

1 INTERNET-DRAFT
2 [draft-ietf-ipp-implementers-guide-01981029.txt](#)

T. Hastings
Xerox Corporation
C. Manros
Xerox Corporation

~~December 6~~[October 29](#), 1998

9 Internet Printing Protocol/1.0: Implementer's Guide
10 Copyright (C) The Internet Society (date). All Rights Reserved.

11 Status of this Memo

12 This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering
13 Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working
14 documents as Internet-Drafts.

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

18 To learn the current status of any Internet-Draft, please check the "Iid-abstracts.txt" listing contained in the
19 Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific
20 Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

21

22 Abstract

23 This document is one of a set of documents, which together describe all aspects of a new Internet Printing
24 Protocol (IPP). IPP is an application level protocol that can be used for distributed printing using Internet
25 tools and technologies. This document contains information that supplements the IPP Model and
26 Semantics [IPP-MOD] and the IPP Transport and Encoding [IPP-PRO] documents. It is intended to help
27 implementers understand IPP/1.0 and some of the considerations that may assist them in the design of their
28 client and/or IPP object implementations. For example, a typical order of processing requests is given,
29 including error checking. Motivation for some of the specification decisions is also included.~~The protocol~~
30 ~~is heavily influenced by the printing model introduced in the Document Printing Application (DPA)~~
31 ~~[ISO10175] standard. Although DPA specifies both end-user and administrative features, IPP version 1.0~~
32 ~~(IPP/1.0) focuses only on end-user functionality~~

33 The full set of IPP documents includes:

34 Design Goals for an Internet Printing Protocol [IPP-REQ] ~~(informational)~~

35 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [IPP-RAT]
36 ~~(informational)~~

37 Internet Printing Protocol/1.0: Model and Semantics [IPP-MOD]

38 Internet Printing Protocol/1.0: Encoding and Transport [IPP-PRO]

39 ~~Internet Printing Protocol/1.0: Implementer's Guide [IPP-IIG] (informational)~~

40 Mapping between LPD and IPP Protocols [IPP LPD] ~~(informational)~~

41 The ~~design goals~~ document, "Design Goals for an Internet Printing Protocol", takes a broad look at
42 distributed printing functionality, and it enumerates real-life scenarios that help to clarify the features that
43 need to be included in a printing protocol for the Internet. It identifies requirements for three types of
44 users: end users, operators, and administrators. The design goals document calls out a subset of end user
45 requirements that are satisfied in IPP/1.0. Operator and administrator requirements are out of scope for
46 version 1.0.

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

51 The ~~model and semantics~~ document, "Internet Printing Protocol/1.0: Model and Semantics", describes a
52 simplified model with abstract objects, their attributes, and their operations. The model introduces a Printer
53 and a Job. The Job supports multiple documents per Job. The model document also addresses how
54 security, internationalization, and directory issues are addressed.

55 The ~~protocol specification document~~, "Internet Printing Protocol/1.0: Encoding and Transport", is a formal
56 mapping of the abstract operations and attributes defined in the model document onto HTTP/1.1. ~~The~~
57 ~~protocol specification~~ ~~It also~~ defines the encoding rules for a new Internet media type called
58 "application/ipp".

59 ~~The implementer's guide, "Internet Printing Protocol/1.0: Implementer's Guide", gives advice to~~
60 ~~implementers of IPP clients and IPP objects related to the model and protocol documents.~~

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

63 ~~Notice~~

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

TABLE OF CONTENTS

67			
68	1	Introduction	5
69	1.1	Conformance language	5
70	1.2	Other terminology	5
71	2	Model and Semantics	5
72	2.1	Summary of Operation Attributes.....	5
73	2.2	Suggested Operation Processing Steps for IPP Objects (Issue 1.21).....	9
74	2.2.1	Suggested Operation Processing Steps for all Operations	9
75	2.2.1.1	Validate version number	10
76	2.2.1.2	Validate operation identifier	10
77	2.2.1.3	Validate the request identifier	10
78	2.2.1.4	Validate attribute group and attribute presence and order	10
79	2.2.1.5	Validate the values of the REQUIRED Operation attributes.....	17
80	2.2.1.6	Validate the values of the OPTIONAL Operation attributes	20
81	2.2.2	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add	
82		Documents.....	22
83	2.2.2.1	Default "ipp-attribute-fidelity" if not supplied.....	22
84	2.2.2.2	Check that the Printer object is accepting jobs	22
85	2.2.2.3	Validate the values of the Job Template attributes	23
86	2.2.3	Algorithm for job validation	23
87	2.2.3.1	Check for conflicting Job Template attributes values.....	28
88	2.2.3.2	Decide whether to REJECT the request.....	28
89	2.2.3.3	For the Validate-Job operation, RETURN one of the success status codes	29
90	2.2.3.4	Create the Job object with attributes to support.....	29
91	2.2.3.5	Return one of the success status codes.....	30
92	2.2.3.6	Accept appended Document Content.....	31
93	2.2.3.7	Scheduling and Starting to Process the Job	31
94	2.2.3.8	Completing the Job	31
95	2.2.3.9	Destroying the Job after completion	31
96	2.2.3.10	Interaction with "ipp-attribute-fidelity"	31
97	2.3	Status codes returned by operation (Issue 1.50)	32
98	2.3.1	Printer Operations	32
99	2.3.1.1	Print-Job.....	32
100	2.3.1.2	Print-URI.....	34
101	2.3.1.3	Validate-Job	34
102	2.3.1.4	Create-Job	34
103	2.3.1.5	Get-Printer-Attributes	34
104	2.3.1.6	Get-Jobs	35
105	2.3.2	Job Operations.....	35
106	2.3.2.1	Send-Document.....	35
107	2.3.2.2	Send-URI	36
108	2.3.2.3	Cancel-Job.....	36
109	2.3.2.4	Get-Job-Attributes.....	37

110	2.4	Validate-Job	37
111	2.5	Case Sensitivity in URIs (issue 1.6).....	37
112	2.6	Character Sets, natural languages, and internationalization	38
113	2.6.1	Character set code conversion support (Issue 1.5).....	38
114	2.6.2	What charset to return when an unsupported charset is requested (Issue 1.19)?.....	39
115	2.6.3	Natural Language Override (NLO) (Issue 1.45).....	39
116	2.7	The "queued-job-count" Printer Description attribute	40
117	2.7.1	Why is "queued-job-count" RECOMMENDED (Issue 1.14)?.....	40
118	2.7.2	Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?.....	41
119	2.8	Sending empty attribute groups (Issue 1.16)	41
120	2.9	Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)	41
121	2.10	Returning job-state in Print-Job response (Issue 1.30)	42
122	2.11	Flow controlling the data portion of a Print-Job request (Issue 1.22).....	42
123	2.12	Multi-valued attributes (Issue 1.31).....	43
124	2.13	Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)..	43
125	2.14	The 'none' value for empty sets (Issue 1.37).....	43
126	2.15	Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?.....	44
127	2.16	The "multiple-document-handling" Job Template attribute and support of multiple document jobs	
128		44	
129	3	Encoding and Transport.....	44
130	3.1	General Headers	46
131	3.2	Request Headers.....	46
132	3.3	Response Headers	47
133	3.4	Entity Headers.....	48
134	3.5	Optional support for HTTP/1.0.....	48
135	3.6	HTTP/1.1 Chunking.....	49
136	4	References	49
137	4.1	Authors' Address	50
138	5	Appendix C: Full Copyright Statement.....	50
139	6	Change History.....	51
140	6.1	Changes to produce the December 6, 1998 version from the November 16, 1998 version:	51
141			

142

143 **1 Introduction**

144 This document contains information that supplements the IPP Model and Semantics [IPP-MOD] and the
145 IPP Transport and Encoding [IPP-PRO] documents. As such this information is not part of the formal
146 specifications. Instead information is presented to help implementers understand the specification,
147 including some of the motivation for decisions taken by the committee in developing the specification.
148 Some of the implementation considerations are intended to help implementers design their client and/or IPP
149 object implementations. If there are any contradictions between this document and [IPP-MOD] or [IPP-
150 PRO], those documents take precedence over this document.

151 1.1 Conformance language

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

157 Implementers should read section 13 in [IPP-MOD] in order to understand these capitalized words. The
158 words MUST, MUST NOT, and REQUIRED indicate what implementations are required to support in a
159 client or IPP object in order to be conformant to [IPP-MOD] and [IPP-PRO]. MAY, NEED NOT, and
160 OPTIONAL indicate was is merely allowed as an implementer option. The verbs SHOULD and SHOULD
161 NOT indicate suggested behavior, but which is not required or disallowed, respectively, in order to
162 conform to the specification.

163 1.2 Other terminology

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

166 **2 Model and Semantics**

167 This section discusses various aspects of IPP/1.0 Model and Semantics [IPP-MOD].

168 ~~1.1~~2.1 Summary of Operation Attributes

169 Legend for the following table:

170 R indicates a REQUIRED operation or attribute for an implementation to support

171 O indicates an OPTIONAL operation or attribute for an implementation to support

Table 1. Summary of operation attributes

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

172

Operation Attributes	Printer Operations						Job Operations				
	Requests					Responses	Requests				Responses
	Print-Job, Validate-Job	Print-URI (O)	Create-Job (O)	Get-Printer-Attributes	Get-Jobs	All Operations	Send-Document (O)	Send-URI (O)	Cancel-Job	Get-Job-Attributes	All Operations
Operation attributes—OPTIONAL to be supplied by the sender											
status-message						O					O
compression	O	O					O	O			
document-format	R	R		O			R	R			
document-name	O	O					O	O			
document-natural-language	O	O					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									O		
my-jobs					R						
requested-attributes				R	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.

