

1 INTERNET-DRAFT
2 draft-ietf-ipp-implementers-guide-01.txt

T. Hastings
Xerox Corporation
C. Manros
Xerox Corporation

~~December 6~~ February 12, 1999

9 Internet Printing Protocol/1.0: Implementer's Guide

10 Copyright (C) The Internet Society (~~date~~ 1999). All Rights Reserved.

11 Status of this Memo

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

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

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

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

21 ~~To learn the current status of any Internet Draft, please check the "1id-abstracts.txt" listing contained in the~~
22 ~~Internet Drafts Shadow Directories on [ftp.is.co.za](ftp://ftp.is.co.za) (Africa), [ftp.nordu.net](ftp://ftp.nordu.net) (Europe), [munnari.oz.au](ftp://ftp.munnari.oz.au) (Pacific~~
23 ~~Rim), [ftp.ietf.org](ftp://ftp.ietf.org) (US East Coast), or [ftp.isi.edu](ftp://ftp.isi.edu) (US West Coast).~~

24

25 Abstract

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

33 The full set of IPP documents includes:

34 Design Goals for an Internet Printing Protocol [IPP-REQ]

35 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [IPP-RAT]

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

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

38 Mapping between LPD and IPP Protocols [IPP LPD]

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

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

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

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

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

56

TABLE OF CONTENTS

57	1	Introduction	5
58	1.1	Conformance language	5
59	1.2	Other terminology	5
60	2	Model and Semantics	5
61	2.1	Summary of Operation Attributes.....	5
62	2.2	Suggested Operation Processing Steps for IPP Objects (Issue 1.21).....	9
63	2.2.1	Suggested Operation Processing Steps for all Operations	9
64	2.2.1.1	Validate version number	9
65	2.2.1.2	Validate operation identifier	10
66	2.2.1.3	Validate the request identifier	10
67	2.2.1.4	Validate attribute group and attribute presence and order	10
68	2.2.1.5	Validate the values of the REQUIRED Operation attributes.....	17
69	2.2.1.6	Validate the values of the OPTIONAL Operation attributes	21
70	2.2.2	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add	
71		Documents.....	23
72	2.2.2.1	Default "ipp-attribute-fidelity" if not supplied.....	23
73	2.2.2.2	Check that the Printer object is accepting jobs	23
74	2.2.2.3	Validate the values of the Job Template attributes	23
75	2.2.3	Algorithm for job validation	24
76	2.2.3.1	Check for conflicting Job Template attributes values.....	28
77	2.2.3.2	Decide whether to REJECT the request.....	29
78	2.2.3.3	For the Validate-Job operation, RETURN one of the success status codes	29
79	2.2.3.4	Create the Job object with attributes to support.....	29
80	2.2.3.5	Return one of the success status codes.....	30
81	2.2.3.6	Accept appended Document Content.....	31
82	2.2.3.7	Scheduling and Starting to Process the Job	31
83	2.2.3.8	Completing the Job	31
84	2.2.3.9	Destroying the Job after completion	31
85	2.2.3.10	Interaction with "ipp-attribute-fidelity"	31
86	2.3	Status codes returned by operation (Issue 1.50)	32
87	2.3.1	Printer Operations	32
88	2.3.1.1	Print-Job.....	32
89	2.3.1.2	Print-URI.....	34
90	2.3.1.3	Validate-Job	34
91	2.3.1.4	Create-Job	34
92	2.3.1.5	Get-Printer-Attributes	34
93	2.3.1.6	Get-Jobs	35
94	2.3.2	Job Operations.....	35
95	2.3.2.1	Send-Document.....	35
96	2.3.2.2	Send-URI	36
97	2.3.2.3	Cancel-Job.....	36
98	2.3.2.4	Get-Job-Attributes.....	37

99	2.4	Validate-Job	37
100	2.5	Case Sensitivity in URIs (issue 1.6).....	37
101	2.6	Character Sets, natural languages, and internationalization	38
102	2.6.1	Character set code conversion support (Issue 1.5).....	38
103	2.6.2	What charset to return when an unsupported charset is requested (Issue 1.19)?.....	39
104	2.6.3	Natural Language Override (NLO) (Issue 1.45).....	39
105	2.7	The "queued-job-count" Printer Description attribute	40
106	2.7.1	Why is "queued-job-count" RECOMMENDED (Issue 1.14)?.....	40
107	2.7.2	Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?.....	41
108	2.8	Sending empty attribute groups (Issue 1.16)	41
109	2.9	Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)	41
110	2.10	Returning job-state in Print-Job response (Issue 1.30)	42
111	2.11	Flow controlling the data portion of a Print-Job request (Issue 1.22).....	42
112	2.12	Multi-valued attributes (Issue 1.31).....	43
113	2.13	Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)..	43
114	2.14	The 'none' value for empty sets (Issue 1.37).....	43
115	2.15	Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?.....	44
116	2.16	The "multiple-document-handling" Job Template attribute and support of multiple document jobs	
117		44	
118	3	Encoding and Transport.....	44
119	3.1	General Headers	45
120	3.2	Request Headers.....	46
121	3.3	Response Headers	47
122	3.4	Entity Headers.....	47
123	3.5	Optional support for HTTP/1.0.....	48
124	3.6	HTTP/1.1 Chunking.....	48
125	3.6.1	Disabling IPP Server Response Chunking.....	48
126	3.6.2	Warning About the Support of Chunked Requests.....	49
127	4	References	49
128	4.1	Authors' Address	50
129	5	Appendix C: Full Copyright Statement Notices	51
130	6	Change History.....	51
131	6.1	Changes to produce the February 12, 1999 version from the January 8, 1999 version:.....	51
132	6.2	Changes to produce the January 8, 1999 version from the December 6, 1998 version:.....	52
133	6.3	Changes to produce the December 6, 1998 version from the November 16, 1998 version:	52
134			

135

136 **1 Introduction**

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

144 1.1 Conformance language

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

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

156 1.2 Other terminology

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

159 **2 Model and Semantics**

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

161 2.1 Summary of Operation Attributes

162 Legend for the following table:

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

164 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	

165

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.

166

167

168 2.2 Suggested Operation Processing Steps for IPP Objects (Issue 1.21)

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

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

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

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

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

195 2.2.1 Suggested Operation Processing Steps for all Operations

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

199 2.2.1.1 Validate version number

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

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

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

214 2.2.1.2 Validate operation identifier

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

218 2.2.1.3 Validate the request identifier

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

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

224 2.2.1.4 Validate attribute group and attribute presence and order

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

226 2.2.1.4.1 Validate the presence and order of attribute groups

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

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

235 Since this kind of attribute group error is most likely to be an error detected by a client developer rather
236 than by a customer, the IPP object NEED NOT return an indication of which attribute group was in error in

237 either the Unsupported Attributes group or the Status Message. Also, the IPP object NEED NOT find all
238 attribute group errors before returning this error.

