.
 116

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

119 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute
 120 syntax,
 121 REJECT/RETURN 'client-error-bad-request' if the length of the attribute syntax is fixed or 'client-
 122 error-request-value-too-long' if the length of the attribute syntax is variable.
 123 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
 124 attribute value to the "out-of-band" 'unsupported' value. Any remaining Job Template
 125 Attributes are either unknown or unsupported Job Template attributes and are validated
 126 algorithmically according to their attribute syntax for proper length (see below).
 127 -----

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

Name	Octet length check for read-write attributes
'textWithLanguage	<= 1023 AND 'naturalLanguage' <= 63
'textWithoutLanguage'	<= 1023
'nameWithLanguage'	<= 255 AND 'naturalLanguage' <= 63
'nameWithoutLanguage'	<= 255

```

141      'keyword'          <= 255
142      'enum'            = 4
143      'uri'             <= 1023
144      'uriScheme'       <= 63
145      'charset'         <= 63
146      'naturalLanguage' <= 63
147      'mimeType'        <= 255
148      'octetString'     <= 1023
149      'boolean'         = 1
150      'integer'         = 4
151      'rangeOfInteger'  = 8
152      'dateTime'       = 11
153      'resolution'     = 9
154      'lsetOf X'
155

```

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

161 3.1.2.3.1 Check for conflicting Job Template attributes values

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

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

175 3.1.2.3.2 Decide whether to REJECT the request

176 If there were any unsupported Job Template attributes or unsupported/conflicting Job Template
 177 attribute values and the client supplied the "ipp-attribute-fidelity" Operation attribute with the 'true'
 178 value in the Operation Attributes group (Group 1) in the request, the Printer object REJECTS the
 179 request and return the status code:

- 180 1. 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes
181 supplied by the client.
182 2. 'client-error-attributes-or-values-not-supported' status code, otherwise.
183

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

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

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

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

- 195 – the Printer supports auto-sensing,
196 – the request "document-format" operation attribute is 'application/octet-stream',
197 – the Printer receives document data before responding,
198 – the Printer auto-senses the document format before responding,
199 – the sensed document format is not supported by the Printer
200 then the Printer should respond with 'client-error-document-format-not-supported' status.

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

- 203 – the client supplies a supported value for the "compression" operation attribute in the request
204 – the Printer receives document data before responding,
205 – the Printer attempts to decompress the document data before responding,
206 – the document data cannot be decompressed using the algorithm specified by the "compression"
207 operation attribute
208 then the Printer should respond with 'client-error-compression-error' status.

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

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

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

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

224 **3.1.2.3.3 For the Validate-Job operation, RETURN one of the success status codes**

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

- 226 1. the "successful-ok" status code, if there are no unsupported or conflicting Job Template
227 attributes or values.
- 228 2. the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or
229 values.
- 230 3. the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template
231 attributes or values.

232

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

237 **3.1.2.3.4 Create the Job object with attributes to support**

238 If the "ipp-attribute-fidelity" Operation attribute is set to 'false' (or it was not supplied by the client in
239 the Operation Attributes group), the Printer object:

- 240 1. creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and
241 initializes all of the job's other supported Job Description attributes.
- 242 2. removes all unsupported attributes from the Job object.
- 243 3. for each unsupported value, removes either the unsupported value or substitutes the
244 unsupported attribute value with some supported value. If an attribute has no values after
245 removing unsupported values from it, the attribute is removed from the Job object (so that the
246 normal default behavior at job processing time will take place for that attribute).
- 247 4. for each conflicting value, removes either the conflicting value or substitutes the conflicting
248 attribute value with some other supported value. If an attribute has no values after removing
249 conflicting values from it, the attribute is removed from the Job object (so that the normal
250 default behavior at job processing time will take place for that attribute).

251

252 If there were no attributes or values flagged as unsupported, or the value of the 'ipp-attribute-fidelity'
253 Operation attribute was 'false', the Printer object is able to accept the create request and create a new
254 Job object. If the "ipp-attribute-fidelity" Operation attribute is set to 'true', the Job Template attributes
255 that populate the new Job object are necessarily all the Job Template attributes supplied in the create
256 request. If the "ipp-attribute-fidelity" Operation attribute is set to 'false', the Job Template attributes
257 that populate the new Job object are all the client supplied Job Template attributes that are supported or
258 that have value substitution. Thus, some of the requested Job Template attributes may not appear in the
259 Job object because the Printer object did not support those attributes. The attributes that populate the
260 Job object are persistently stored with the Job object for that Job. A Get-Job-Attributes operation on
261 that Job object will return only those attributes that are persistently stored with the Job object.

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

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

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

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

283 3.1.2.3.5 Return one of the success status codes

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

- 285 1. the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes
286 or values.
- 287 2. the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template
288 attribute or values.