173

174

175 2.2 Suggested Operation Processing Steps for IPP Objects (Issue 1.21)

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

185 When an IPP object receives a request, the IPP object either accepts or rejects the request. In order to
186 determine whether or not to accept or reject the request, the IPP object SHOULD execute the following
187 steps. The order of the steps may be rearranged and/or combined, including making one or multiple passes
188 over the request. ~~Therefore, the error status codes returned may differ between implementations.~~

189 A client MUST supply requests that would pass all of the error checks indicated here in order to be a
190 conforming client. Therefore, a client SHOULD supply requests that are conforming, in order to avoid
191 being rejected by some IPP object implementations and/or risking different semantics by different
192 implementations of forgiving implementations. For example, a forgiving implementation that accepts
193 multiple occurrences of the same attribute, rather than rejecting the request might use the first occurrences,
194 while another might use the last occurrence. Thus such a non-conforming client would get different results
195 from the two forgiving implementations.~~The next section contains the additional steps for the Print-Job,~~
196 ~~Validate-Job, Print-URI, Create-Job, Send-Document, and Send-URI operations that create jobs, adds~~
197 ~~documents, and validates jobs.~~

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

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

204 ~~1.1.1~~2.2.1 Suggested Operation Processing Steps for all Operations

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

208 1.1.1.2.2.1.1 Validate version number

209 Every request and every response contains the "version-number" attribute. The value of this attribute is the
210 major and minor version number of the syntax and semantics that the client and IPP object is using,
211 respectively. The "version-number" attribute remains in a fixed position across all future versions so that
212 all clients and IPP object that support future versions can determine which version is being used. The IPP
213 object checks to see if the major version number supplied in the request is supported. If not, the Printer
214 object REJECTS the request and RETURNS the 'server-error-version-not-supported' status code in the
215 response. The IPP object returns in the "version-number" response attribute the major and minor version
216 for the error response. Thus the client can learn at least one major and minor version that the IPP object
217 supports. The IPP object is encouraged to return the closest version number to the one supplied by the
218 client.

219 The checking of the minor version number is implementation dependent, however if the client supplied
220 minor version is explicitly supported, the IPP object MUST respond using that identical minor version
221 number. If the requested minor version is not supported (the requested minor version is either higher or
222 lower) than a supported minor version, the IPP object SHOULD return the closest supported minor version.

223 1.1.1.2.2.1.2 Validate operation identifier

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

227 1.1.1.3.2.1.3 Validate the request identifier

228 The Printer object **SHOULD NOT** check to see if the "request-id" attribute supplied by the client is in
229 range. ~~If the value is not~~ between 1 and $2^{*}31 - 1$ (inclusive), ~~but copies all 32 bits~~ ~~the Printer object~~
230 ~~REJECTS the request and returns the 'client error bad request' status code in the response.~~

231 Note: The "version-number", ~~attribute,~~ "operation-id", and the "request-id" ~~attributes-parameters are in~~
232 ~~the same~~ fixed octet positions in the IPP/1.0 encoding. The "version-number" parameter will be the same
233 fixed octet position in all versions of the protocol. These fields are validated before proceeding with the
234 rest of the validation.

235 1.1.1.4.2.1.4 Validate attribute group and attribute presence and order

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

237 1.1.1.1.2.2.1.4.1 Validate the presence and order of attribute groups

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

241 If an IPP object receives a request with (1) required attribute groups missing, or (2) the attributes groups are
242 out of order, or (3) the groups are repeated, the IPP object REJECTS the request and RETURNS the 'client-

243 error-bad-request' status code. For example, it is an error for the Job Template Attributes group to occur
244 before the Operation Attributes group, for the Operation Attributes group to be omitted, or for an attribute
245 group to occur more than once, except in the Get-Jobs response.

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

250 1.1.1.1.2.2.1.4.2 Ignore unknown attribute groups in the expected position

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

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

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

263 1.1.1.1.3.2.2.1.4.3 Validate the presence of a single occurrence of required Operation attributes

264 Client requests and IPP object responses contain Operation attributes that [IPP-MOD] Section 3 requires to
265 be present. Attributes within a group may be in any order, except for the ordering of target, charset, and
266 natural languages attributes. These attributes ~~must~~**MUST** be first, and ~~must~~**MUST** be supplied in the
267 following order: charset, natural language, and then target. An IPP object verifies that the attributes that
268 Section 4 requires to be supplied by the client have been supplied in the request (attributes without an * in
269 the following tables). An asterisk (*) indicates groups and Operation attributes that the client may omit in a
270 request or an IPP object may omit in a response.

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

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

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

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

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

291 R indicates a REQUIRED attribute that an IPP object MUST support
292 O indicates an OPTIONAL attribute that an IPP object NEED NOT support
293 * indicates that a client MAY omit the attribute in a request and that an IPP object MAY omit
294 the attribute in a response. The absence of an * means that a client MUST supply the
295 attribute in a request and an IPP object MUST supply the attribute in a response.
296

297 Operation Requests

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

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

301

302 Print-Job Request:

303 Group 1: Operation Attributes (R)

304 attributes-charset (R)

305 attributes-natural-language (R)

306 printer-uri (R)

307 requesting-user-name (R*)

308 job-name (R*)

309 ipp-attribute-fidelity (R*)

310 document-name (R*)

311 document-format (R*)

312 document-natural-language (O*)

313 compression (O*)

314 job-k-octets (O*)

315 job-impressions (O*)

316 job-media-sheets (O*)

317 Group 2: Job Template Attributes (R*)

318 <Job Template attributes> (O*)

319 (see [IPP-MOD] Section 4.2)

320 Group 3: Document Content (R)

321 <document content>

322

323 Validate-Job Request:

324 Group 1: Operation Attributes (R)

325 attributes-charset (R)

326 attributes-natural-language (R)

327 printer-uri (R)

328 requesting-user-name (R*)

329 job-name (R*)

330 ipp-attribute-fidelity (R*)

331 document-name (R*)

332 document-format (R*)

333 document-natural-language (O*)

334 compression (O*)

335 job-k-octets (O*)

336 job-impressions (O*)

337 job-media-sheets (O*)

338 Group 2: Job Template Attributes (R*)

339 <Job Template attributes> (O*)

340 (see [IPP-MOD] Section 4.2)

341

342 Create-Job Request:

343 Group 1: Operation Attributes (R)

344 attributes-charset (R)

345 attributes-natural-language (R)

346 printer-uri (R)

347 requesting-user-name (R*)

348 job-name (R*)

349 ipp-attribute-fidelity (R*)
350 job-k-octets (O*)
351 job-impressions (O*)
352 job-media-sheets (O*)
353 Group 2: Job Template Attributes (R*)
354 <Job Template attributes> (O*) (see
355 (see [IPP-MOD] Section 4.2)
356
357 Print-URI Request:
358 Group 1: Operation Attributes (R)
359 attributes-charset (R)
360 attributes-natural-language (R)
361 printer-uri (R)
362 document-uri (R)
363 requesting-user-name (R*)
364 job-name (R*)
365 ipp-attribute-fidelity (R*)
366 document-name (R*)
367 document-format (R*)
368 document-natural-language (O*)
369 compression (O*)
370 job-k-octets (O*)
371 job-impressions (O*)
372 job-media-sheets (O*)
373 Group 2: Job Template Attributes (R*)
374 <Job Template attributes> (O*) (see
375 (see [IPP-MOD] Section 4.2)
376
377 Send-Document Request:
378 Group 1: Operation Attributes (R)
379 attributes-charset (R)
380 attributes-natural-language (R)
381 (printer-uri & job-id) | job-uri (R)
382 last-document (R)
383 requesting-user-name (R*)
384 document-name (R*)
385 document-format (R*)
386 document-natural-language (O*)
387 compression (O*)
388 Group 2: Document Content (R*)
389 <document content>
390
391 Send-URI Request:
392 Group 1: Operation Attributes (R)
393 attributes-charset (R)
394 attributes-natural-language (R)
395 (printer-uri & job-id) | job-uri (R)

396 last-document (R)
397 document-uri (R)
398 requesting-user-name (R*)
399 document-name (R*)
400 document-format (R*)
401 document-natural-language (O*)
402 compression (O*)
403
404 Cancel-Job Request:
405 Group 1: Operation Attributes (R)
406 attributes-charset (R)
407 attributes-natural-language (R)
408 (printer-uri & job-id) | job-uri (R)
409 requesting-user-name (R*)
410 message (O*)
411
412 Get-Printer-Attributes Request:
413 Group 1: Operation Attributes (R)
414 attributes-charset (R)
415 attributes-natural-language (R)
416 printer-uri (R)
417 requesting-user-name (R*)
418 requested-attributes (R*)
419 document-format (R*)
420
421 Get-Job-Attributes Request:
422 Group 1: Operation Attributes (R)
423 attributes-charset (R)
424 attributes-natural-language (R)
425 (printer-uri & job-id) | job-uri (R)
426 requesting-user-name (R*)
427 requested-attributes (R*)
428
429 Get-Jobs Request:
430 Group 1: Operation Attributes (R)
431 attributes-charset (R)
432 attributes-natural-language (R)
433 printer-uri (R)
434 requesting-user-name (R*)
435 limit (R*)
436 requested-attributes (R*)
437 which-jobs (R*)
438 my-jobs (R*)
439

440 Operation Responses

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

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

444

445 Print-Job Response:

446 Print-URI Response:

447 Create-Job Response:

448 Send-Document Response:

449 Send-URI Response:

450 Group 1: Operation Attributes (R)

451 attributes-charset (R)

452 attributes-natural-language (R)

453 status-message (O*)

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

455 <unsupported attributes> (R*)

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

457 job-uri (R)

458 job-id (R)

459 job-state (R)

460 job-state-reasons (O*)

461 job-state-message (O*)

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

463

464 Validate-Job Response:

465 Cancel-Job Response:

466 Group 1: Operation Attributes (R)

467 attributes-charset (R)

468 attributes-natural-language (R)

469 status-message (O*)

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

471 <unsupported attributes> (R*)

472

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

475

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

478
479 Get-Printer-Attributes Response:
480 Group 1: Operation Attributes (R)
481 attributes-charset (R)
482 attributes-natural-language (R)
483 status-message (O*)
484 Group 2: Unsupported Attributes (R*) (see Note 4)
485 <unsupported attributes> (R*)
486 Group 3: Printer Object Attributes(R*) (see Note 2)
487 <requested attributes> (R*)
488

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

491
492 Get-Job-Attributes Response:
493 Group 1: Operation Attributes (R)
494 attributes-charset (R)
495 attributes-natural-language (R)
496 status-message (O*)
497 Group 2: Unsupported Attributes (R*) (see Note 4)
498 <unsupported attributes> (R*)
499 Group 3: Job Object Attributes(R*) (see Note 2)
500 <requested attributes> (R*)
501

502 Get-Jobs Response:
503 Group 1: Operation Attributes (R)
504 attributes-charset (R)
505 attributes-natural-language (R)
506 status-message (O*)
507 Group 2: Unsupported Attributes (R*) (see Note 4)
508 <unsupported attributes> (R*)
509 Group 3: Job Object Attributes(R*) (see Note 2, 5)
510 <requested attributes> (R*)
511

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

514 [1.1.1.5.2.1.5](#) Validate the values of the REQUIRED Operation attributes

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

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

- 519 a) that the length of each Operation attribute value is correct for the attribute syntax tag supplied
520 by the client according to [\[IPP-MOD\]](#) Section 4.1,
- 521 b) that the attribute syntax tag is correct for that Operation attribute according to [\[IPP-MOD\]](#)
522 Section 3,
- 523 c) that the value is in the range specified for that Operation attribute according to [\[IPP-MOD\]](#)
524 Section 3,
- 525 d) that multiple values are supplied by the client only for operation attributes that are multi-valued,
526 i.e., that are 1setOf X according to [\[IPP-MOD\]](#) Section 3.