239 2.2.1.4.2 Ignore unknown attribute groups in the expected position

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

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

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

252 2.2.1.4.3 Validate the presence of a single occurrence of required Operation attributes

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

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

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

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

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

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

280 R indicates a REQUIRED attribute that an IPP object MUST support
281 O indicates an OPTIONAL attribute that an IPP object NEED NOT support
282 * indicates that a client MAY omit the attribute in a request and that an IPP object MAY omit
283 the attribute in a response. The absence of an * means that a client MUST supply the
284 attribute in a request and an IPP object MUST supply the attribute in a response.
285

286

Operation Requests

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

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

290

291 Print-Job Request:
292 Group 1: Operation Attributes (R)
293 attributes-charset (R)
294 attributes-natural-language (R)
295 printer-uri (R)
296 requesting-user-name (R*)
297 job-name (R*)
298 ipp-attribute-fidelity (R*)
299 document-name (R*)
300 document-format (R*)
301 document-natural-language (O*)
302 compression (O*)
303 job-k-octets (O*)
304 job-impressions (O*)
305 job-media-sheets (O*)
306 Group 2: Job Template Attributes (R*)
307 <Job Template attributes> (O*)
308 (see [IPP-MOD] Section 4.2)
309 Group 3: Document Content (R)
310 <document content>
311
312 Validate-Job Request:
313 Group 1: Operation Attributes (R)
314 attributes-charset (R)
315 attributes-natural-language (R)
316 printer-uri (R)
317 requesting-user-name (R*)
318 job-name (R*)
319 ipp-attribute-fidelity (R*)
320 document-name (R*)
321 document-format (R*)
322 document-natural-language (O*)
323 compression (O*)
324 job-k-octets (O*)
325 job-impressions (O*)
326 job-media-sheets (O*)
327 Group 2: Job Template Attributes (R*)
328 <Job Template attributes> (O*)
329 (see [IPP-MOD] Section 4.2)
330
331 Create-Job Request:
332 Group 1: Operation Attributes (R)
333 attributes-charset (R)
334 attributes-natural-language (R)
335 printer-uri (R)
336 requesting-user-name (R*)
337 job-name (R*)

338 ipp-attribute-fidelity (R*)
339 job-k-octets (O*)
340 job-impressions (O*)
341 job-media-sheets (O*)
342 Group 2: Job Template Attributes (R*)
343 <Job Template attributes> (O*) (see
344 (see [IPP-MOD] Section 4.2)
345
346 Print-URI Request:
347 Group 1: Operation Attributes (R)
348 attributes-charset (R)
349 attributes-natural-language (R)
350 printer-uri (R)
351 document-uri (R)
352 requesting-user-name (R*)
353 job-name (R*)
354 ipp-attribute-fidelity (R*)
355 document-name (R*)
356 document-format (R*)
357 document-natural-language (O*)
358 compression (O*)
359 job-k-octets (O*)
360 job-impressions (O*)
361 job-media-sheets (O*)
362 Group 2: Job Template Attributes (R*)
363 <Job Template attributes> (O*) (see
364 (see [IPP-MOD] Section 4.2)
365
366 Send-Document Request:
367 Group 1: Operation Attributes (R)
368 attributes-charset (R)
369 attributes-natural-language (R)
370 (printer-uri & job-id) | job-uri (R)
371 last-document (R)
372 requesting-user-name (R*)
373 document-name (R*)
374 document-format (R*)
375 document-natural-language (O*)
376 compression (O*)
377 Group 2: Document Content (R*)
378 <document content>
379
380 Send-URI Request:
381 Group 1: Operation Attributes (R)
382 attributes-charset (R)
383 attributes-natural-language (R)
384 (printer-uri & job-id) | job-uri (R)

385 last-document (R)
386 document-uri (R)
387 requesting-user-name (R*)
388 document-name (R*)
389 document-format (R*)
390 document-natural-language (O*)
391 compression (O*)
392
393 Cancel-Job Request:
394 Group 1: Operation Attributes (R)
395 attributes-charset (R)
396 attributes-natural-language (R)
397 (printer-uri & job-id) | job-uri (R)
398 requesting-user-name (R*)
399 message (O*)
400
401 Get-Printer-Attributes Request:
402 Group 1: Operation Attributes (R)
403 attributes-charset (R)
404 attributes-natural-language (R)
405 printer-uri (R)
406 requesting-user-name (R*)
407 requested-attributes (R*)
408 document-format (R*)
409
410 Get-Job-Attributes Request:
411 Group 1: Operation Attributes (R)
412 attributes-charset (R)
413 attributes-natural-language (R)
414 (printer-uri & job-id) | job-uri (R)
415 requesting-user-name (R*)
416 requested-attributes (R*)
417
418 Get-Jobs Request:
419 Group 1: Operation Attributes (R)
420 attributes-charset (R)
421 attributes-natural-language (R)
422 printer-uri (R)
423 requesting-user-name (R*)
424 limit (R*)
425 requested-attributes (R*)
426 which-jobs (R*)
427 my-jobs (R*)
428

429 Operation Responses

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

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

433

434 Print-Job Response:

435 Print-URI Response:

436 Create-Job Response:

437 Send-Document Response:

438 Send-URI Response:

439 Group 1: Operation Attributes (R)

440 attributes-charset (R)

441 attributes-natural-language (R)

442 status-message (O*)

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

444 <unsupported attributes> (R*)

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

446 job-uri (R)

447 job-id (R)

448 job-state (R)

449 job-state-reasons (O*)

450 job-state-message (O*)

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

452

453 Validate-Job Response:

454 Cancel-Job Response:

455 Group 1: Operation Attributes (R)

456 attributes-charset (R)

457 attributes-natural-language (R)

458 status-message (O*)

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

460 <unsupported attributes> (R*)

461

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

464

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

467
468 Get-Printer-Attributes Response:
469 Group 1: Operation Attributes (R)
470 attributes-charset (R)
471 attributes-natural-language (R)
472 status-message (O*)
473 Group 2: Unsupported Attributes (R*) (see Note 4)
474 <unsupported attributes> (R*)
475 Group 3: Printer Object Attributes(R*) (see Note 2)
476 <requested attributes> (R*)
477

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

480
481 Get-Job-Attributes Response:
482 Group 1: Operation Attributes (R)
483 attributes-charset (R)
484 attributes-natural-language (R)
485 status-message (O*)
486 Group 2: Unsupported Attributes (R*) (see Note 4)
487 <unsupported attributes> (R*)
488 Group 3: Job Object Attributes(R*) (see Note 2)
489 <requested attributes> (R*)
490

491 Get-Jobs Response:
492 Group 1: Operation Attributes (R)
493 attributes-charset (R)
494 attributes-natural-language (R)
495 status-message (O*)
496 Group 2: Unsupported Attributes (R*) (see Note 4)
497 <unsupported attributes> (R*)
498 Group 3: Job Object Attributes(R*) (see Note 2, 5)
499 <requested attributes> (R*)
500

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

503 2.2.1.5 Validate the values of the REQUIRED Operation attributes

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

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

- 508 a) that the length of each Operation attribute value is correct for the attribute syntax tag supplied
509 by the client according to [IPP-MOD] Section 4.1,
- 510 b) that the attribute syntax tag is correct for that Operation attribute according to [IPP-MOD]
511 Section 3,
- 512 c) that the value is in the range specified for that Operation attribute according to [IPP-MOD]
513 Section 3,
- 514 d) that multiple values are supplied by the client only for operation attributes that are multi-valued,
515 i.e., that are 1setOf X according to [IPP-MOD] Section 3.