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

291

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

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

298 **3.1.2.3.6 Accept appended Document Content**

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

301 **3.1.2.3.7 Scheduling and Starting to Process the Job**

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

307 **3.1.2.3.8 Completing the Job**

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

313 **3.1.2.3.9 Destroying the Job after completion**

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

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

321 **3.1.2.3.10 Interaction with "ipp-attribute-fidelity"**

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

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

333 **3.1.2.3.11 Character set code conversion support**

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

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

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

346 According to the RFC2911 document (section 3.1.4.2), the value of the "attributes-charset" for the
347 response of the second request must be "us-ascii" since that is the charset specified in the request. The
348 "job-name" value, however, is in "utf-8" format. Should the request be rejected even though both "utf-
349 8" and "us-ascii" charsets are supported by the server? or should the "job-name" value be converted to
350 "us-ascii" and return "successful-ok-conflicting-attributes" (0x0002) as the status code?

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

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

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

364 **3.1.2.3.12 What charset to return when an unsupported charset is requested (Issue 1.19)?**

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

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

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

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

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

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

382 3.1.2.3.13 Natural Language Override (NLO)

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

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

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

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

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

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

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

426 3.1.3 Status codes returned by operation

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

430 3.1.3.1 Printer Operations

431 3.1.3.1.1 Print-Job

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

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

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

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

443 successful-ok-conflicting-attributes: some supplied attribute values conflicted with the values of
444 other supplied attributes and were either substituted or ignored. Attributes or values which
445 conflict with other attributes and have been substituted or ignored MUST be returned in the
446 Unsupported Attributes group of the response as supplied by the client.
447

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

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

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

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

457 client-error-forbidden: The request is being refused for authorization or authentication reasons.

458 The implementation security policy is to not reveal whether the failure is one of

459 authentication or authorization.

460 client-error-not-authenticated: Either the request requires authentication information to be

461 supplied or the authentication information is not sufficient for authorization.

462 client-error-not-authorized: The requester is not authorized to perform the request on the target

463 object.

464 client-error-not-possible: The request cannot be carried out because of the state of the system.

465 See also 'server-error-not-accepting-jobs' status code, which **MUST** take precedence if the

466 Printer object's "printer-accepting-jobs" attribute is 'false'.

467 client-error-timeout: not applicable.

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

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

470 client-error-request-entity-too-large: the size of the request and/or print data exceeds the

471 capacity of the IPP Printer to process it.

472 client-error-request-value-too-long: the size of request variable length attribute values, such as

473 'text' and 'name' attribute syntax's, exceed the maximum length specified in [RFC2911] for the

474 attribute and **MUST** be returned in the Unsupported Attributes Group.

475 client-error-document-format-not-supported: the document format supplied is not supported.

476 The "document-format" attribute with the unsupported value **MUST** be returned in the

477 Unsupported Attributes Group. This error **SHOULD** take precedence over any other 'xxx-

478 not-supported' error, except 'client-error-charset-not-supported'.

479 client-error-attributes-or-values-not-supported: one or more supplied attributes, attribute

480 syntax's, or values are not supported and the client supplied the "ipp-attributes-fidelity"

481 operation attribute with a 'true' value. They **MUST** be returned in the Unsupported

482 Attributes Group as explained below.

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

484 client-error-charset-not-supported: the charset supplied in the "attributes-charset" operation

485 attribute is not supported. The Printer's "configured-charset" **MUST** be returned in the

486 response as the value of the "attributes-charset" operation attribute and used for any 'text' and

487 'name' attributes returned in the error response. This error **SHOULD** take precedence over

488 any other error, unless the request syntax is so bad that the client's supplied "attributes-

489 charset" cannot be determined.

490 client-error-conflicting-attributes: one or more supplied attribute values conflicted with each

491 other and the client supplied the "ipp-attributes-fidelity" operation attribute with a 'true'

492 value. They **MUST** be returned in the Unsupported Attributes Group as explained below.

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

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

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

496 server-error-version-not-supported: the version in the request is not supported. The "closest"

497 version number supported **MUST** be returned in the response.

498 server-error-device-error: a device error occurred while receiving or spooling the request or

499 document data or the IPP Printer object can only accept one job at a time.

500 server-error-temporary-error: a temporary error such as a buffer full write error, a memory
501 overflow, or a disk full condition occurred while receiving the request and/or the document
502 data.
503 server-error-not-accepting-jobs: the Printer object's "printer-is-not-accepting-jobs" attribute is
504 'false'.
505 server-error-busy: the Printer is too busy processing jobs to accept another job at this time.
506 server-error-job-canceled: the job has been canceled by an operator or the system while the
507 client was transmitting the document data.