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

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

539

540 -----
attributes-charset (charset)

541 IF NOT any single non-empty 'charset' value less than or equal to 63 octets, REJECT/RETURN 'client-
542 error-request-value-too-long'.

543 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-charset-
544 not-supported".
545

546 attributes-natural-language(naturalLanguage)

547 IF NOT any single non-empty 'naturalLanguage' value less than or equal to 63 octets,
548 REJECT/RETURN 'client-error-request-value-too-long'.

549 ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-
550 language-supported" attribute.
551

552 requesting-user-name

553 IF NOT any single 'name' value less than or equal to 255 octets, REJECT/RETURN 'client-error-
554 request-value-too-long'.

555 IF the IPP object can obtain a better authenticated name, use it instead.
556

557 job-name(name)
558 IF NOT any single 'name' value less than or equal to 255 octets, REJECT/RETURN 'client-error-
559 request-value-too-long'.
560 IF NOT supplied by the client, the Printer object creates a name from the document-name or document-
561 uri.
562

563 document-name (name)
564 IF NOT any single 'name' value less than or equal to 255 octets, REJECT/RETURN 'client-error-
565 request-value-too-long'.
566

567 ipp-attribute-fidelity (boolean)
568 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-error-
569 bad-request'.
570 IF NOT supplied by the client, the IPP object assumes the value 'false'.
571

572 document-format (mimeType)
573 IF NOT any single non-empty 'mimeType' value less than or equal to 255 octets,
574 REJECT/RETURN 'client-error-request-value-too-long'.
575 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN 'client-error-
576 document-format-not-supported'.
577 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's "document-
578 format-default" attribute.
579

580 document-uri (uri)
581 IF NOT any single non-empty 'uri' value less than or equal to 1023 octets, REJECT/RETURN 'client-
582 error-request-value-too-long'.
583 IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.
584 IF scheme is NOT in the Printer object's "reference-uri-schemes-supported" attribute,
585 REJECT/RETURN 'client-error-uri-scheme-not-supported'.
586

587 last-document (boolean)
588 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-error-
589 bad-request'.
590

591 job-id (integer(1:MAX))
592 IF NOT any single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN
593 'client-error-bad-request'.
594 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-error-
595 gone' status code, if keep track of recently deleted jobs.
596

597 requested-attributes (1setOf keyword)

598 IF NOT any number of 'keyword' values less than or equal to 255 octets, REJECT/RETURN 'client-
599 error-request-value-too-long'.

600 Ignore unsupported values which are the keyword names of unsupported attributes. Don't bother to
601 copy such requested (unsupported) attributes to the Unsupported Attribute response group since the
602 response will not return them.
603

604 which-jobs (type2 keyword)

605 IF NOT a single 'keyword' value less than or equal to 255 octets, REJECT/RETURN 'client-error-
606 request-value-too-long'.

607 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to the
608 Unsupported Attributes response group and REJECT/RETURN 'client-error-attributes-or-values-
609 not-supported'.

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

612 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.
613

614 my-jobs (boolean)

615 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-error-
616 bad-request'.

617 IF NOT supplied by the client, the IPP object assumes the 'false' value.
618

619 limit (integer(1:MAX))

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

622 IF NOT supplied by the client, the IPP object returns all jobs, no matter how many.
623

624 -----
625

626 ~~1.1.1.6~~2.2.1.6 Validate the values of the OPTIONAL Operation attributes

627 OPTIONAL Operation attributes are those that an IPP object MAY or MAY NOT support. An IPP object
628 validates the values of the OPTIONAL attributes supplied by the client. The IPP object performs the same
629 syntactic validation checks for each OPTIONAL attribute value as in Section 2.2.1.5. As in Section
630 2.2.1.5, if any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' or the
631 'client-error-request-value-too-long' status code.

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

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

639 -----

640 document-natural-language (naturalLanguage)

641 IF NOT any single non-empty 'naturalLanguage' value less than or equal to 63 octets,

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

643 IF NOT a value that the Printer object supports in document formats, (no corresponding "xxx-

644 supported" Printer attribute), REJECT/RETURN 'client-error-natural-language-not-supported'.

645

646 compression (type3 keyword)

647 IF NOT any single 'keyword' values less than or equal to 255 octets, REJECT/RETURN 'client-error-
648 request-value-too-long'.

649 IF NOT in the Printer object's "compression-supported" attribute, copy the attribute and the

650 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-

651 error-attributes-or-values-not-supported'.

652

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

654 IF NOT any single 'integer' value equal to 4 octets,

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

656 IF NOT in the range of the Printer object's "job-k-octets-supported" attribute, copy the attribute and the

657 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-

658 error-attributes-or-values-not-supported'.

659

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

661 IF NOT any single 'integer' value equal to 4 octets,

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

663 IF NOT in the range of the Printer object's "job-impressions-supported" attribute, copy the attribute and

664 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-

665 error-attributes-or-values-not-supported'.

666

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

668 IF NOT any single 'integer' value equal to 4 octets,

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

670 IF NOT in the range of the Printer object's "job-media-supported" attribute, copy the attribute and the

671 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-

672 error-attributes-or-values-not-supported'.

673

674 message (text(127))

675 IF NOT any single 'text' value less than or equal to 127 octets,

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

677

678 unknown or unsupported attribute

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

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

683

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

699 2.2.2 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add
700 Documents

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

704 ~~1.1.1~~2.2.2.1 Default "ipp-attribute-fidelity" if not supplied

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

707 ~~1.1.1~~2.2.2.2 Check that the Printer object is accepting jobs

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

710 4.1.1.32.2.2.3 Validate the values of the Job Template attributes

711 An IPP object validates the values of all Job Template attribute supplied by the client. The IPP object
712 performs the analogous syntactic validation checks of each Job Template attribute value that it performs for
713 Operation attributes (see Section 2.2.1.5.):

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

716 b) that the attribute syntax tag is correct for that attribute according to [IPP-MOD] Sections 4.2 to
717 4.4.

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

720 As in Section 2.2.1.5, if any of these syntactic checks fail, the IPP object REJECTS the request and
721 RETURNS the 'client-error-bad-request' or 'client-error-request-value-too-long' status code as appropriate,
722 independent of the value of the "ipp-attribute-fidelity". Since such an error is most likely to be an error
723 detected by a client developer, rather than by an end-user, the IPP object NEED NOT return an indication
724 of which attribute had the error in either the Unsupported Attributes Group or the Status Message. The
725 description for each of these syntactic checks is explicitly expressed in the first IF statement in the
726 following table.

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

729 1. reject the operation and return the 'client-error-bad syntax' error status code

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

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

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

734 2.2.3 Algorithm for job validation

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

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

739 1. If U is multi-valued, validate each value X of U by performing the algorithm in Table 2 with
740 each value X. Each validation is separate from the standpoint of returning unsupported values.

741 Example: If U is "finishings" that the client supplies with 'staple', 'bind' values, then X takes on
742 the successive values: 'staple', then 'bind'

743 2. If V is multi-valued, validate X against each Z of V by performing the algorithm in Table 2 with
 744 each value Z. If a value Z validates, the validation for the attribute value X succeeds. If it fails,
 745 the algorithm is applied to the next value Z of V. If there are no more values Z of V, validation
 746 fails.

747 Example" If V is "sides-supported" with values: 'one-sided', 'two-sided-long', and 'two-sided-
 748 short', then Z takes on the successive values: 'one-sided', 'two-sided-long', and 'two-sided-short'.
 749 If the client supplies "sides" with 'two-sided-long', the first comparison fails ('one-sided' is not
 750 equal to 'two-sided-long'), the second comparison succeeds ('two-sided-long' is equal to 'two-
 751 sided-long"), and the third comparison ('two-sided-short' with 'two-sided-long') is not even
 752 performed.

753 3. If both U and V are single-valued, let X be U and Z be V and use the validation rules in Table 2.

754 **Table 2 - 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.

755

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

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

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

773 -----

- 774 job-priority (integer(1:100))
- 775 IF NOT **any** single 'integer' value with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
776 request'.
- 777 IF NOT supplied by the client, use the value of the Printer object's "job-priority-default" attribute at job
778 submission time.
- 779 IF NOT in the range 1 to 100, inclusive, copy the attribute and the unsupported value to the
780 Unsupported Attributes response group.
- 781 Map the value to the nearest supported value in the range 1:100 as specified by the number of discrete
782 values indicated by the value of the Printer's "job-priority-supported" attribute. See the formula in
783 [IPP-MOD] Section 4.2.1.
784
- 785 job-hold-until (type3 keyword | name)
- 786 IF NOT **any** single 'keyword' or 'name' value with a length less than or equal to 255 octets,
787 REJECT/RETURN 'client-error-request-value-too-long'.
- 788 IF NOT supplied by the client, use the value of the Printer object's "job-hold-until" attribute at job
789 submission time.
- 790 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the
791 unsupported value to the Unsupported Attributes response group.
792
- 793 job-sheets (type3 keyword | name)
- 794 IF NOT **any** single 'keyword' or 'name' value with a length less than or equal to 255 octets,
795 REJECT/RETURN 'client-error-request-value-too-long'.
- 796 IF NOT in the Printer object's "job-sheets-supported" attribute, copy the attribute and the unsupported
797 value to the Unsupported Attributes response group.
798
- 799 multiple-document-handling (type2 keyword)
- 800 IF NOT **any** single 'keyword' value with a length less than or equal to 255 octets, REJECT/RETURN
801 'client-error-request-value-too-long'.
- 802 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute and
803 the unsupported value to the Unsupported Attributes response group.
804
- 805 copies (integer(1:MAX))
- 806 IF NOT **any** single 'integer' value with a length equal to 4 octets,
807 REJECT/RETURN 'client-error-bad-request'.
- 808 IF NOT in range of the Printer object's "copies-supported" attribute
809 copy the attribute and the unsupported value to the Unsupported Attributes response group.
810
- 811 finishings (1setOf type2 enum)
- 812 IF NOT **any** 'enum' value(s) each with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
813 request'.