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

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

528 -----

529 attributes-charset (charset)

530 IF NOT **any** single non-empty 'charset' value, REJECT/RETURN 'client-error-request-bad-request'.
531 IF the value length is greater than ~~less than or equal to~~ 63 octets, REJECT/RETURN 'client-error-
532 request-value-too-long'.

533 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-charset-
534 not-supported".
535

536 attributes-natural-language(naturalLanguage)

537 IF NOT **any** single non-empty 'naturalLanguage' value, REJECT/RETURN 'client-error-request-bad-
538 request'.

539 IF the value length is greater than ~~less than or equal to~~ 63 octets, REJECT/RETURN 'client-error-
540 request-value-too-long'.

541 ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-
542 language-supported" attribute. If the supplied value is not a member of the Printer object's
543 "generated-natural-language-supported" attribute, use the Printer object's "natural-language-
544 configured" value.
545

546 requesting-user-name

547 IF NOT **any** single 'name' value, REJECT/RETURN 'client-error-request-bad-request'.
548 IF the value length is greater than ~~less than or equal to~~ 255 octets, REJECT/RETURN 'client-error-
549 request-value-too-long'.

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

551

552 job-name(name)

553 IF NOT **any** single 'name' value, REJECT/RETURN 'client-error-request-bad-request'.
554 IF the value length is greater than ~~less than or equal to~~ 255 octets, REJECT/RETURN 'client-error-
555 request-value-too-long'.

556 IF NOT supplied by the client, the Printer object creates a name from the document-name or document-
557 uri.

558

559 document-name (name)

560 IF NOT **any** single 'name' value, REJECT/RETURN 'client-error-request-bad-request'.
561 IF the value length is greater than ~~less than or equal to~~ 255 octets, REJECT/RETURN 'client-error-
562 request-value-too-long'.

563

564 ipp-attribute-fidelity (boolean)

565 IF ~~NOT either~~ NEITHER a single 'true' NOR a single 'false' 'boolean' value ~~equal to 1 octet~~,
566 REJECT/RETURN 'client-error-bad-request'.

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

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

569

570 document-format (mimeType)

571 IF NOT **any** single non-empty 'mimeType' value, REJECT/RETURN 'client-error-request-bad-
572 request'.

573 IF the value length is greater than ~~less than or equal to~~ 255 octets, REJECT/RETURN 'client-error-
574 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, REJECT/RETURN 'client-error-request-bad-request'.

582 IF the value length is greater than ~~less than or equal to~~ 1023 octets, REJECT/RETURN 'client-error-
583 request-value-too-long'.

584 IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.

585 IF scheme is NOT in the Printer object's "reference-uri-schemes-supported" attribute,
586 REJECT/RETURN 'client-error²-uri-scheme-not-supported'.

587 The Printer object MAY check to see if the document exists and is accessible. If the document is not
588 found or is not accessible, REJECT/RETURN 'client-error-not found'.

589 last-document (boolean)

590 IF ~~NOT either~~ **NEITHER** a single 'true' ~~or~~ **NOR a single** 'false' 'boolean' value ~~equal to 1 octet,~~
591 REJECT/RETURN 'client-error-bad-request'.

592 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'
593

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

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

597 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-error-
598 gone' status code, if keep track of recently deleted jobs.
599

600 requested-attributes (1setOf keyword)

601 IF NOT ~~any number of~~ **one or more** 'keyword' values, REJECT/RETURN 'client-error-request-bad-
602 request'.

603 IF the value length is greater than ~~less than or equal to~~ 255 octets, REJECT/RETURN 'client-error-
604 request-value-too-long'.

605 Ignore unsupported values which are the keyword names of unsupported attributes. Don't bother to
606 copy such requested (unsupported) attributes to the Unsupported Attribute response group since the
607 response will not return them.
608

609 which-jobs (type2 keyword)

610 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-request-bad-request'.
611 IF the value length is greater than ~~less than or equal to~~ 255 octets, REJECT/RETURN 'client-error-
612 request-value-too-long'.

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

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

618 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.
619

620 my-jobs (boolean)

621 IF ~~NOT either~~ **NEITHER** a single 'true' ~~or~~ **NOR a single** 'false' 'boolean' value ~~equal to 1 octet,~~
622 REJECT/RETURN 'client-error-bad-request'.

623 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'
624 IF NOT supplied by the client, the IPP object assumes the 'false' value.
625

626 limit (integer(1:MAX))

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

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

630

631 -----

632

633 2.2.1.6 Validate the values of the OPTIONAL Operation attributes

634 OPTIONAL Operation attributes are those that an IPP object MAY or MAY NOT support. An IPP object
635 validates the values of the OPTIONAL attributes supplied by the client. The IPP object performs the same
636 syntactic validation checks for each OPTIONAL attribute value as in Section 2.2.1.5. As in Section
637 2.2.1.5, if any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' or the
638 'client-error-request-value-too-long' status code.

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

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

646 -----

647 document-natural-language (naturalLanguage)

648 IF NOT **any** single non-empty 'naturalLanguage' value, **REJECT/RETURN 'client-error-request-bad-**
649 **request'**.

650 **IF the value length is greater than**~~less than or equal to~~ 63 octets, REJECT/RETURN 'client-error-
651 request-value-too-long'.

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

654

655 compression (type3 keyword)

656 IF NOT **any** single 'keyword' values, **REJECT/RETURN 'client-error-request-bad-request'**.

657 **IF the value length is greater than**~~less than or equal to~~ 255 octets, REJECT/RETURN 'client-error-
658 request-value-too-long'.

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

662

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

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

665 REJECT/RETURN 'client-error-bad-request'.
666 IF NOT in the range of the Printer object's "job-k-octets-supported" attribute, copy the attribute and the
667 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-
668 error-attributes-or-values-not-supported'.
669

670 job-impressions (integer(0:MAX))
671 IF NOT **any** single 'integer' value equal to 4 octets,
672 REJECT/RETURN 'client-error-bad-request'.
673 IF NOT in the range of the Printer object's "job-impressions-supported" attribute, copy the attribute and
674 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-
675 error-attributes-or-values-not-supported'.
676

677 job-media-sheets (integer(0:MAX))
678 IF NOT **any** single 'integer' value equal to 4 octets,
679 REJECT/RETURN 'client-error-bad-request'.
680 IF NOT in the range of the Printer object's "job-media-~~sheets~~-supported" attribute, copy the attribute
681 and the unsupported value to the Unsupported Attributes response group and REJECT/RETURN
682 'client-error-attributes-or-values-not-supported'.
683

684 message (text(127))
685 IF NOT **any** single 'text' value, **REJECT/RETURN 'client-error-request-bad-request'.**
686 **IF the value length is greater than**~~less than or equal to~~ 127 octets,
687 REJECT/RETURN 'client-error-request-value-too-long'.
688