508 3.1.3.1.2 Print-URI

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

512 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation
513 attribute is not supported and is returned in the Unsupported Attributes group.
514 server-error-operation-not-supported: the Print-URI operation is not supported.
515

516 3.1.3.1.3 Validate-Job

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

519 3.1.3.1.4 Create-Job

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

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

527 3.1.3.1.5 Get-Printer-Attributes

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

531 For the following success status codes, the requested attributes are returned in Group 3 in the response:
532 successful-ok: no operation attributes or values were substituted or ignored (same as Print-Job)and
533 no requested attributes were unsupported.

534 *Note to client implementers: If the client requests attributes that are not supported by the*
 535 *Printer, the Printer is supposed to return 'successful-ok-ignored-or-substituted-attributes',*
 536 *rather than 'successful-ok'. However, a number of implementations have been found not to*
 537 *conform to this requirement, so clients should be tolerant of such Printers.*

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

540 *Note to client implementers: Although NOT RECOMMENDED, the Unsupported Attribute*
 541 *Group and its contents MAY be omitted. Clients SHOULD be prepared for this behavior.*

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

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

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

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

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

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

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

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

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

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

555 server-error-busy: same as Print-Job, except the IPP object is too busy to accept even query
 556 requests.

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

558 3.1.3.1.6 Get-Jobs

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

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

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

564 successful-ok-ignored-or-substituted-attributes: same as Get-Printer-Attributes (see section
 565 3.1.3.1.5).

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

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

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

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

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

574 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation
575 attributes and/or values MUST be ignored and an appropriate success code returned (see
576 above).
577 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not
578 involved.
579 server-error-operation-not-supported: not applicable (since Get-Jobs is REQUIRED).
580 server-error-device-error: same as Print-Job, except that no document data is involved.
581 server-error-temporary-error: same as Print-Job, except that no document data is involved.
582 server-error-not-accepting-jobs: not applicable.
583 server-error-job-canceled: not applicable.

584 **3.1.3.1.7 Pause-Printer**

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

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

590 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
591 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
592 successful-ok-conflicting-attributes: same as Print-Job.
593

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

596 client-error-not-possible: not applicable.
597 client-error-not-found: the target Printer object does not exist.
598 client-error-gone: the target Printer object no longer exists and no forwarding address is known.
599 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
600 client-error-document-format-not-supported: not applicable.
601 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-
602 accepting-jobs" attribute is not involved.
603 server-error-operation-not-supported: the Pause-Printer operation is not supported.
604 server-error-device-error: not applicable.
605 server-error-temporary-error: same as Print-Job, except no document data is involved.
606 server-error-not-accepting-jobs: not applicable.
607 server-error-job-canceled: not applicable.

608 **3.1.3.1.8 Resume-Printer**

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

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

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

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

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

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

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

618

619 **3.1.3.1.8.1 What about Printers unable to change state due to an error condition?**

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

624 The Resume-Printer operation must clear the 'paused' or 'moving-to-paused' 'printer-state-message'.

625 The operation must return a 'successful-ok' status code.

626 **3.1.3.1.8.2 How is "printer-state" handled on Resume-Printer?**

627

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

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

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

632 1. 'idle' - no additional jobs and no error conditions present

633 2. 'processing' - job available and no error conditions present

634 3. current state (i.e. no change) an error condition is present (e.g. media jam)

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

638 **3.1.3.1.9 Purge-Printer**

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

642 For the following success status codes, the Printer object purges all it's jobs.
643 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
644 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
645 successful-ok-conflicting-attributes: same as Print-Job.

646 For any of the error status codes, the Printer object does not purge any jobs.
647 server-error-operation-not-supported: the Purge-Printer operation is not supported.

648 **3.1.3.2 Job Operations**

649 **3.1.3.2.1 Send-Document**

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

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

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

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

660 client-error-not-possible: Same as Print-Job, except that the Printer's "printer-is-accepting-jobs"
661 attribute is not involved, so that the client is able to finish submitting a job that was created
662 with a Create-Job operation after this attribute has been set to 'true'. Another condition is
663 that the state of the job precludes Send-Document, i.e., the job has already been closed out
664 by the client. However, if the IPP Printer closed out the job due to timeout, the 'client-error-
665 timeout' error status SHOULD be returned instead.
666 client-error-timeout: This request was sent after the Printer closed the job, because it has not
667 received a Send-Document or Send-URI operation within the Printer's "multiple-operation-
668 time-out" period .
669 client-error-request-entity-too-large: same as Print-Job.
670 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attributes-fidelity"
671 operation attribute is not involved..
672 server-error-operation-not-supported: the Send-Document request is not supported.
673 server-error-not-accepting-jobs: not applicable.
674 server-error-job-canceled: the job has been canceled by an operator or the system while the
675 client was transmitting the data.