814 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the unsupported
815 value(s), but not any supported values, to the Unsupported Attributes response group.
816

817 page-ranges (1setOf rangeOfInteger(1:MAX))

818 IF NOT **any** 'rangeOfInteger' value(s) each with a length equal to 8 octets, REJECT/RETURN 'client-
819 error-bad-request'.

820 IF first value is greater than second value in any range, the ranges are not in ascending order, or ranges
821 overlap, REJECT/RETURN 'client-error-bad-request'.

822 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to the
823 Unsupported Attributes response group and set the value to the "out-of-band" 'unsupported' value.
824

825 sides (type2 keyword)

826 IF NOT **any** single 'keyword' value with a length less than or equal to 255 octets, REJECT/RETURN
827 'client-error-request-value-too-long'.

828 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported value
829 to the Unsupported Attributes response group.
830

831 number-up (integer(1:MAX))

832 IF NOT **any** single 'integer' value with a length equal to 4 octets,
833 REJECT/RETURN 'client-error-bad-request'.

834 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"
835 attribute, copy the attribute and value to the Unsupported Attribute response group.
836

837 orientation-requested (type2 enum)

838 IF NOT **any** single 'enum' value with a length equal to 4 octets,
839 REJECT/RETURN 'client-error-bad-request'.

840 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the
841 unsupported value to the Unsupported Attributes response group.
842

843 media (type3 keyword | name)

844 IF NOT **any** single 'keyword' or 'name' value with a length less than or equal to 255 octets,
845 REJECT/RETURN 'client-error-request-value-too-long'.

846 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported value
847 to the Unsupported Attributes response group.
848

849 printer-resolution (resolution)

850 IF NOT **any** single 'resolution' value with a length equal to 9 octets,
851 REJECT/RETURN 'client-error-bad-request'.

852 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute and
853 the unsupported value to the Unsupported Attributes response group.
854

855 print-quality (type2 enum)

856 IF NOT ~~any~~ single 'enum' value with a length equal to 4 octets,

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

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

860

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

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

864 REJECT/RETURN 'client-error-bad-request' if the length of the attribute syntax is fixed or 'client-error-
865 request-value-too-long' if the length of the attribute syntax is variable.

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

870 -----

871

872 If the attribute syntax is supported AND the length check fails, the IPP object REJECTS the request and
873 RETURNS the 'client-error-bad-request' if the length of the attribute syntax is fixed or the 'client-error-
874 request-value-too-long' status code if the length of the attribute syntax is variable.; ~~else~~ Otherwise, the IPP
875 object copies the unsupported Job Template attribute to the Unsupported Attributes response group and
876 changes the attribute value to the "out-of-band" 'unsupported' value. The following table shows the length
877 checks for all attribute syntaxes. In the following table: "<=" means less than or equal, "=" means equal
878 to:

879	Name	Octet length check for read-write attributes
880	-----	-----
881	'textWithLanguage	<= 1023 AND 'naturalLanguage' <= 63
882	'textWithoutLanguage'	<= 1023
883	'nameWithLanguage'	<= 255 AND 'naturalLanguage' <= 63
884	'nameWithoutLanguage'	<= 255
885	'keyword'	<= 255
886	'enum'	= 4
887	'uri'	<= 1023
888	'uriScheme'	<= 63
889	'charset'	<= 63
890	'naturalLanguage'	<= 63
891	'mimeType'	<= 255
892	'octetString'	<= 1023
893	'boolean'	= 1
894	'integer'	= 4
895	'rangeOfInteger'	= 8
896	'dateTime'	= 11
897	'resolution'	= 9
898	'1setOf X'	
899		

900 2.2.3.1 Check for conflicting Job Template attributes values

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

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

914 2.2.3.2 Decide whether to REJECT the request

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

918 (1) 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes
 919 supplied by the client.

920 (2) 'client-error-attributes-or-values-not-supported' status code, otherwise.

921

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

926 2.2.3.3 For the Validate-Job operation, RETURN one of the success status codes

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

928 (1) the "successful-ok" status code, if there are no unsupported or conflicting Job Template attributes or
929 values.

930 (2) the "successful-ok-conflicting-attributes", if there are any conflicting Job Template attribute or
931 values.

932 (3) the "successful-ok-ignored-or-substituted-attributes", if there are only unsupported Job Template
933 attributes or values.

934

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

939 2.2.3.4 Create the Job object with attributes to support

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

941 (1) creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and
942 initializes all of the job's other supported Job Description attributes.

943 (2) removes all unsupported attributes from the Job object.

944 (3) for each unsupported value, removes either the unsupported value or substitutes the unsupported
945 attribute value with some supported value. If an attribute has no values after removing unsupported
946 values from it, the attribute is removed from the Job object (so that the normal default behavior at
947 job processing time will take place for that attribute).

948 (4) for each conflicting value, removes either the conflicting value or substitutes the conflicting
949 attribute value with some other supported value. If an attribute has no values after removing
950 conflicting values from it, the attribute is removed from the Job object (so that the normal default
951 behavior at job processing time will take place for that attribute).

952

953 If there were no attributes or values flagged as unsupported, or the value of "ipp-attribute-fidelity" was
954 'false', the Printer object is able to accept the create request and create a new Job object. If the "ipp-
955 attribute-fidelity" attribute is set to 'true', the Job Template attributes that populate the new Job object are
956 necessarily all the Job Template attributes supplied in the create request. If the "ipp-attribute-fidelity"
957 attribute is set to 'false', the Job Template attributes that populate the new Job object are all the client
958 supplied Job Template attributes that are supported or that have value substitution. Thus, some of the

959 requested Job Template attributes may not appear in the Job object because the Printer object did not
960 support those attributes. The attributes that populate the Job object are persistently stored with the Job
961 object for that Job. A Get-Job-Attributes operation on that Job object will return only those attributes that
962 are persistently stored with the Job object.

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

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

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

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

984 2.2.3.5 Return one of the success status codes

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

- 986 (1) the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes or
987 values.
- 988 (2) the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template
989 attribute or values.
- 990 (3) the 'successful-ok-ignored-or-substituted-attributes' status code, if there are only unsupported Job
991 Template attributes or values.

992
993 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this
994 step. If the unsupported Operation attribute was a serious error, the above already rejected the request in a
995 previous step. If control gets to this step with unsupported Operation attributes being returned, they are not
996 serious errors.

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

999 2.2.3.6 Accept appended Document Content

1000 The Printer object accepts the appended Document Content data and either starts it printing, or spools it for
1001 later processing.

1002 2.2.3.7 Scheduling and Starting to Process the Job

1003 The Printer object uses its own configuration and implementation specific algorithms for scheduling the
1004 Job in the correct processing order. Once the Printer object begins processing the Job, the Printer changes
1005 the Job's state to 'processing'. If the Printer object supports PDL override (the "pdl-override-supported"
1006 attribute set to 'attempted'), the implementation does its best to see that IPP attributes take precedence over
1007 embedded instructions in the document data.

1008 2.2.3.8 Completing the Job

1009 The Printer object continues to process the Job until it can move the Job into the 'completed' state. If an
1010 Cancel-Job operation is received, the implementation eventually moves the Job into the 'canceled' state. If
1011 the system encounters errors during processing that do not allow it to progress the Job into a completed
1012 state, the implementation halts all processing, cleans up any resources, and moves the Job into the 'aborted'
1013 state.

1014 2.2.3.9 Destroying the Job after completion

1015 Once the Job moves to the 'completed', 'aborted', or 'canceled' state, it is an implementation decision as to
1016 when to destroy the Job object and release all associated resources. Once the Job has been destroyed, the
1017 Printer would return either the "client-error-not-found" or "client-error-gone" status codes for operations
1018 directed at that Job.

1019 Note: the Printer object SHOULD NOT re-use a "job-uri" or "job-id" value for a sufficiently long time
1020 after a job has been destroyed, so that stale references kept by clients are less likely to access the wrong
1021 (newer) job.

1022 2.2.3.10 Interaction with "ipp-attribute-fidelity"

1023 Some Printer object implementations may support "ipp-attribute-fidelity" set to 'true' and "pdl-override-
1024 supported" set to 'attempted' and yet still not be able to realize exactly what the client specifies in the create
1025 request. This is due to legacy decisions and assumptions that have been made about the role of job
1026 instructions embedded within the document data and external job instructions that accompany the
1027 document data and how to handle conflicts between such instructions. The inability to be 100% precise
1028 about how a given implementation will behave is also compounded by the fact that the two special
1029 attributes, "ipp-attribute-fidelity" and "pdl-override-supported", apply to the whole job rather than specific

1030 values for each attribute. For example, some implementations may be able to override almost all Job
1031 Template attributes except for "number-up".

1032 2.3 Status codes returned by operation (Issue 1.50)

1033 This section lists all status codes once in the first operation (Print-Job). Then it lists the status codes that
1034 are different or specialized for subsequent operations under each operation.

1035 ~~4.1.1~~2.3.1 Printer Operations

1036 2.3.1.1 Print-Job

1037 The Printer object MUST return one of the following "status-code" values for the indicated reason.
1038 Whether all of the document data has been accepted or not before returning the success or error response
1039 depends on implementation. See Section 14 for a more complete description of each status code.

1040 For the following success status codes, the Job object has been created and the "job-id", and "job-uri"
1041 assigned and returned in the response:

1042 successful-ok: no request attributes were substituted or ignored.