689 unknown or unsupported attribute
690 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute
691 syntax, REJECT/RETURN 'client-error-request-value-too-long'.
692 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
693 attribute value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.
694

695 Note: Future Operation attributes may be added to the protocol specification that may occur anywhere
696 in the specified group. When the operation is otherwise successful, the IPP object returns the
697 'successful-ok-ignored-or-substituted-attributes' status code. Ignoring unsupported Operation attributes
698 in all operations is analogous to the handling of unsupported Job Template attributes in the create and
699 Validate-Job operations when the client supplies the "ipp-attribute-fidelity" Operation attribute with the
700 'false' value. This last rule is so that we can add OPTIONAL Operation attributes to future versions of
701 IPP so that older clients can inter-work with new IPP objects and newer clients can inter-work with
702 older IPP objects. (If the new attribute cannot be ignored without performing unexpectedly, the major
703 version number would have been increased in the protocol document and in the request). This rule for
704 Operation attributes is independent of the value of the "ipp-attribute-fidelity" attribute. For example, if
705 an IPP object doesn't support the OPTIONAL "job-k-octets" attribute', the IPP object treats "job-k-
706 octets" as an unknown attribute and only checks the length for the 'integer' attribute syntax supplied by
707 the client. If it is not four octets, the IPP object REJECTS the request and RETURNS the 'client-error-

708 bad-request' status code, else the IPP object copies the attribute to the Unsupported Attribute response
709 group, setting the value to the "out-of-band" 'unsupported' value, but otherwise ignores the attribute.

710 2.2.2 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add 711 Documents

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

715 2.2.2.1 Default "ipp-attribute-fidelity" if not supplied

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

718 2.2.2.2 Check that the Printer object is accepting jobs

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

721 2.2.2.3 Validate the values of the Job Template attributes

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

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

727 b) that the attribute syntax tag is correct for that attribute according to [IPP-MOD] Sections 4.2 to
728 4.4.

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

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

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

- 740 1. reject the operation and return the 'client-error-bad syntax' error status code
- 741 2. accept the operation and use the first occurrence of the attribute
- 742 3. accept the operation and use the last occurrence of the attribute

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

745 2.2.3 Algorithm for job validation

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

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

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

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

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

758 Example" If V is "sides-supported" with values: 'one-sided', 'two-sided-long', and 'two-sided-
759 short', then Z takes on the successive values: 'one-sided', 'two-sided-long', and 'two-sided-short'.
760 If the client supplies "sides" with 'two-sided-long', the first comparison fails ('one-sided' is not
761 equal to 'two-sided-long'), the second comparison succeeds ('two-sided-long' is equal to 'two-
762 sided-long'), and the third comparison ('two-sided-short' with 'two-sided-long') is not even
763 performed.

- 764 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.

765 **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.

766

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

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

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

784 -----

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

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

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

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

792 Map the value to the nearest supported value in the range 1:100 as specified by the number of discrete
793 values indicated by the value of the Printer's "job-priority-supported" attribute. See the formula in
794 [IPP-MOD] Section 4.2.1.
795

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

797 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-request-bad-request'.
798 IF the value -with a length less than or equal to- is greater than 255 octets, REJECT/RETURN 'client-
799 error-request-value-too-long'.

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

802 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the
803 unsupported value to the Unsupported Attributes response group.
804

805 job-sheets (type3 keyword | name)

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

807 IF the value length is greater than~~with a length less than or equal to~~ 255 octets, REJECT/RETURN
808 'client-error-request-value-too-long'.
809 IF NOT in the Printer object's "job-sheets-supported" attribute, copy the attribute and the unsupported
810 value to the Unsupported Attributes response group.
811

812 multiple-document-handling (type2 keyword)

813 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-request-bad-request'.
814 IF the value length is greater than~~with a length less than or equal to~~ 255 octets, REJECT/RETURN
815 'client-error-request-value-too-long'.
816 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute and
817 the unsupported value to the Unsupported Attributes response group.
818

819 copies (integer(1:MAX))

820 IF NOT a single 'integer' value with a length equal to 4 octets,
821 REJECT/RETURN 'client-error-bad-request'.
822 IF NOT in range of the Printer object's "copies-supported" attribute
823 copy the attribute and the unsupported value to the Unsupported Attributes response group.
824

825 finishings (1setOf type2 enum)

826 IF NOT an 'enum' value(s) each with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
827 request'.
828 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the unsupported
829 value(s), but not any supported values, to the Unsupported Attributes response group.
830

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

832 IF NOT a 'rangeOfInteger' value(s) each with a length equal to 8 octets, REJECT/RETURN 'client-
833 error-bad-request'.
834 IF first value is greater than second value in any range, the ranges are not in ascending order, or ranges
835 overlap, REJECT/RETURN 'client-error-bad-request'.
836 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to the
837 Unsupported Attributes response group and set the value to the "out-of-band" 'unsupported' value.
838

839 sides (type2 keyword)

840 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-request-bad-request'.
841 IF the value length is greater than~~with a length less than or equal to~~ 255 octets, REJECT/RETURN
842 'client-error-request-value-too-long'.
843 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported value
844 to the Unsupported Attributes response group.
845

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

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

848 REJECT/RETURN 'client-error-bad-request'.
849 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"
850 attribute, copy the attribute and value to the Unsupported Attribute response group.
851

852 orientation-requested (type2 enum)

853 IF NOT a single 'enum' value with a length equal to 4 octets,
854 REJECT/RETURN 'client-error-bad-request'.
855 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the
856 unsupported value to the Unsupported Attributes response group.
857

858 media (type3 keyword | name)

859 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-request-bad-request'.
860 IF the with a value length less than or equal to is greater than 255 octets, REJECT/RETURN 'client-
861 error-request-value-too-long'.
862 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported value
863 to the Unsupported Attributes response group.
864

865 printer-resolution (resolution)

866 IF NOT a single 'resolution' value with a length equal to 9 octets,
867 REJECT/RETURN 'client-error-bad-request'.
868 IF NOT in the Printer object's "printer-resolution-supported~~multiple-document-handling-supported~~"
869 attribute, copy the attribute and the unsupported value to the Unsupported Attributes response
870 group.
871

872 print-quality (type2 enum)

873 IF NOT a single 'enum' value with a length equal to 4 octets,
874 REJECT/RETURN 'client-error-bad-request'.
875 IF NOT in the Printer object's "print-quality-supported" attribute, copy the attribute and the
876 unsupported value to the Unsupported Attributes response group.
877

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

879 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute
880 syntax,
881 REJECT/RETURN 'client-error-bad-request' if the length of the attribute syntax is fixed or 'client-error-
882 request-value-too-long' if the length of the attribute syntax is variable.
883 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
884 attribute value to the "out-of-band" 'unsupported' value. Any remaining Job Template Attributes are
885 either unknown or unsupported Job Template attributes and are validated algorithmically according
886 to their attribute syntax for proper length (see below).
887 -----