676 3.1.3.2.2 Send-URI

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

680 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri"
681 operation attribute is not supported and the "document-uri" attribute MUST be returned in
682 the Unsupported Attributes group.
683 server-error-operation-not-supported: the Send-URI operation is not supported.
684

685 3.1.3.2.3 Cancel-Job

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

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

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

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

695 client-error-not-possible: The request cannot be carried out because of the state of the Job
696 object ('completed', 'canceled', or 'aborted') or the state of the system.
697 client-error-not-found: the target Printer and/or Job object does not exist.
698 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding
699 address is known.
700 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
701 client-error-document-format-not-supported: not applicable.
702 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation
703 attributes and values MUST be ignored.
704 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-
705 accepting-jobs" attribute is not involved.
706 server-error-operation-not-supported: not applicable (Cancel-Job is REQUIRED).
707 server-error-device-error: same as Print-Job, except no document data is involved.
708 server-error-temporary-error: same as Print-Job, except no document data is involved.
709 server-error-not-accepting-jobs: not applicable..
710 server-error-job-canceled: not applicable.

711 3.1.3.2.4 Get-Job-Attributes

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

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

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

717 successful-ok-ignored-or-substituted-attributes: same as Get-Printer-Attributes (see section
718 3.1.3.1.5).

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

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

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

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

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

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

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

729 client-error-conflicting-attributes: not applicable

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

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

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

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

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

735 3.1.3.2.5 Hold-Job

736 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to
737 Hold-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 held or has been held:

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 held or was previously held.

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-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-
753 accepting-jobs" attribute is not involved.
754 server-error-operation-not-supported: the Hold-Job operation is not supported.
755 server-error-device-error: not applicable.
756 server-error-temporary-error: same as Print-Job, except no document data is involved.
757 server-error-not-accepting-jobs: not applicable.
758 server-error-job-canceled: not applicable.

759 3.1.3.2.6 Release-Job

760 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
761 specialization's described for Hold-Job are applicable to Release-Job. See Section 13 in [RFC2911] for
762 a more complete description of each status code.
763 server-error-operation-not-supported: the Release-Job operation is not supported.

764 3.1.3.2.7 Restart-Job

765 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
766 specialization's described for Hold-Job are applicable to Restart-Job. See Section 13 in [RFC2911] for
767 a more complete description of each status code.
768 server-error-operation-not-supported: the Restart-Job operation is not supported.
769

770 3.1.3.2.7.1 Can documents be added to a restarted job?

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

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

780 **3.1.4 Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)**

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

788 **3.1.5 Sending empty attribute groups**

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

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

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

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

806 **3.2 Printer Operations**

807 **3.2.1 Print-Job operation**

808 **3.2.1.1 Flow controlling the data portion of a Print-Job request (Issue 1.22)**

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

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

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

818 **3.2.1.2 Returning job-state in Print-Job response (Issue 1.30)**

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

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

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

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

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

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

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

841 **3.2.2 Get-Printer-Attributes operation**

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

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

847 3.2.3 Get-Jobs operation

848 3.2.3.1 Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?

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

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

855 3.2.3.2 Why is there a "limit" attribute in the Get-Jobs operation?

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

865 3.2.4 Create-Job operation

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

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

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

882 3.3 Job Operations

883 3.3.1 Validate-Job

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

888 3.3.2 Restart-Job

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

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

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

898 4 Object Attributes

899 4.1 Attribute Syntax's

900 4.1.1 The 'none' value for empty sets (Issue 1.37)

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

909 4.1.2 Multi-valued attributes (Issue 1.31)

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

916 4.1.3 Case Sensitivity in URIs (issue 1.6)

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

923 The reason that the IPP specification does not make any restrictions on URIs, is so that implementations
924 of IPP may use off-the-shelf components that conform to the standards that define URIs, such as RFC
925 2396 and the HTTP/1.1 specifications [RFC2616]. See these specifications for rules of matching,
926 comparison, and case-sensitivity.

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

930 Example of equivalent URI's

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

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

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

934 Example of equivalent URI's using the IPP scheme

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

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

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

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

939 **4.1.4 Maximum length for xxxWithLanguage and xxxWithoutLanguage**

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

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

947 **4.2 Job Template Attributes**

948 **4.2.1 multiple-document-handling(type2 keyword)**

949 **4.2.1.1 Support of multiple document jobs**

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

958 **4.3 Job Description Attributes**

959 **4.3.1 Getting the date and time of day**

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

- 963 1. A Printer can get time from an NTP timeserver if there's one reachable on the network . See
964 RFC 1305. Also DHCP option 32 in RFC 2132 returns the IP address of the NTP server.
- 965 2. Get the date and time at startup from a human operator
- 966 3. Have an operator set the date and time using a web administrative interface
- 967 4. Get the date and time from incoming HTTP requests, though the problems of spoofing need
968 to be considered. Perhaps comparing several HTTP requests could reduce the chances of spoofing.
- 969 5. Internal date time clock battery driven.