1043 successful-ok-ignored-or-substituted-attributes: some supplied (1) attributes were ignored or (2)
1044 unsupported attribute syntaxes or values were substituted with supported values or were ignored.
1045 Unsupported attributes, attribute syntaxes, or values MUST be returned in the Unsupported
1046 Attributes group of the response.

1047 successful-ok-conflicting-attributes: some supplied attribute values conflicted with the values of other
1048 supplied attributes and were either substituted or ignored. Attributes or values which conflict with
1049 other attributes and have been substituted or ignored MUST be returned in the Unsupported
1050 Attributes group of the response as supplied by the client.

1051

1052 [ipp-mod] section 3.1.6 Operation Status Codes and Messages states (Issue 1.19):

1053 If the Printer object supports the "status-message" operation attribute, it SHOULD use the
1054 REQUIRED 'utf-8' charset to return a status message for the following error status codes (see
1055 section 14): 'client-error-bad-request', 'client-error-charset-not-supported', 'server-error-internal-
1056 error', 'server-error-operation-not-supported', and 'server-error-version-not-supported'. In this case,
1057 it MUST set the value of the "attributes-charset" operation attribute to 'utf-8' in the error response.

1058 For the following error status codes, no job is created and no "job-id" or "job-uri" is returned:

1059 client-error-bad-request: The request syntax does not conform to the specification. ~~The IPP object~~
1060 ~~SHOULD NOT return the "status-message" operation attributes, if supported, if the "attributes-~~
1061 ~~charset" in the request has not been processed.~~

1062 client-error-forbidden: The request is being refused for authorization or authentication reasons. The
1063 implementation security policy is to not reveal whether the failure is one of authentication or
1064 authorization.

1065 client-error-not-authenticated: Either the request requires authentication information to be supplied or
1066 the authentication information is not sufficient for authorization.

1067 client-error-not-authorized: The requester is not authorized to perform the request on the target object.

1068 client-error-not-possible: The request cannot be carried out because of the state of the system. See also
1069 'server-error-not-accepting-jobs' status code which MUST take precedence if the Printer object's
1070 "printer-accepting-jobs" attribute is 'false'.

1071 client-error-timeout: not applicable.

1072 client-error-not-found: the target object does not exist.

1073 client-error-gone: the target object no longer exists and no forwarding address is known.

1074 client-error-request-entity-too-large: the size of the request and/or print data exceeds the capacity of the
1075 IPP Printer to process it.

1076 client-error-request-value-too-long: the size of request variable length attribute values, such as 'text'
1077 and 'name' attribute syntaxes, exceed the maximum length specified in [IPP-MOD] for the attribute
1078 and MUST be returned in the Unsupported Attributes Group.

1079 client-error-document-format-not-supported: the document format supplied is not supported. The
1080 "document-format" attribute with the unsupported value MUST be returned in the Unsupported
1081 Attributes Group. This error SHOULD take precedence over any other 'xxx-not-supported' error,
1082 except 'client-error-charset-not-supported'.

1083 client-error-attributes-or-values-not-supported: one or more supplied attributes, attribute syntaxes, or
1084 values are not supported and the client supplied the "ipp-attributes-fidelity" operation attribute with
1085 a 'true' value. They MUST be returned in the Unsupported Attributes Group as explained below.

1086 client-error-uri-scheme-not-supported: not applicable.

1087 client-error-charset-not-supported: the charset supplied in the "attributes-charset" operation attribute is
1088 not supported. The Printer's "configured-charset" MUST be returned in the response as the value of
1089 the "attributes-charset" operation attribute and used for any 'text' and 'name' attributes returned in
1090 the error response. This error SHOULD take precedence over any other error, unless the request
1091 syntax is so bad that the client's supplied "attributes-charset" cannot be determined.

1092 client-error-conflicting-attributes: one or more supplied attribute values conflicted with each other and
1093 the client supplied the "ipp-attributes-fidelity" operation attribute with a 'true' value. They MUST
1094 be returned in the Unsupported Attributes Group as explained below.

1095 server-error-internal-error: an unexpected condition prevents the request from being fulfilled.

1096 server-error-operation-not-supported: not applicable (since Print-Job is REQUIRED).

1097 server-error-service-unavailable: the service is temporarily overloaded.

1098 server-error-version-not-supported: the version in the request is not supported. The "closest" version
1099 number supported MUST be returned in the response.

1100 server-error-device-error: a device error occurred while receiving or spooling the request or document
1101 data or the IPP Printer object can only accept one job at a time.

1102 server-error-temporary-error: a temporary error such as a buffer full write error, a memory overflow, or
1103 a disk full condition occurred while receiving the request and/or the document data.

1104 server-error-not-accepting-jobs: the Printer object's "printer-is-not-accepting-jobs" attribute is 'false'.

1105 server-error-busy: the Printer is too busy processing jobs to accept another job at this time.

1106 server-error-job-canceled: the job has been canceled by an operator or the system while the client was
1107 transmitting the document data.

1108 [1.1.1.2.3.1.2](#) Print-URI

1109 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Print-
1110 URI with the following specializations and differences. See Section 14 for a more complete description of
1111 each status code.

1112 server-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation
1113 attribute is not supported and is returned in the Unsupported Attributes group.

1114 [1.1.1.3.2.3.1.3](#) Validate-Job

1115 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Validate-
1116 Job. See Section 14 for a more complete description of each status code.

1117 [1.1.1.4.2.3.1.4](#) Create-Job

1118 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Create-
1119 Job with the following specializations and differences. See Section 14 for a more complete description of
1120 each status code.

1121 server-error-operation-not-supported: the Create-Job operation is not supported.

1122 [1.1.1.5.2.3.1.5](#) Get-Printer-Attributes

1123 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-
1124 Printer-Attributes operation with the following specializations and differences. See Section 14 for a more
1125 complete description of each status code.

1126 For the following success status codes, the requested attributes are returned in Group 3 in the response:

1127 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1128 attributes were unsupported.

1129 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1130 operation attribute MAY, but NEED NOT, be returned with the unsupported values.

1131 successful-ok-conflicting-attributes: same as Print-Job.

1132 For the error status codes, Group 3 is returned containing no attributes or is not returned at all:

1133 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.

1134 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.

1135 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1136 MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.

1137 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.

1138 server-error-operation-not-supported: not applicable (since Get-Printer-Attributes is REQUIRED).

1139 server-error-device-error: same as Print-Job, except that no document data is involved.

1140 server-error-temporary-error: same as Print-Job, except that no document data is involved.

1141 server-error-not-accepting-jobs: not applicable..

1142 server-error-busy: same as Print-Job, except the IPP object is too busy to accept even query requests.

1143 server-error-job-canceled: not applicable..

1144 1.1.1.6.2.3.1.6 Get-Jobs

1145 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-
1146 Jobs operation with the following specializations and differences. See Section 14 for a more complete
1147 description of each status code.

1148 For the following success status codes, the requested attributes are returned in Group 3 in the response:

1149 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1150 attributes were unsupported.

1151 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1152 operation attribute MAY, but NEED NOT, be returned with the unsupported values.

1153 successful-ok-conflicting-attributes: same as Print-Job.

1154 For any error status codes, Group 3 is returned containing no attributes or is not returned at all. The
1155 following brief error status code descriptions contain unique information for use with Get-Jobs operation.
1156 See section 14 for the other error status codes that apply uniformly to all operations:

1157 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.

1158 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.

1159 client-error-document-format-not-supported: not applicable.

1160 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1161 MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.

1162 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.

1163 server-error-operation-not-supported: not applicable (since Get-Jobs is REQUIRED).

1164 server-error-device-error: same as Print-Job, except that no document data is involved.

1165 server-error-temporary-error: same as Print-Job, except that no document data is involved.

1166 server-error-not-accepting-jobs: not applicable.

1167 server-error-job-canceled: not applicable.

1168 1.1.2.2.3.2 Job Operations

1169 2.3.2.1 Send-Document

1170 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-
1171 Printer-Attributes operation with the following specializations and differences. See Section 14 for a more
1172 complete description of each status code.

1173 For the following success status codes, the document has been added to the specified Job object and the
1174 job's "number-of-documents" attribute has been incremented:

1175 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

1176 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

1177 successful-ok-conflicting-attributes: same as Print-Job.

1178 For the error status codes, no document has been added to the Job object and the job's "number-of-
1179 documents" attribute has not been incremented:

1180 client-error-not-possible: Same as Print-Job, except that the Printer's "printer-is-accepting-jobs"
1181 attribute is not involved, so that the client is able to finish submitting a multi-document job after this

1182 attribute has been set to 'true'. Another condition is that the state of the job precludes Send-
1183 Document, i.e., the job has already been closed out by the client. However, if the IPP Printer closed
1184 out the job due to timeout, the 'client-error-timeout' error status SHOULD be returned instead.
1185 client-error-timeout: This request was sent after the Printer closed the job, because it has not received a
1186 Send-Document or Send-URI operation within the Printer's "multiple-operation-time-out" period ~~to~~
1187 [Print Job, Create Job, or Validate Job operations](#).
1188 client-error-request-entity-too-large: same as Print-Job.
1189 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attributes-fidelity" operation
1190 attribute is not involved..
1191 server-error-operation-not-supported: the Send-Document request is not supported.
1192 server-error-not-accepting-jobs: not applicable.
1193 server-error-job-canceled: the job has been canceled by an operator or the system while the client was
1194 transmitting the data.

1195 [1.1.1.2.3.2.2](#) Send-URI

1196 All of the Print-Job status code descriptions in Section 3.2.1.2 Print-Job Response with the specializations
1197 described for Send-Document are applicable to Send-URI. See Section 14 for a more complete description
1198 of each status code.

1199 server-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation
1200 attribute is not supported and the "document-uri" attribute MUST be returned in the Unsupported
1201 Attributes group.

1202 [1.1.1.3.2.3](#) Cancel-Job

1203 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Cancel-
1204 Job with the following specializations and differences. See Section 14 for a more complete description of
1205 each status code.