888

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

895 Name	Octet length check for read-write attributes
896 -----	-----
897 'textWithLanguage	<= 1023 AND 'naturalLanguage' <= 63
898 'textWithoutLanguage'	<= 1023
899 'nameWithLanguage'	<= 255 AND 'naturalLanguage' <= 63
900 'nameWithoutLanguage'	<= 255
901 'keyword'	<= 255
902 'enum'	= 4
903 'uri'	<= 1023
904 'uriScheme'	<= 63
905 'charset'	<= 63
906 'naturalLanguage'	<= 63
907 'mimeType'	<= 255
908 'octetString'	<= 1023
909 'boolean'	= 1
910 'integer'	= 4
911 'rangeOfInteger'	= 8
912 'dateTime'	= 11
913 'resolution'	= 9
914 'lsetOf X'	
915	

916 2.2.3.1 Check for conflicting Job Template attributes values

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

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

930 2.2.3.2 Decide whether to REJECT the request

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

- 934 (1) 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes
935 supplied by the client.
936 (2) 'client-error-attributes-or-values-not-supported' status code, otherwise.
937

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

942 2.2.3.3 For the Validate-Job operation, RETURN one of the success status codes

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

- 944 (1) the "successful-ok" status code, if there are no unsupported or conflicting Job Template attributes or
945 values.
946 (2) the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or
947 values.
948 (3) the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template
949 attributes or values.

950

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

955 2.2.3.4 Create the Job object with attributes to support

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

- 957 (1) creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and
958 initializes all of the job's other supported Job Description attributes.
959 (2) removes all unsupported attributes from the Job object.
960 (3) for each unsupported value, removes either the unsupported value or substitutes the unsupported
961 attribute value with some supported value. If an attribute has no values after removing unsupported
962 values from it, the attribute is removed from the Job object (so that the normal default behavior at
963 job processing time will take place for that attribute).
964 (4) for each conflicting value, removes either the conflicting value or substitutes the conflicting
965 attribute value with some other supported value. If an attribute has no values after removing
966 conflicting values from it, the attribute is removed from the Job object (so that the normal default
967 behavior at job processing time will take place for that attribute).

968

969 If there were no attributes or values flagged as unsupported, or the value of "ipp-attribute-fidelity" was
970 'false', the Printer object is able to accept the create request and create a new Job object. If the "ipp-
971 attribute-fidelity" attribute is set to 'true', the Job Template attributes that populate the new Job object are
972 necessarily all the Job Template attributes supplied in the create request. If the "ipp-attribute-fidelity"
973 attribute is set to 'false', the Job Template attributes that populate the new Job object are all the client
974 supplied Job Template attributes that are supported or that have value substitution. Thus, some of the
975 requested Job Template attributes may not appear in the Job object because the Printer object did not
976 support those attributes. The attributes that populate the Job object are persistently stored with the Job
977 object for that Job. A Get-Job-Attributes operation on that Job object will return only those attributes that
978 are persistently stored with the Job object.

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

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

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

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

1000 2.2.3.5 Return one of the success status codes

1001 Once the Job object has been created, the Printer object accepts the request and returns to the client:

- 1002 (1) the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes or
1003 values.
- 1004 (2) the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template
1005 attribute or values.
- 1006 (3) the 'successful-ok-ignored-or-substituted-attributes' status code, if there are only unsupported Job
1007 Template attributes or values.

1008

1009 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this
1010 step. If the unsupported Operation attribute was a serious error, the above already rejected the request in a
1011 previous step. If control gets to this step with unsupported Operation attributes being returned, they are not
1012 serious errors.

1013 The Printer object also returns Job status attributes that indicate the initial state of the Job ('pending',
1014 'pending-held', 'processing', etc.), etc. See Print-Job Response, [IPP-MOD] section 3.2.1.2.

1015 2.2.3.6 Accept appended Document Content

1016 The Printer object accepts the appended Document Content data and either starts it printing, or spools it for
1017 later processing.

1018 2.2.3.7 Scheduling and Starting to Process the Job

1019 The Printer object uses its own configuration and implementation specific algorithms for scheduling the
1020 Job in the correct processing order. Once the Printer object begins processing the Job, the Printer changes
1021 the Job's state to 'processing'. If the Printer object supports PDL override (the "pdl-override-supported"
1022 attribute set to 'attempted'), the implementation does its best to see that IPP attributes take precedence over
1023 embedded instructions in the document data.

1024 2.2.3.8 Completing the Job

1025 The Printer object continues to process the Job until it can move the Job into the 'completed' state. If an
1026 Cancel-Job operation is received, the implementation eventually moves the Job into the 'canceled' state. If
1027 the system encounters errors during processing that do not allow it to progress the Job into a completed
1028 state, the implementation halts all processing, cleans up any resources, and moves the Job into the 'aborted'
1029 state.

1030 2.2.3.9 Destroying the Job after completion

1031 Once the Job moves to the 'completed', 'aborted', or 'canceled' state, it is an implementation decision as to
1032 when to destroy the Job object and release all associated resources. Once the Job has been destroyed, the
1033 Printer would return either the "client-error-not-found" or "client-error-gone" status codes for operations
1034 directed at that Job.

1035 Note: the Printer object SHOULD NOT re-use a "job-uri" or "job-id" value for a sufficiently long time
1036 after a job has been destroyed, so that stale references kept by clients are less likely to access the wrong
1037 (newer) job.

1038 2.2.3.10 Interaction with "ipp-attribute-fidelity"

1039 Some Printer object implementations may support "ipp-attribute-fidelity" set to 'true' and "pdl-override-
1040 supported" set to 'attempted' and yet still not be able to realize exactly what the client specifies in the create

1041 request. This is due to legacy decisions and assumptions that have been made about the role of job
1042 instructions embedded within the document data and external job instructions that accompany the
1043 document data and how to handle conflicts between such instructions. The inability to be 100% precise
1044 about how a given implementation will behave is also compounded by the fact that the two special
1045 attributes, "ipp-attribute-fidelity" and "pdl-override-supported", apply to the whole job rather than specific
1046 values for each attribute. For example, some implementations may be able to override almost all Job
1047 Template attributes except for "number-up".

1048 2.3 Status codes returned by operation (Issue 1.50)

1049 This section lists all status codes once in the first operation (Print-Job). Then it lists the status codes that
1050 are different or specialized for subsequent operations under each operation.

1051 2.3.1 Printer Operations

1052 2.3.1.1 Print-Job

1053 The Printer object MUST return one of the following "status-code" values for the indicated reason.
1054 Whether all of the document data has been accepted or not before returning the success or error response
1055 depends on implementation. See Section 14 for a more complete description of each status code.

1056 For the following success status codes, the Job object has been created and the "job-id", and "job-uri"
1057 assigned and returned in the response:

1058 successful-ok: no request attributes were substituted or ignored.

1059 successful-ok-ignored-or-substituted-attributes: some supplied (1) attributes were ignored or (2)
1060 unsupported attribute syntaxes or values were substituted with supported values or were ignored.
1061 Unsupported attributes, attribute syntaxes, or values MUST be returned in the Unsupported
1062 Attributes group of the response.