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

971 4.4 Printer Description Attributes

972 4.4.1 printer-state-reasons (1setOf type2 keyword)

973 4.4.1.1 Is a suffix needed for the "printer-state-reasons" 'none' value (Issue 3.6)?

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

980 4.4.2 queued-job-count (integer(0:MAX))

981 4.4.2.1 Why is "queued-job-count" RECOMMENDED (Issue 1.14)?

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

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

988 4.4.2.2 Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?

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

993 4.4.3 printer-current-time (dateTime)

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

- 996 1. an internal date time clock,
- 997 2. from the operator at startup using the console,
- 998 3. from an operator using an administrative web page,

- 999 4. from HTTP headers supplied in client requests,
000 5. use HTTP to query "<http://tycho.usno.navy.mil/cgi-bin/timer.pl>"
001 6. from the network, using NTP [RFC1305] or DHCP option 32 [RFC2132] that returns the IP
002 address of the NTP server.

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

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

008 **4.4.4 Printer-uri**

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

010 A forgiving printer implementation would not reject the operation. But the implementation has its rights
011 to reject a printer or job operation if the operational attribute printer-uri is not a value of the printer-uri-
012 supported. The printer may not be improperly configured. The request obviously reached the printer.
013 The printer could treat the printer-uri as the logical equivalent of a value in the printer-uri-supported. It
014 would be implementation dependent for which value, and associated security policy, would apply. This
015 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
016 for how to compare URI's.

017 **4.5 Empty Jobs**

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

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

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

029 **5 Directory Considerations**

030 **5.1 General Directory Schema Considerations**

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

033 The SLP printer template is defined in the "Definition of the Printer Abstract Service Type v2.0"
034 document [svrloc-printer] as used with SLPv2 [RFC2608, RFC2609, RFC2926]. The LDAP printer
035 schema is defined in the "Internet Printing Protocol (IPP): LDAP Schema for Printer Services"
036 document [ldap-printer] as used with LDAPv3 [RFC2251, RFC2252]. Both documents systematically
037 add "printer-" to any attribute that doesn't already start with "printer-" in order to keep the printer
038 directory attributes distinct from other directory attributes. Also, instead of using "printer-uri-
039 supported", "uri-authentication-supported", and "uri-security-supported", they use a "printer-xri-
040 supported" attribute with special syntax to contain all of the same information in a single attribute. The
041 "printer-xri-supported" (1setOf collection) Printer Description attribute is also defined as an IPP
042 extension for use with the Set-Printer-Attributes operation [ipp-set-ops].

043 **5.2 IPP Printer with a DNS name**

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

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

047 **6 Security Considerations**

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

051 **6.1 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)**

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

053 8.5 Queries on jobs submitted using non-IPP protocols

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

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

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

072 **7 Encoding and Transport**

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

074 A server is not required to send a response until after it has received the client's entire request. Hence, a
075 client need not expect a response until after it has sent the entire request. An exception to this statement
076 is the case of the client specifying the "Expect: 100-continue" header. See section 7.7.

077 We recommend that the server return a response as soon as possible if an error is detected while the
078 client is still sending the data, rather than waiting until all of the data is received. Therefore, we also
079 recommend that a client listen for an error response that an IPP server MAY send before it receives all
080 the data. In this case a client, if chunking the data, can send a premature zero-length chunk to end the
081 request before sending all the data (and so the client can keep the connection open for other requests,
082 rather than closing it). If the request is blocked for some reason, a client MAY determine the reason by
083 opening another connection to query the server using Get-Printer-Attributes.

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

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

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

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

- 102 – the "header" column contains the name of a header
- 103 – the "request/client" column indicates whether a client sends the header.
- 104 – the "request/ server" column indicates whether a server supports the header when received.
- 105 – the "response/ server" column indicates whether a server sends the header.
- 106 – the "response /client" column indicates whether a client supports the header when received.
- 107 – the "values and conditions" column specifies the allowed header values and the conditions for
108 the header to be present in a request/response.

109

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

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

- 113 – **must:** the client or server **MUST** send the header,
- 114 – **must-if:** the client or server **MUST** send the header when the condition described in the "values
115 and conditions" column is met,
- 116 – **may:** the client or server **MAY** send the header
- 117 – **not:** the client or server **SHOULD NOT** send the header. It is not relevant to an IPP
118 implementation.

119

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

- 121 – **must:** the client or server **MUST** support the header,
- 122 – **may:** the client or server **MAY** support the header
- 123 – **not:** the client or server **SHOULD NOT** support the header. It is not relevant to an IPP
124 implementation.

125 7.1 General Headers

126 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	

127 **7.2 Request Headers**

128 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	

129 7.3 Response Headers

130 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.