1206 For the following success status codes, the Job object is being canceled or has been canceled:

1207 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
1208 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
1209 successful-ok-conflicting-attributes: same as Print-Job.
1210

1211 For any of the error status codes, the Job object has not been canceled or was previously canceled.

1212 client-error-not-possible: The request cannot be carried out because of the state of the Job object
1213 ('completed', 'canceled', or 'aborted') or the state of the system.
1214 client-error-not-found: the target Printer and/or Job object does not exist.
1215 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding address is
1216 known.
1217 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
1218 client-error-document-format-not-supported: not applicable.
1219 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1220 and values MUST be ignored.

- 1221 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-
1222 jobs" attribute is not involved.
- 1223 server-error-operation-not-supported: not applicable (Cancel-Job is REQUIRED).
- 1224 server-error-device-error: same as Print-Job, except no document data is involved.
- 1225 server-error-temporary-error: same as Print-Job, except no document data is involved.
- 1226 server-error-not-accepting-jobs: not applicable..
- 1227 server-error-job-canceled: not applicable.
- 1228 [1.1.1.42.3.2.4](#) Get-Job-Attributes
- 1229 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Get-Job-
1230 Attributes with the following specializations and differences. See Section 14 for a more complete
1231 description of each status code.
- 1232 For the following success status codes, the requested attributes are returned in Group 3 in the response:
- 1233 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1234 attributes were unsupported.
- 1235 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1236 operation attribute MAY, but NEED NOT, be returned with the unsupported values.
- 1237 successful-ok-conflicting-attributes: same as Print-Job.
- 1238 For the error status codes, Group 3 is returned containing no attributes or is not returned at all.
- 1239 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.
- 1240 client-error-document-format-not-supported: not applicable.
- 1241 client-error-attributes-or-values-not-supported: not applicable.
- 1242 client-error-uri-scheme-not-supported: not applicable.
- 1243 client-error-conflicting-attributes: not applicable
- 1244 server-error-operation-not-supported: not applicable (since Get-Job-Attributes is REQUIRED).
- 1245 server-error-device-error: same as Print-Job, except no document data is involved.
- 1246 server-error-temporary-error: sane as Print-Job, except no document data is involved..
- 1247 server-error-not-accepting-jobs: not applicable.
- 1248 server-error-job-canceled: not applicable.
- 1249 2.4 Validate-Job
- 1250 The Validate-Job operation has been designed so that its implementation may be a part of the Print-Job
1251 operation. Therefore, requiring Validate-Job is not a burden on implementers. Also it is useful for client's
1252 to be able to count on its presence in all conformance implementations, so that the client can determine
1253 before sending a long document, whether the job will be accepted by the IPP Printer or not.
- 1254 2.5 Case Sensitivity in URIs (issue 1.6)
- 1255 IPP client and server implementations must be aware of the diverse uppercase/lowercase nature of URIs.
1256 RFC 2396 defines URL schemes and Host names as case insensitive but reminds us that the rest of the
1257 URL may well demonstrate case sensitivity. When creating URL's for fields where the choice is

1258 completely arbitrary, it is probably best to select lower case. However, this cannot be guaranteed and
1259 implementations MUST NOT rely on any fields being case-sensitive or case-insensitive in the URL beyond
1260 the URL scheme and host name fields.

1261 The reason that the IPP specification does not make any restrictions on URIs, is so that implementations of
1262 IPP may use off-the-shelf components that conform to the standards that define URIs, such as RFC 2396
1263 and the HTTP/1.1 specifications [RFC2068]. See these specifications for rules of matching, comparison,
1264 and case-sensitivity.

1265 It is also recommended that that System Administrators and implementations avoid creating URLs for
1266 different printers that differ only in their case. For example, don't have Printer1 and printer1 as two
1267 different IPP Printers.

1268 The HTTP/1.1 specification [RFC2068-] contains more details on comparing URLs.

1269 2.6 Character Sets, natural languages, and internationalization

1270 This section discusses character set support, natural language support and internationalization.

1271 ~~1.1.1~~2.6.1 Character set code conversion support (Issue 1.5)

1272 IPP clients and IPP objects are REQUIRED to support UTF-8. They MAY support additional charsets. An
1273 IPP object is required to code covert with as little loss as possible between the charsets that it supports, as
1274 indicated in the Printer's "charsets-supported" attribute.

1275 How should the server handle the situation where the "attributes-charset" of the response itself is "us-ascii",
1276 but one or more attributes in that response is in the "utf-8" format?

1277 Example: Consider a case where a client sends a Print-Job request with "utf-8" as the value of "attributes-
1278 charset" and with the "job-name" attribute supplied. Later another client submits a Get-Job-Attribute or
1279 Get-Jobs request. This second request contains the "attributes-charset" with value "us-ascii" and
1280 "requested-attributes" attribute with exactly one value "job-name".

1281 According to the IPP-Mod document (section 3.1.4.2), the value of the "attributes-charset" for the response
1282 of the second request must be "us-ascii" since that is the charset specified in the request. The "job-name"
1283 value, however, is in "utf-8" format. Should the request be rejected even though both "utf-8" and "us-ascii"
1284 charsets are supported by the server? or should the "job-name" value be converted to "us-ascii" and return
1285 "successful-ok-conflicting-attributes" (0x0002) as the status code?

1286 Answer: An IPP object that supports both utf-8 (REQUIRED) and us-ascii, the second paragraph of
1287 section 3.1.4.2 applies so that the IPP object MUST accept the request, perform code set conversion
1288 between these two charsets with "the highest fidelity possible" and return 'successful-ok', rather than a
1289 warning 'successful-ok-conflicting-attributes, or an error. The printer will do the best it can to convert
1290 between each of the character sets that it supports--even if that means providing a string of question marks
1291 because none of the characters are representable in US ASCII. If it can't perform such conversion, it

1292 MUST NOT advertise us-ascii as a value of its "attributes-charset-supported" and MUST reject any request
1293 that requests 'us-ascii'.

1294 One IPP object implementation strategy is to convert all request text and name values to a Unicode internal
1295 representation. This is 16-bit and virtually universal. Then convert to the specified operation attributes-
1296 charset on output.

1297 Also it would be smarter for a client to ask for 'utf-8', rather than 'us-ascii' and throw away characters that it
1298 doesn't understand, rather than depending on the code conversion of the IPP object.

1299 4.1.2.6.2 What charset to return when an unsupported charset is requested (Issue 1.19)?

1300 Section 3.1.4.1 Request Operation attributes was clarified in November 1998 as follows:

1301 All clients and IPP objects MUST support the 'utf-8' charset [RFC2044] and MAY support
1302 additional charsets provided that they are registered with IANA [IANA-CS]. If the Printer object
1303 does not support the client supplied charset value, the Printer object MUST reject the request, set
1304 the "attributes-charset" to 'utf-8' in the response, and return the 'client-error-charset-not-supported'
1305 status code and any 'text' or 'name' attributes using the 'utf-8' charset.

1306 Since the client and IPP object MUST support UTF-8, returning any text or name attributes in UTF-8 when
1307 the client requests a charset that is not supported should allow the client to display the text or name.

1308 Since such an error is a client error, rather than a user error, the client should check the status code first so
1309 that it can avoid displaying any other returned 'text' and 'name' attributes that are not in the charset
1310 requested.

1311 Furthermore, [ipp-mod] section 14.1.4.14 client-error-charset-not-supported (0x040D) was clarified in
1312 November 1998 as follows:

1313 For any operation, if the IPP Printer does not support the charset supplied by the client in the
1314 "attributes-charset" operation attribute, the Printer MUST reject the operation and return this status
1315 and any 'text' or 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1).

1316 4.1.3.2.6.3 Natural Language Override (NLO) (Issue 1.45)

1317 The 'text' and 'name' attributes each have two forms. One has an implicit natural language, and the other
1318 has an explicit natural language. The 'textWithoutLanguage' and 'textWithLanguage' are the two 'text'
1319 forms. The 'nameWithoutLanguage' and 'nameWithLanguage' are the two 'name' forms. If a receiver (IPP
1320 object or IPP client) supports an attribute with attribute syntax 'text', it MUST support both forms in a
1321 request and a response. A sender (IPP client or IPP object) MAY send either form for any such attribute.
1322 When a sender sends a WithoutLanguage form, the implicit natural language is specified in the "attributes-
1323 natural-language" operation attribute which all senders MUST include in every request and response.

1324 When a sender sends a WithLanguage form, it MAY be different from the implicit natural language
1325 supplied by the sender or it MAY be the same. The receiver MUST treat either form equivalently.

1326 There is an implementation decision for senders, whether to always send the WithLanguage forms or use
1327 the WithoutLanguage form when the attribute's natural language is the same as the request or response.
1328 The former approach makes the sender implementation simpler. The latter approach is more efficient on
1329 the wire and allows inter-working with non-conforming receivers that fail to support the WithLanguage
1330 forms. As each approach have advantages, the choice is completely up to the implementer of the sender.

1331 Furthermore, when a client receives a 'text' or 'name' job attribute that it had previously supplied, that client
1332 MUST NOT expect to see the attribute in the same form, i.e., in the same WithoutLanguage or
1333 WithLanguage form as the client supplied when it created the job. The IPP object is free to transform the
1334 attribute from the WithLanguage form to the WithoutLanguage form and vice versa, as long as the natural
1335 language is preserved. However, in order to meet this latter requirement, it is usually simpler for the IPP
1336 object implementation to store the natural language explicitly with the attribute value, i.e., to store using an
1337 internal representation that resembles the WithLanguage form.

1338 The IPP Printer MUST copy the natural language of a job, i.e., the value of the "attributes-natural-
1339 language" operation attribute supplied by the client in the create operation, to the Job object as a Job
1340 Description attribute, so that a client is able to query it. In returning a Get-Job-Attributes response, the IPP
1341 object MAY return one of three natural language values in the response's "attributes-natural-language"
1342 operation attribute: (1) that requested by the requester, (2) the natural language of the job, or (3) the
1343 configured natural language of the IPP Printer, if the requested language is not supported by the IPP
1344 Printer.

1345 This "attributes-natural-language" Job Description attribute is useful for an IPP object implementation that
1346 prints start sheets in the language of the user who submitted the job. This same Job Description attribute is
1347 useful to a multi-lingual operator who has to communicate with different job submitters in different natural
1348 languages. This same Job Description attribute is expected to be used in the future to generate notification
1349 messages in the natural language of the job submitter.