1063 successful-ok-conflicting-attributes: some supplied attribute values conflicted with the values of other
1064 supplied attributes and were either substituted or ignored. Attributes or values which conflict with
1065 other attributes and have been substituted or ignored MUST be returned in the Unsupported
1066 Attributes group of the response as supplied by the client.
1067

1068 [ipp-mod] section 3.1.6 Operation Status Codes and Messages states (Issue 1.19):

1069 If the Printer object supports the "status-message" operation attribute, it SHOULD use the
1070 REQUIRED 'utf-8' charset to return a status message for the following error status codes (see
1071 section 14): 'client-error-bad-request', 'client-error-charset-not-supported', 'server-error-internal-
1072 error', 'server-error-operation-not-supported', and 'server-error-version-not-supported'. In this case,
1073 it MUST set the value of the "attributes-charset" operation attribute to 'utf-8' in the error response.

1074 For the following error status codes, no job is created and no "job-id" or "job-uri" is returned:

1075 client-error-bad-request: The request syntax does not conform to the specification.

1076 client-error-forbidden: The request is being refused for authorization or authentication reasons. The
1077 implementation security policy is to not reveal whether the failure is one of authentication or
1078 authorization.

1079 client-error-not-authenticated: Either the request requires authentication information to be supplied or
1080 the authentication information is not sufficient for authorization.

1081 client-error-not-authorized: The requester is not authorized to perform the request on the target object.

1082 client-error-not-possible: The request cannot be carried out because of the state of the system. See also
1083 'server-error-not-accepting-jobs' status code which MUST take precedence if the Printer object's
1084 "printer-accepting-jobs" attribute is 'false'.

1085 client-error-timeout: not applicable.

1086 client-error-not-found: the target object does not exist.

1087 client-error-gone: the target object no longer exists and no forwarding address is known.

1088 client-error-request-entity-too-large: the size of the request and/or print data exceeds the capacity of the
1089 IPP Printer to process it.

1090 client-error-request-value-too-long: the size of request variable length attribute values, such as 'text'
1091 and 'name' attribute syntaxes, exceed the maximum length specified in [IPP-MOD] for the attribute
1092 and MUST be returned in the Unsupported Attributes Group.

1093 client-error-document-format-not-supported: the document format supplied is not supported. The
1094 "document-format" attribute with the unsupported value MUST be returned in the Unsupported
1095 Attributes Group. This error SHOULD take precedence over any other 'xxx-not-supported' error,
1096 except 'client-error-charset-not-supported'.

1097 client-error-attributes-or-values-not-supported: one or more supplied attributes, attribute syntaxes, or
1098 values are not supported and the client supplied the "ipp-attributes-fidelity" operation attribute with
1099 a 'true' value. They MUST be returned in the Unsupported Attributes Group as explained below.

1100 client-error-uri-scheme-not-supported: not applicable.

1101 client-error-charset-not-supported: the charset supplied in the "attributes-charset" operation attribute is
1102 not supported. The Printer's "configured-charset" MUST be returned in the response as the value of
1103 the "attributes-charset" operation attribute and used for any 'text' and 'name' attributes returned in
1104 the error response. This error SHOULD take precedence over any other error, unless the request
1105 syntax is so bad that the client's supplied "attributes-charset" cannot be determined.

1106 client-error-conflicting-attributes: one or more supplied attribute values conflicted with each other and
1107 the client supplied the "ipp-attributes-fidelity" operation attribute with a 'true' value. They MUST
1108 be returned in the Unsupported Attributes Group as explained below.

1109 server-error-internal-error: an unexpected condition prevents the request from being fulfilled.

1110 server-error-operation-not-supported: not applicable (since Print-Job is REQUIRED).

1111 server-error-service-unavailable: the service is temporarily overloaded.

1112 server-error-version-not-supported: the version in the request is not supported. The "closest" version
1113 number supported MUST be returned in the response.

1114 server-error-device-error: a device error occurred while receiving or spooling the request or document
1115 data or the IPP Printer object can only accept one job at a time.

1116 server-error-temporary-error: a temporary error such as a buffer full write error, a memory overflow, or
1117 a disk full condition occurred while receiving the request and/or the document data.

1118 server-error-not-accepting-jobs: the Printer object's "printer-is-not-accepting-jobs" attribute is 'false'.

1119 server-error-busy: the Printer is too busy processing jobs to accept another job at this time.

1120 server-error-job-canceled: the job has been canceled by an operator or the system while the client was
1121 transmitting the document data.

1122 2.3.1.2 Print-URI

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

1126 server-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation
1127 attribute is not supported and is returned in the Unsupported Attributes group.

1128 2.3.1.3 Validate-Job

1129 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Validate-
1130 Job. See Section 14 for a more complete description of each status code.

1131 2.3.1.4 Create-Job

1132 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Create-
1133 Job with the following specializations and differences. See Section 14 for a more complete description of
1134 each status code.

1135 server-error-operation-not-supported: the Create-Job operation is not supported.

1136 2.3.1.5 Get-Printer-Attributes

1137 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-
1138 Printer-Attributes operation with the following specializations and differences. See Section 14 for a more
1139 complete description of each status code.

1140 For the following success status codes, the requested attributes are returned in Group 3 in the response:

1141 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1142 attributes were unsupported.

1143 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1144 operation attribute MAY, but NEED NOT, be returned with the unsupported values.

1145 successful-ok-conflicting-attributes: same as Print-Job.

1146 For the error status codes, Group 3 is returned containing no attributes or is not returned at all:

1147 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.

1148 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.

1149 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1150 MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.

1151 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.

1152 server-error-operation-not-supported: not applicable (since Get-Printer-Attributes is REQUIRED).

1153 server-error-device-error: same as Print-Job, except that no document data is involved.

1154 server-error-temporary-error: same as Print-Job, except that no document data is involved.

1155 server-error-not-accepting-jobs: not applicable..

1156 server-error-busy: same as Print-Job, except the IPP object is too busy to accept even query requests.

1157 server-error-job-canceled: not applicable..

1158 2.3.1.6 Get-Jobs

1159 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-
1160 Jobs operation with the following specializations and differences. See Section 14 for a more complete
1161 description of each status code.

1162 For the following success status codes, the requested attributes are returned in Group 3 in the response:

1163 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1164 attributes were unsupported.

1165 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1166 operation attribute MAY, but NEED NOT, be returned with the unsupported values.

1167 successful-ok-conflicting-attributes: same as Print-Job.

1168 For any error status codes, Group 3 is returned containing no attributes or is not returned at all. The
1169 following brief error status code descriptions contain unique information for use with Get-Jobs operation.
1170 See section 14 for the other error status codes that apply uniformly to all operations:

1171 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.

1172 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.

1173 client-error-document-format-not-supported: not applicable.

1174 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1175 MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.

1176 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.

1177 server-error-operation-not-supported: not applicable (since Get-Jobs is REQUIRED).

1178 server-error-device-error: same as Print-Job, except that no document data is involved.

1179 server-error-temporary-error: same as Print-Job, except that no document data is involved.

1180 server-error-not-accepting-jobs: not applicable.

1181 server-error-job-canceled: not applicable.