131 7.4 Entity Headers

132 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	

133 7.5 Optional support for HTTP/1.0

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

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

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

145 7.6 HTTP/1.1 Chunking

146 7.6.1 Disabling IPP Server Response Chunking

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

151 TE: identity

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

154 **7.6.2 Warning About the Support of Chunked Requests**

155 This section describes some problems with the use of chunked requests and HTTP/1.1 servers. For
156 additional known problems with implementations of HTTP proxies and caching, see "Known HTTP
157 Proxy/Caching Problems" [RFC3143].

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

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

170 **7.7 HTTP "continue" interim response**

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

173 The specific HTTP client and server requirements for '100 Continue' are laid out in section 8.2.3, "Use
174 of the 100 (Continue) Status", in [RFC2616]. Section 7.8 summarizes the HTTP requirements and
175 provides IPP implementation guidance related to the 100-Continue mechanism and its use.
176

177 **7.8 How can an IPP client Provoke authentication challenges from IPP Printers**

178 The IPP operation 'Validate-Job' was created to allow clients to confirm that an identical 'Print-Job'
179 operation (with the document data) would be accepted. The 'Validate-Job' also performs the same
180 security negotiation as the 'Print-Job' operation. This allows a client to verify that the security
181 requirements can be met and the job template attributes honored before sending any document data.
182 Due to the nature of HTTP connection management there is no guarantee that the client will not be
183 required to re-authenticate on the following operation. Clients that wish to provoke an IPP Printer to
184 issue an authentication challenge prior to sending an IPP operation have the ability to do so.

185 In some cases, a request may be rejected on the basis of the HTTP header alone. (Here, the HTTP
186 "header" includes the HTTP request-line, the HTTP header fields, and the terminating double CRLF.)
187 This is likely to be the case when the requested resource is protected by Digest Authentication: the
188 client needs the "nonce" value from the Printer's challenge in order to form a proper Authorization
189 header field value. In these cases, a client may wish to avoid transmitting the HTTP request body
190 containing the IPP request. For one thing, transmitting a large document for a request, only to have that
191 request rejected on the basis of the HTTP header alone, would be a waste of time and network
192 resources. For another, some clients, especially those transmitting dynamically generated content may
193 find it difficult, inefficient, or even impossible to tell the content generator to back up and regenerate the
194 content from the beginning. The HTTP 100-continue mechanism provides a solution to this problem.
195 The purpose of the 100-continue status is to allow a client that is sending a message with a request body
196 to determine if the Printer is willing to accept the request (based on the HTTP request header) before
197 the client sends the request body.

198 Here is a summary of the rules for HTTP 100-continue:

- 199 • If a client will wait for a 100 (Continue) response before sending the request body, it **MUST**
200 send an "Expect: 100-continue" header field.
- 201 • If an HTTP request contains an "Expect: 100-continue" header field, the Printer **MUST**
202 either respond with 100 (Continue) status and continue to read from the input stream, or
203 reject the request with a final HTTP status code.
- 204 • The Printer **MUST NOT** wait for the request body before sending the 100 (Continue)
205 response.
- 206 • If the Printer responds with a final status code instead of 100 (Continue), it **MAY** close the
207 connection (preferably, only the Printer's input side of the connection) or it **MAY** continue to
208 read and discard the rest of the response. It **MUST NOT** perform the requested method.
- 209 • A Printer **SHOULD NOT** send a 100 (Continue) response if the request does not include
210 "Expect: 100-continue".
- 211 • A Printer **MUST NOT** send a 100 (Continue) response to an HTTP/1.0 request. - A
212 Printer **MAY** omit a 100 (Continue) response if it has already received some of the request
213 body for the corresponding request.
- 214 • A Printer that sends a 100 (Continue) response **MUST** ultimately send a final status code,
215 once the request body is received and processed, unless it terminates the transport connection
216 prematurely.

217 Some finer points:

- 218 • A client waiting for a 100 (Continue) response **SHOULD NOT** wait for an indefinite period
219 before sending the request body
- 220 • A client **SHOULD** ignore any unexpected 100 (Continue) responses.

221

222 The basic algorithm is this:

- 223 1. The client sends an HTTP request header containing the "Expect: 100-continue" header field, but
224 waits before transmitting the request body.

- 225 2. The Printer examines the HTTP header and decides whether or not to accept the HTTP request.
- 226 3. If the Printer accepts the HTTP request, it sends a 100 (Continue) response and continues to read
227 from the input stream.
- 228 4. If the client receives a 100 (Continue) response, it now has a reasonable expectation that the HTTP
229 request will succeed. The client now transmits the request body.
- 230 5. After the Printer receives and processes the request body, it sends a final HTTP status code in
231 response.