1350 Early drafts of [IPP-MOD] contained a job-level natural language override (NLO) for the Get-Jobs
1351 response. A job-level (NLO) is an (unrequested) Job Attribute which then specified the implicit natural
1352 language for any other WithoutLanguage job attributes returned in the response for that job.
1353 Interoperability testing of early implementations showed that no one was implementing the job-level NLO
1354 in Get-Job responses. So the job-level NLO was eliminated from the Get-Jobs response. This
1355 simplification makes all requests and responses consistent in that the implicit natural language for any
1356 WithoutLanguage 'text' or 'name' form is always supplied in the request's or response's "attributes-natural-
1357 language" operation attribute.

1358 2.7 The "queued-job-count" Printer Description attribute

1359 2.7.1 Why is "queued-job-count" RECOMMENDED (Issue 1.14)?

1360 The reason that "queued-job-count" is RECOMMENDED, is that some clients look at that attribute alone
1361 when summarizing the status of a list of printers, instead of doing a Get-Jobs to determine the number of
1362 jobs in the queue. Implementations that fail to support the "queued-job-count" will cause that client to
1363 display 0 jobs when there are actually queued jobs.

1364 We would have made it a REQUIRED Printer attribute, but some implementations had already been
1365 completed before the issue was raised, so making it a SHOULD was a compromise.

1366 1.1.2.7.2 Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?

1367 The "queued-job-count" is not a good measure of how busy the printer is when there are held jobs. A
1368 future registration could be to add a "held-job-count" (or an "active-job-count") Printer Description
1369 attribute if experience shows that such an attribute (combination) is needed to quickly indicate how busy a
1370 printer really is.

1371 2.8 Sending empty attribute groups (Issue 1.16)

1372 The [IPP-MOD] and [IPP-PRO] specifications RECOMMEND that a sender not send an empty attribute
1373 group in a request or a response. However, they REQUIRE a receiver to accept an empty attribute group as
1374 equivalent to the omission of that group. allows empty groups to be sent by a "sender" as being entirely
1375 equivalent to omitting the group entirely. The term "sender" means a client for a request and an IPP object
1376 for a response. So a client ~~SHOULD~~~~MAY~~ omit the Job Template Attributes group entirely, ~~or send an~~
1377 ~~empty group~~ in a create operation that is not supplying any Job Template attributes. Similarly, an IPP
1378 object ~~MAY~~~~SHOULD~~ return omit an empty Unsupported Attributes group ~~or MAY omit the group~~
1379 ~~entirely,~~ if there are no unsupported attributes to be returned in a response.

1380 The [IPP-PRO] specification REQUIRES a receiver to be able to receive either an empty attribute group or
1381 an omitted attribute group and treat them equivalently. The term "receiver" means an IPP object for a
1382 request and a client for a response. The term "sender" means a client for a request and an IPP object for a
1383 response.

1384 There is an exception to the rule for Get-Jobs when there are no attributes to be returned. [ipp-pro]
1385 contains the following paragraph:

1386 The syntax allows an xxx-attributes-tag to be present when the xxx-attribute-sequence that follows
1387 is empty. The syntax is defined this way to allow for the response of Get-Jobs where no attributes
1388 are returned for some job-objects. Although it is RECOMMENDED that the sender not send an
1389 xxx-attributes-tag if there are no attributes (except in the Get-Jobs response just mentioned), the
1390 receiver MUST be able to decode such syntax.

1391 ~~Issues 1.16 and 1.17 clarified the [IPP-MOD] to agree with [IPP-PRO] concerning empty attributes being~~
1392 ~~equivalent to omitted attribute groups in requests and responses.~~

1393 2.9 Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)

1394 In the Get-Printer-Attributes, Get-Jobs, or Get-Job-Attributes responses, ~~t~~The client cannot depend on
1395 getting unsupported attributes returned in the Unsupported Attributes group ~~of Get-Printer-Attributes, Get-~~
1396 ~~Jobs, or Get-Job-Attributes responses~~ that the client requested, but are not supported by the IPP object.
1397 However, such unsupported requested attributes will not be returned in the Job Attributes or Printer
1398 Attributes group (since they are unsupported). ~~However~~Furthermore, the IPP object is REQUIRED to

1399 return the 'successful-ok-ignored-or-substituted-attributes' status code, so that the client knows that not all
1400 that was requested has ~~not~~ been returned.

1401 2.10 Returning job-state in Print-Job response (Issue 1.30)

1402 An IPP client submits a small job via Print-Job. By the time the IPP printer/print server is putting together
1403 a response to the operation, the job has finished printing and been removed as an object from the print
1404 system. What should the job-state be in the response?

1405 The Model suggests that the Printer return a response before it even accepts the document content (~~see~~
1406 [Implementer's Guide](#)). The Job Object Attributes are returned only if the IPP object returns one of the
1407 success status codes. Then the job-state would always be "pending" or "pending-held".

1408 This issue comes up for the implementation of an IPP Printer object as a server that forwards jobs to
1409 devices that do not provide job status back to the server. If the server is reasonably certain that the job
1410 completed successfully, then it should return the job-state as 'completed'. Also the server can keep the job
1411 in its "job history" long after the job is no longer in the device. Then a user could query the server and see
1412 that the job was in the 'completed' state and completed as specified by the job's "time-at-completed" time
1413 which would be the same as the server submitted the job to the device.

1414 An alternative is for the server to respond to the client before or while sending the job to the device, instead
1415 of waiting until the server has finished sending the job to the device. In this case, the server can return the
1416 job's state as 'pending' with the 'job-outgoing' value in the job's "job-state-reasons" attribute.

1417 If the server doesn't know for sure whether the job completed successfully (or at all), it could return the
1418 (out-of-band) 'unknown' value.

1419 On the other hand, if the server is able to query the device and/or setup some sort of event notification that
1420 the device initiates when the job makes state transitions, then the server can return the current job state in
1421 the Print-Job response and in subsequent queries because the server knows what the job state is in the
1422 device (or can query the device).

1423 All of these alternatives depend on implementation of the server and the device.

1424 2.11 Flow controlling the data portion of a Print-Job request (Issue 1.22)

1425 A paused printer (or one that is stopped due to paper out or jam or spool space full or buffer space full, may
1426 flow control the data of a Print-Job operation (at the TCP/IP layer), so that the client is not able to send all
1427 the document data. Consequently, the Printer will not return a response until the condition is changed.

1428 The Printer should not return a Print-Job response with an error code in any of these conditions, since either
1429 the printer will be resumed and/or the condition will be freed either by human intervention or as jobs print.

1430 In writing test scripts to test IPP Printers, the script must also be written not to expect a response, if the
1431 printer has been paused, until the printer is resumed, in order to work with all possible implementations.

1432 2.12 Multi-valued attributes (Issue 1.31)

1433 What is the attribute syntax for a multi-valued attribute? Since some attributes support values in more than
1434 one data type, such as "media", "job-hold-until", and "job-sheets", IPP semantics associate the attribute
1435 syntax with each value, not with the attribute as a whole. The protocol associates the attribute syntax tag
1436 with each value. Don't be fooled, just because the attribute syntax tag comes before the attribute keyword.
1437 All attribute values after the first have a zero length attribute keyword as the indication of a subsequent
1438 value of the same attribute.

1439 2.13 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)

1440 The following clarification was added to [ipp-mod] section 8.5:

1441 8.5 Queries on jobs submitted using non-IPP protocols

1442 If the device that an IPP Printer is representing is able to accept jobs using other job submission
1443 protocols in addition to IPP, it is RECOMMEND that such an implementation at least allow such
1444 "foreign" jobs to be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an
1445 implementation NEED NOT support all of the same IPP job attributes as for IPP jobs. The IPP
1446 object returns the 'unknown' out-of-band value for any requested attribute of a foreign job that is
1447 supported for IPP jobs, but not for foreign jobs.

1448 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such
1449 "foreign jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-
1450 Attributes and Cancel-Job. Such an implementation also needs to deal with the problem of
1451 authentication of such foreign jobs. One approach would be to treat all such foreign jobs as
1452 belonging to users other than the user of the IPP client. Another approach would be for the foreign
1453 job to belong to 'anonymous'. Only if the IPP client has been authenticated as an operator or
1454 administrator of the IPP Printer object, could the foreign jobs be queried by an IPP request.
1455 Alternatively, if the security policy is to allow users to query other users' jobs, then the foreign jobs
1456 would also be visible to an end-user IPP client using Get-Jobs and Get-Job-Attributes.

1457 Thus IPP MAY be implemented as a "universal" protocol that provides access to jobs submitted with any
1458 job submission protocol. As IPP becomes widely implemented, providing a more universal access makes
1459 sense.

1460 2.14 The 'none' value for empty sets (Issue 1.37)

1461 [ipp-mod] states that the 'none' value should be used as the value of a 1SetOf when the set is empty. In most
1462 cases, sets that are potentially empty contain keywords so the keyword 'none' is used, but for the 3
1463 finishings attributes, the values are enums and thus the empty set is represented by the enum 3. Currently
1464 there are no other attributes with 1SetOf values which can be empty and can contain values that are not
1465 keywords. This exception requires special code and is a potential place for bugs. It would have been better
1466 if we had chosen an out-of-band value, either "no-value" or some new value, such as 'none'. Since we
1467 didn't, implementations have to deal with the different representations of 'none', depending on the attribute
1468 syntax.

1469 2.15 Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?

1470 In [ipp-mod] section 3.2.6.1 'Get-Jobs Request', if the attribute 'my-jobs' is present and set to TRUE, MUST
1471 the 'requesting-user-name' attribute be there to, and if it's not present what should the IPP printer do?

1472 [ipp-mod] Section 8.3 describes the various cases of "requesting-user-name" being present or not for any
1473 operation. If the client does not supply a value for "requesting-user-name", the printer MUST assume that
1474 the client is supplying some anonymous name, such as "anonymous".

1475 2.16 The "multiple-document-handling" Job Template attribute and support of multiple document jobs

1476 ISSUE: IPP/1.0 is silent on which of the four effects an implementation would perform if it supports
1477 Create-Job, but does not support "multiple-document-handling".