1182 2.3.2 Job Operations

1183 2.3.2.1 Send-Document

1184 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-
1185 Printer-Attributes operation with the following specializations and differences. See Section 14 for a more
1186 complete description of each status code.

1187 For the following success status codes, the document has been added to the specified Job object and the
1188 job's "number-of-documents" attribute has been incremented:

1189 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

1190 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

1191 successful-ok-conflicting-attributes: same as Print-Job.

1192 For the error status codes, no document has been added to the Job object and the job's "number-of-
1193 documents" attribute has not been incremented:

1194 client-error-not-possible: Same as Print-Job, except that the Printer's "printer-is-accepting-jobs"
1195 attribute is not involved, so that the client is able to finish submitting a multi-document job after this

1196 attribute has been set to 'true'. Another condition is that the state of the job precludes Send-
1197 Document, i.e., the job has already been closed out by the client. However, if the IPP Printer closed
1198 out the job due to timeout, the 'client-error-timeout' error status SHOULD be returned instead.
1199 client-error-timeout: This request was sent after the Printer closed the job, because it has not received a
1200 Send-Document or Send-URI operation within the Printer's "multiple-operation-time-out" period .
1201 client-error-request-entity-too-large: same as Print-Job.
1202 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attributes-fidelity" operation
1203 attribute is not involved..
1204 server-error-operation-not-supported: the Send-Document request is not supported.
1205 server-error-not-accepting-jobs: not applicable.
1206 server-error-job-canceled: the job has been canceled by an operator or the system while the client was
1207 transmitting the data.

1208 2.3.2.2 Send-URI

1209 All of the Print-Job status code descriptions in Section 3.2.1.2 Print-Job Response with the specializations
1210 described for Send-Document are applicable to Send-URI. See Section 14 for a more complete description
1211 of each status code.

1212 server-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation
1213 attribute is not supported and the "document-uri" attribute MUST be returned in the Unsupported
1214 Attributes group.

1215 2.3.2.3 Cancel-Job

1216 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Cancel-
1217 Job with the following specializations and differences. See Section 14 for a more complete description of
1218 each status code.

1219 For the following success status codes, the Job object is being canceled or has been canceled:

1220 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
1221 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
1222 successful-ok-conflicting-attributes: same as Print-Job.
1223

1224 For any of the error status codes, the Job object has not been canceled or was previously canceled.

1225 client-error-not-possible: The request cannot be carried out because of the state of the Job object
1226 ('completed', 'canceled', or 'aborted') or the state of the system.
1227 client-error-not-found: the target Printer and/or Job object does not exist.
1228 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding address is
1229 known.
1230 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
1231 client-error-document-format-not-supported: not applicable.
1232 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1233 and values MUST be ignored.
1234 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-
1235 jobs" attribute is not involved.

- 1236 server-error-operation-not-supported: not applicable (Cancel-Job is REQUIRED).
- 1237 server-error-device-error: same as Print-Job, except no document data is involved.
- 1238 server-error-temporary-error: same as Print-Job, except no document data is involved.
- 1239 server-error-not-accepting-jobs: not applicable..
- 1240 server-error-job-canceled: not applicable.

1241 2.3.2.4 Get-Job-Attributes

1242 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Get-Job-
1243 Attributes with the following specializations and differences. See Section 14 for a more complete
1244 description of each status code.

1245 For the following success status codes, the requested attributes are returned in Group 3 in the response:

- 1246 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1247 attributes were unsupported.
- 1248 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1249 operation attribute MAY, but NEED NOT, be returned with the unsupported values.
- 1250 successful-ok-conflicting-attributes: same as Print-Job.

1251 For the error status codes, Group 3 is returned containing no attributes or is not returned at all.

- 1252 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.
- 1253 client-error-document-format-not-supported: not applicable.
- 1254 client-error-attributes-or-values-not-supported: not applicable.
- 1255 client-error-uri-scheme-not-supported: not applicable.
- 1256 client-error-conflicting-attributes: not applicable
- 1257 server-error-operation-not-supported: not applicable (since Get-Job-Attributes is REQUIRED).
- 1258 server-error-device-error: same as Print-Job, except no document data is involved.
- 1259 server-error-temporary-error: sane as Print-Job, except no document data is involved..
- 1260 server-error-not-accepting-jobs: not applicable.
- 1261 server-error-job-canceled: not applicable.

1262 2.4 Validate-Job

1263 The Validate-Job operation has been designed so that its implementation may be a part of the Print-Job
1264 operation. Therefore, requiring Validate-Job is not a burden on implementers. Also it is useful for client's
1265 to be able to count on its presence in all conformance implementations, so that the client can determine
1266 before sending a long document, whether the job will be accepted by the IPP Printer or not.

1267 2.5 Case Sensitivity in URIs (issue 1.6)

1268 IPP client and server implementations must be aware of the diverse uppercase/lowercase nature of URIs.
1269 RFC 2396 defines URL schemes and Host names as case insensitive but reminds us that the rest of the
1270 URL may well demonstrate case sensitivity. When creating URL's for fields where the choice is
1271 completely arbitrary, it is probably best to select lower case. However, this cannot be guaranteed and

1272 implementations MUST NOT rely on any fields being case-sensitive or case-insensitive in the URL beyond
1273 the URL scheme and host name fields.

1274 The reason that the IPP specification does not make any restrictions on URIs, is so that implementations of
1275 IPP may use off-the-shelf components that conform to the standards that define URIs, such as RFC 2396
1276 and the HTTP/1.1 specifications [RFC2068]. See these specifications for rules of matching, comparison,
1277 and case-sensitivity.

1278 It is also recommended that ~~that~~ System Administrators and implementations avoid creating URLs for
1279 different printers that differ only in their case. For example, don't have Printer1 and printer1 as two
1280 different IPP Printers.

1281 The HTTP/1.1 specification [RFC2068] contains more details on comparing URLs.

1282 2.6 Character Sets, natural languages, and internationalization

1283 This section discusses character set support, natural language support and internationalization.

1284 2.6.1 Character set code conversion support (Issue 1.5)

1285 IPP clients and IPP objects are REQUIRED to support UTF-8. They MAY support additional charsets. It
1286 is RECOMMENDED that an IPP object also support US-ASCII, since many clients support US-ASCII,
1287 and indicate that UTF-8 and US-ASCII are supported by populating the Printer's "charset-supported" with
1288 'utf-8' and 'us-ascii' values. An IPP object is required to code convert with as little loss as possible between
1289 the charsets that it supports, as indicated in the Printer's "charsets-supported" attribute.

1290 How should the server handle the situation where the "attributes-charset" of the response itself is "us-ascii",
1291 but one or more attributes in that response is in the "utf-8" format?

1292 Example: Consider a case where a client sends a Print-Job request with "utf-8" as the value of "attributes-
1293 charset" and with the "job-name" attribute supplied. Later another client submits a Get-Job-Attribute or
1294 Get-Jobs request. This second request contains the "attributes-charset" with value "us-ascii" and
1295 "requested-attributes" attribute with exactly one value "job-name".