232

233 If the Request-URI identifies a resource protected by digest authentication, the flow of events is more
234 like this:

- 235 1. The client sends an HTTP request header containing the "Expect: 100-continue" header field, but
236 waits before transmitting the request body.
- 237 2. The Printer examines the HTTP header and rejects the request with 401 (Unauthorized) status and
238 a "WWW-Authenticate" header field containing at least one challenge.
- 239 3. The client sends a new HTTP request header containing an "Authorization" header field and an
240 "Expect: 100-continue" header field.
- 241 4. If the Printer accepts the new HTTP request, it sends a 100 (Continue) response and continues to
242 read from the input stream.
- 243 5. If the client receives a 100 (Continue) response, it now has a reasonable expectation that the
244 HTTP request will succeed. The client now transmits the request body.

245 After the Printer receives and processes the request body, it sends a final HTTP status code in response.

246 Note that a Printer can reject a request at either the HTTP level or the IPP level. E.g., you could get an
247 HTTP (401 Unauthorized) or you could get HTTP 200 (OK) with an IPP client-error-not-authenticated
248 (0x0402). Receiving 100 (Continue) status tells a client that the Printer is willing to accept the HTTP
249 request, but says nothing about whether or not an IPP request (in the body of the HTTP request) will be
250 accepted. A client should use the Validate-Job IPP operation to determine whether or not an IPP Print-
251 Job request will be accepted. Printers MUST always apply the same authorization requirements to
252 Validate-Job as to Print-Job. I.e., if a given Print-Job request would result in a challenge, then so must
253 the corresponding Validate-Job request.

254 Some Printers may authorize access by object, identified by the HTTP Request-URI, while others may
255 authorize access by operation, identified by the IPP "operation-id" request attribute. If a client receives
256 the HTTP 200 (OK)/IPP client-error-not-authenticated (0x0402) combination, it means that the client
257 should look at the Printer's "uri-authentication-supported" and "uri-supported" attributes and look for a
258 more authenticated URI.

259 According to the Digest Authentication standard [RFC2617], the "nonce" value in the Printer's
260 challenge may be good for one use only (for those really paranoid about replay attacks). Therefore, a
261 Printer may issue a challenge for each new request. A client may include an Authorization header
262 preemptively; doing so improves server efficiency and avoids extra round trips for authentication
263 challenges. The Printer may choose to accept the old Authorization header information, even though the
264 nonce value included might not be fresh. Alternatively, the Printer may return a 401 HTTP response
265 with a new nonce value, causing the client to retry the request; by specifying stale=TRUE with this
266 response, the server tells the client to retry with the new nonce, but without prompting for a new
267 username and password.

268 Some clients cannot produce the document data for a Print-Job more than one time, making complete
269 retries impossible. Such clients should use this algorithm to print jobs reliably:

- 270 1. The client sends an HTTP POST request header containing the "Expect: 100-continue" header
271 field.
- 272 2. The client waits for a response before transmitting the request body.
 - 273 a) If the client receives a 100 (Continue) response the client transmits an HTTP request body
274 containing a Validate-Job IPP request.
 - 275 b) If the client receives a 401 (Unauthorized) response, it sends a new HTTP POST request
276 header containing an "Authorization" header field with a response the Printer's "WWW-
277 Authenticate" challenge, and goes back to step 2.
- 278 3. If the client receives an HTTP 200 (OK) response containing an IPP response with one of the
279 success status codes, the client sends an HTTP POST request header containing the "Expect: 100-
280 continue" header field and an "Authorization" header field containing any cached credentials.
- 281 4. The client waits for a response before transmitting the request body.
 - 282 a) If the client receives a 100 (Continue) response the client transmits an HTTP request body
283 containing a Print-Job IPP request.
 - 284 b) If the client receives a 401 (Unauthorized) response, it sends a new HTTP POST request
285 header containing an "Authorization" header field with a response the Printer's "WWW-
286 Authenticate" challenge, and goes back to step 4.

287

288 It is possible to achieve the same results without using 100-continue, but it takes more round trips.:

- 289 1. Send a Validate-Job request to provoke a challenge from the Printer.
- 290 2. If the Printer responds with HTTP 401 (Unauthorized), send another Validate-Job request
291 containing an "Authorization" HTTP header field with a response the Printer's "WWW-Authenticate"
292 challenge, to see if the Print-Job request will be accepted.

293 3. If the Printer accepts the Validate-Job, send another Validate-Job without an "Authorization"
294 header field, to get a fresh nonce.

295 4. Finally, send the Print-Job request containing an "Authorization" HTTP header field with a
296 response the Printer's "WWW-Authenticate" challenge.

297 Note that for this to work, the response to a Printer's "WWW-Authenticate" challenge for Validate-Job
298 must also be valid for Print-Job.