1478 A fix to IPP/1.0 would be to require implementing all four values of "multiple-document-handling" if
1479 Create-Job is supported at all. Or at least 'single-document-new-sheet' and 'separate-documents-uncollated-
1480 copies'. In any case, an implementation that supports Create-Job SHOULD also support "multiple-
1481 document-handling". Support for all four values is RECOMMENDED, but at least the 'single-document-
1482 new-sheet' and 'separate-documents-uncollated-copies' values, along with the "multiple-document-
1483 handling-default" indicating the default behavior and "multiple-document-handling-supported" values. If
1484 an implementation spools the data, it should also support the 'separate-documents-collated-copies' value as
1485 well.

1486 3 Encoding and Transport

1487 This section discusses various aspects of IPP/1.0 Encoding and Transport [IPP-PRO].

1488 HTTP/1.1 is the transport layer for this protocol.

1489 The operation layer has been designed with the assumption that the transport layer contains the following
1490 information:

- 1491 □ the URI of the target job or printer operation
- 1492 □ the total length of the data in the operation layer, either as a single length or as a sequence of chunks
1493 each with a length.

1494 It is REQUIRED that a printer implementation support HTTP over the IANA assigned Well Known Port
1495 631 (the IPP default port), though a printer implementation may support HTTP over port some other port as
1496 well. In addition, a printer may have to support another port for privacy (See Section 5 "Security
1497 Considerations").

1498 Note: even though port 631 is the IPP default, port 80 remains the default for an HTTP URI. Thus a URI
1499 for a printer using port 631 MUST contain an explicit port, e.g. "http://forest:631/pinetrete".

1500 Note: Consistent with RFC 2068 (HTTP/1.1), HTTP URI's for IPP implicitly reference port 80. If a URI
1501 references some other port, the port number MUST be explicitly specified in the URI.

1502 ~~Each HTTP operation MUST use the POST method where the request URI is the object target of the~~
1503 ~~operation, and where the "Content-Type" of the message body in each request and response MUST be~~
1504 ~~"application/ipp". The message body MUST contain the operation layer and MUST have the syntax~~
1505 ~~described in section 3.2 "Syntax of Encoding". A client implementation MUST adhere to the rules for a~~
1506 ~~client described in RFC 2068 [rfe2068]. A printer (server) implementation MUST adhere the rules for an~~
1507 ~~origin server described in RFC 2068.~~

1508 The IPP layer doesn't have to deal with chunking. In the context of CGI scripts, the HTTP layer removes
1509 any chunking information in the received data.

1510 A client MUST NOT expect a response from an IPP server until after the client has sent the entire response.
1511 But a client MAY listen for an error response that an IPP server MAY send before it receives all the data.
1512 In this case a client, if chunking the data, can send a premature zero-length chunk to end the request before
1513 sending all the data. If the request is blocked for some reason, a client MAY determine the reason by
1514 opening another connection to query the server.

1515 In the following sections, there are a tables of all HTTP headers which describe their use in an IPP client or
1516 server. The following is an explanation of each column in these tables.

- 1517 • the "header" column contains the name of a header
- 1518 • the "request/client" column indicates whether a client sends the header.
- 1519 • the "request/ server" column indicates whether a server supports the header when received.
- 1520 • the "response/ server" column indicates whether a server sends the header.
- 1521 • the "response /client" column indicates whether a client supports the header when received.
- 1522 • the "values and conditions" column specifies the allowed header values and the conditions for the
1523 header to be present in a request/response.

1524 The table for "request headers" does not have columns for responses, and the table for "response headers"
1525 does not have columns for requests.

1526 The following is an explanation of the values in the "request/client" and "response/ server" columns.

- 1527 • **must:** the client or server MUST send the header,
- 1528 • **must-if:** the client or server MUST send the header when the condition described in the "values and
1529 conditions" column is met,
- 1530 • **may:** the client or server MAY send the header
- 1531 • **not:** the client or server SHOULD NOT send the header. It is not relevant to an IPP
1532 implementation.

1533 The following is an explanation of the values in the "response/client" and "request/ server" columns.

- 1534 • **must:** the client or server MUST support the header,
- 1535 • **may:** the client or server MAY support the header
- 1536 • **not:** the client or server SHOULD NOT support the header. It is not relevant to an IPP
1537 implementation.

1538 3.1 General Headers

1539 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 2068 [RFC2068]
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	

1540 [4.23.2](#) Request Headers

1541 The following is a table for the request headers.

Request-Header	Client	Server	Request Values and Conditions
Accept	may	must	“application/ipp” only. This value is the default if the client omits it
Accept-Charset	not	not	Charset information is within the application/ipp entity
Accept-Encoding	may	must	empty and per RFC 2068 [RFC2068] and IANA registry for content-codings
Accept-Language	not	not	language information is within the application/ipp entity
Authorization	must-if	must	per RFC 2068. A client MUST send this header when it receives a 401 “Unauthorized” response and does not receive a “Proxy-Authenticate”

Request-Header	Client	Server	Request Values and Conditions header.
From	not	not	per RFC 2068. Because RFC recommends sending this header only with the user's approval, it is not very useful
Host	must	must	per RFC 2068
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 2068. A client MUST send this header when it receives a 401 "Unauthorized" response and a "Proxy-Authenticate" header.
Range	not	not	
Referer	not	not	
User-Agent	not	not	

1542 [4.3.3](#) Response Headers

1543 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 2068. When URI needs redirection.
Proxy-Authenticate	not	must	per RFC 2068
Public	may	may	per RFC 2068
Retry-After	may	may	per RFC 2068
Server	not	not	

Response-Header	Server	Client	Response Values and Conditions
Vary	not	not	
Warning	may	may	per RFC 2068
WWW-Authenticate	must-if	must	per RFC 2068. When a server needs to authenticate a client.

1544 [4.43.4](#) Entity Headers

1545 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 2068 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 2068. Header MUST be present if Transfer-Encoding is absent..
Content-Location	not	not	not	not	
Content-MD5	may	may	may	may	per RFC 2068
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	

1546 [4.53.5](#) Optional support for HTTP/1.0

1547 IPP implementations consist of an HTTP layer and an IPP layer. In the following discussion, the term
 1548 "client" refers to the HTTP client layer and the term "server" refers to the HTTP server layer. The

1549 Encoding and Transport document [IPP-PRO] requires that HTTP 1.1 MUST be supported by all clients
1550 and all servers. However, a client and/or a server implementation may choose to also support HTTP 1.0.

1551 • This option means that a server may choose to communicate with a (non-conforming) client that only
1552 supports HTTP 1.0. In such cases the server should not use any HTTP 1.1 specific parameters or
1553 features and should respond using HTTP version number 1.0.

1554 • This option also means that a client may choose to communicate with a (non-conforming) server that
1555 only supports HTTP 1.0. In such cases, if the server responds with an HTTP 'unsupported version
1556 number' to an HTTP 1.1 request, the client should retry using HTTP version number 1.0.

1557 3.6 HTTP/1.1 Chunking

1558 Clients ~~should~~**MUST** anticipate that the HTTP/1.1 server may chunk responses and ~~should~~**MUST** accept
1559 them in responses. However, a (non-conforming) HTTP client that is unable to accept chunked responses
1560 may attempt to request an HTTP 1.1 server not to ~~not~~ use chunking in its response to an operation by using
1561 the following HTTP header:

1562 TE: identity

1563 This mechanism should not be used by a server to disable a client from chunking a request, since chunking
1564 of document data is an important feature for clients to send long documents.

1565

1566 4 References

1567 ~~[IPP-IG]~~

1568 ~~———— Hastings, T., Manros, C., "Internet Printing Protocol/1.0: Implementer's Guide", draft-ietf-ipp-~~
1569 ~~implementors-guide-00.txt, November 1998.~~

1570 [IPP-LPD]

1571 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", draft-
1572 ietf-ipp-lpd-ipp-map-04.txt, June 1998.

1573 [IPP-MOD]

1574 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
1575 Semantics", draft-ietf-ipp-model-11.txt, November, 1998.

1576 [IPP-PRO]

1577 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and
1578 Transport", draft-ietf-ipp-pro-06.txt, June, 1998.

1579 [IPP-RAT]

1580 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
1581 draft-ietf-ipp-rat-03.txt, June, 1998.

- 1582 [IPP-REQ]
1583 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-02.txt, June, 1998.
- 1584 [RFC1123]
1585 Braden, S., "Requirements for Internet Hosts - Application and Support", RFC 1123, October, 1989.
- 1586 [RFC2068]
1587 R Fielding, et al, "Hypertext Transfer Protocol – HTTP/1.1" RFC 2068, January 1997.
- 1588 [rfc2119]
1589 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March
1590 1997.
- 1591 [RFC2396]
1592 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic
1593 Syntax", RFC 2396, August 1998.
- 1594 [SSL]
1595 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.
- 1596

1597 4.4.1 Authors' Address

1598 Thomas N. Hastings
1599 Xerox Corporation
1600 701 Aviation Blvd.
1601 El Segundo, CA 90245
1602 hastings@cp10.es.xerox.com
1603

1604 Carl-Uno Manros
1605 Xerox Corporation
1606 701 Aviation Blvd.
1607 El Segundo, CA 90245
1608 manros@cp10.es.xerox.com

1609 **5 Appendix C: Full Copyright Statement**

1610 Copyright (C) The Internet Society (1998). All Rights Reserved

1611 This document and translations of it may be copied and furnished to others, and derivative works that
1612 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
1613 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
1614 and this paragraph are included on all such copies and derivative works. However, this document itself
1615 may not be modified in any way, such as by removing the copyright notice or references to the Internet
1616 Society or other Internet organizations, except as needed for the purpose of developing Internet standards

1617 in which case the procedures for copyrights defined in the Internet Standards process must be followed, or
1618 as required to translate it into languages other than English.

1619 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its
1620 successors or assigns.

1621 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET
1622 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,
1623 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE
1624 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
1625 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1626 **6 Change History**

1627 The change history is in reverse chronological order:

1628 ~~4.1~~6.1 Changes to produce the December 6, 1998 version from the November 16, 1998 version:

1629 Included all of the remaining agreed issues raised before the November 16, 1998 production of the Internet-
1630 Drafts for IPP/1.0 that included adding explanations to the Implementers Guide.

1631