299 **8 References (Informational)**

300 Since this is an information document, all these references are informational.

301 [CGI]

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

303 [IANA-CS]

304 IANA Registry of Coded Character Sets: <ftp://ftp.iana.org/in-notes/iana/assignments/character-sets>

305 [ldap-printer]

306 Fleming, P., Jones, K., Lewis, H., McDonald, I., "Internet Printing Protocol (IPP): LDAP Schema
307 for Printer Services", <draft-ietf-ipp-ldap-printer-schema-05.txt>, work in progress, August 28,
308 2001.

309 [ipp-set-ops]

310 Kugler, C., Hastings, T., Herriot, R., Lewis, H., "Internet Printing Protocol (IPP): Job and Printer Set
311 Operations", <draft-ietf-ipp-job-printer-set-ops-05.txt>, work in progress, August 28, 2001.

312 [RFC793]

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

314 [RFC1123]

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

316 [RFC2026]

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

318 [RFC2119]

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

320 [RFC2251]

321 Wahl, Howes, Kille. Lightweight Directory Access Protocol (v3), RFC 2251, December 1997.

322 [RFC2252]

323 Wahl, Coulbeck, Howes, Kille. Lightweight Directory Access Protocol (v3): Attribute Syntax
324 Definitions, RFC 2252, December 1997.

- 325 [RFC2396]
326 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax",
327 RFC 2396, August 1998.
- 328 [RFC2565]
329 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
330 Semantics", RFC 2566, April 1999.
- 331 [RFC2566]
332 Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.0: Encoding and
333 Transport", RFC 2565, April 1999.
- 334 [RFC2567]
335 Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.
- 336 [RFC2568]
337 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
338 RFC 2568, April 1999.
- 339 [RFC2569]
340 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC
341 2569, April 1999.
- 342 [RFC2608]
343 E. Guttman, C. Perkins, J. Veizades, M. Day. Service Location Protocol, Version 2, RFC 2608,
344 June 1999. (Updates RFC 2165)
- 345 [RFC2609]
346 E. Guttman, C. Perkins, J. Kempf. Service Templates and service: Schemes, RFC 2609, June 1999.
347 (Updates RFC 2165)
- 348 [RFC2616]
349 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext
350 Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
- 351 [RFC2910]
352 Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.0: Encoding and
353 Transport", RFC 2910, September, 2000.
- 354 [RFC2911]
355 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
356 Semantics", RFC 2911, September, 2000.
- 357 [RFC2926]
358 Kempf, Moats, St. Pierre. Conversion of LDAP Schemas to and from SLP Templates, RFC 2926,
359 September 2000.

- 360 [RFC3143]
361 Known HTTP Proxy/Caching Problems. I. Cooper, J. Dilley. June 2001. (Status:
362 INFORMATIONAL)
- 363 [Servlet]
364 Servlet Specification Version 2.1 (<http://java.sun.com/products/servlet/2.1/index.html>).
- 365 [svrloc-printer]
366 St. Pierre, P., Isaacson, S., McDonald, I., "Definition of the Printer Abstract Service Type v2.0",
367 <http://www.isi.edu/in-notes/iana/assignments/svrloc-templates/printer.2.0.en> (IANA Registered, May
368 27, 2000).
- 369 [SSL]
370 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.

371 **9 Authors' Address**

- 372
373 Thomas N. Hastings
374 Xerox Corporation
375 701 Aviation Blvd.
376 El Segundo, CA 90245
377 hastings@cp10.es.xerox.com
378
- 379 Carl-Uno Manros
380 Xerox Corporation
381 701 Aviation Blvd.
382 El Segundo, CA 90245
383 cmanros@cp10.es.xerox.com
384
- 385 Carl Kugler
386 Mail Stop 003G
387 IBM Printing Systems Co
388 6300 Diagonal Hwy
389 Boulder CO 80301
390 Kugler@us.ibm.com
391
- 392 Henrik Holst
393 i-data Printing Systems
394 Vadstrupvej 35-43
395 2880 Bagsvaerd, Denmark
396 hh@I-data.com
397
- 398 Peter Zehler
399 Xerox Corporation

400 800 Philips Road
401 Webster, NY 14580
402 pzehler@crt.xerox.com
403

404 IPP Web Page: <http://www.pwg.org/ipp/>
405 IPP Mailing List: ipp@pwg.org
406

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

- 408 1) send it to majordomo@pwg.org
- 409 2) leave the subject line blank
- 410 3) put the following two lines in the message body:
411 subscribe ipp
412 end

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

417

418 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

419

420 10 Description of the Base IPP Documents

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

422 Design Goals for an Internet Printing Protocol [RFC2567]

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

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

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

426 Mapping between LPD and IPP Protocols [RFC2569]

427

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

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

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

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

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

449 **11 Full Copyright Statement**

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

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

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

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

467 **Acknowledgement**

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