1296 According to the IPP-Mod document (section 3.1.4.2), the value of the "attributes-charset" for the response
1297 of the second request must be "us-ascii" since that is the charset specified in the request. The "job-name"
1298 value, however, is in "utf-8" format. Should the request be rejected even though both "utf-8" and "us-ascii"
1299 charsets are supported by the server? or should the "job-name" value be converted to "us-ascii" and return
1300 "successful-ok-conflicting-attributes" (0x0002) as the status code?

1301 Answer: An IPP object that supports both utf-8 (REQUIRED) and us-ascii, the second paragraph of
1302 section 3.1.4.2 applies so that the IPP object MUST accept the request, perform code set conversion
1303 between these two charsets with "the highest fidelity possible" and return 'successful-ok', rather than a
1304 warning 'successful-ok-conflicting-attributes, or an error. The printer will do the best it can to convert
1305 between each of the character sets that it supports--even if that means providing a string of question marks
1306 because none of the characters are representable in US ASCII. If it can't perform such conversion, it

1307 MUST NOT advertise us-ascii as a value of its "attributes-charset-supported" and MUST reject any request
1308 that requests 'us-ascii'.

1309 One IPP object implementation strategy is to convert all request text and name values to a Unicode internal
1310 representation. This is 16-bit and virtually universal. Then convert to the specified operation attributes-
1311 charset on output.

1312 Also it would be smarter for a client to ask for 'utf-8', rather than 'us-ascii' and throw away characters that it
1313 doesn't understand, rather than depending on the code conversion of the IPP object.

1314 2.6.2 What charset to return when an unsupported charset is requested (Issue 1.19)?

1315 Section 3.1.4.1 Request Operation attributes was clarified in November 1998 as follows:

1316 All clients and IPP objects MUST support the 'utf-8' charset [RFC2044] and MAY support
1317 additional charsets provided that they are registered with IANA [IANA-CS]. If the Printer object
1318 does not support the client supplied charset value, the Printer object MUST reject the request, set
1319 the "attributes-charset" to 'utf-8' in the response, and return the 'client-error-charset-not-supported'
1320 status code and any 'text' or 'name' attributes using the 'utf-8' charset.

1321 Since the client and IPP object MUST support UTF-8, returning any text or name attributes in UTF-8 when
1322 the client requests a charset that is not supported should allow the client to display the text or name.

1323 Since such an error is a client error, rather than a user error, the client should check the status code first so
1324 that it can avoid displaying any other returned 'text' and 'name' attributes that are not in the charset
1325 requested.

1326 Furthermore, [ipp-mod] section 14.1.4.14 client-error-charset-not-supported (0x040D) was clarified in
1327 November 1998 as follows:

1328 For any operation, if the IPP Printer does not support the charset supplied by the client in the
1329 "attributes-charset" operation attribute, the Printer MUST reject the operation and return this status
1330 and any 'text' or 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1).

1331 2.6.3 Natural Language Override (NLO) (Issue 1.45)

1332 The 'text' and 'name' attributes each have two forms. One has an implicit natural language, and the other
1333 has an explicit natural language. The 'textWithoutLanguage' and 'textWithLanguage' are the two 'text'
1334 forms. The 'nameWithoutLanguage' and 'nameWithLanguage' are the two 'name' forms. If a receiver (IPP
1335 object or IPP client) supports an attribute with attribute syntax 'text', it MUST support both forms in a
1336 request and a response. A sender (IPP client or IPP object) MAY send either form for any such attribute.
1337 When a sender sends a WithoutLanguage form, the implicit natural language is specified in the "attributes-
1338 natural-language" operation attribute which all senders MUST include in every request and response.

1339 When a sender sends a WithLanguage form, it MAY be different from the implicit natural language
1340 supplied by the sender or it MAY be the same. The receiver MUST treat either form equivalently.

1341 There is an implementation decision for senders, whether to always send the WithLanguage forms or use
1342 the WithoutLanguage form when the attribute's natural language is the same as the request or response.
1343 The former approach makes the sender implementation simpler. The latter approach is more efficient on
1344 the wire and allows inter-working with non-conforming receivers that fail to support the WithLanguage
1345 forms. As each approach have advantages, the choice is completely up to the implementer of the sender.

1346 Furthermore, when a client receives a 'text' or 'name' job attribute that it had previously supplied, that client
1347 MUST NOT expect to see the attribute in the same form, i.e., in the same WithoutLanguage or
1348 WithLanguage form as the client supplied when it created the job. The IPP object is free to transform the
1349 attribute from the WithLanguage form to the WithoutLanguage form and vice versa, as long as the natural
1350 language is preserved. However, in order to meet this latter requirement, it is usually simpler for the IPP
1351 object implementation to store the natural language explicitly with the attribute value, i.e., to store using an
1352 internal representation that resembles the WithLanguage form.

1353 The IPP Printer MUST copy the natural language of a job, i.e., the value of the "attributes-natural-
1354 language" operation attribute supplied by the client in the create operation, to the Job object as a Job
1355 Description attribute, so that a client is able to query it. In returning a Get-Job-Attributes response, the IPP
1356 object MAY return one of three natural language values in the response's "attributes-natural-language"
1357 operation attribute: (1) that requested by the requester, (2) the natural language of the job, or (3) the
1358 configured natural language of the IPP Printer, if the requested language is not supported by the IPP
1359 Printer.

1360 This "attributes-natural-language" Job Description attribute is useful for an IPP object implementation that
1361 prints start sheets in the language of the user who submitted the job. This same Job Description attribute is
1362 useful to a multi-lingual operator who has to communicate with different job submitters in different natural
1363 languages. This same Job Description attribute is expected to be used in the future to generate notification
1364 messages in the natural language of the job submitter.

1365 Early drafts of [IPP-MOD] contained a job-level natural language override (NLO) for the Get-Jobs
1366 response. A job-level (NLO) is an (unrequested) Job Attribute which then specified the implicit natural
1367 language for any other WithoutLanguage job attributes returned in the response for that job.
1368 Interoperability testing of early implementations showed that no one was implementing the job-level NLO
1369 in Get-Job responses. So the job-level NLO was eliminated from the Get-Jobs response. This
1370 simplification makes all requests and responses consistent in that the implicit natural language for any
1371 WithoutLanguage 'text' or 'name' form is always supplied in the request's or response's "attributes-natural-
1372 language" operation attribute.

1373 2.7 The "queued-job-count" Printer Description attribute

1374 2.7.1 Why is "queued-job-count" RECOMMENDED (Issue 1.14)?

1375 The reason that "queued-job-count" is RECOMMENDED, is that some clients look at that attribute alone
1376 when summarizing the status of a list of printers, instead of doing a Get-Jobs to determine the number of
1377 jobs in the queue. Implementations that fail to support the "queued-job-count" will cause that client to
1378 display 0 jobs when there are actually queued jobs.

1379 We would have made it a REQUIRED Printer attribute, but some implementations had already been
1380 completed before the issue was raised, so making it a SHOULD was a compromise.

1381 2.7.2 Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?

1382 The "queued-job-count" is not a good measure of how busy the printer is when there are held jobs. A
1383 future registration could be to add a "held-job-count" (or an "active-job-count") Printer Description
1384 attribute if experience shows that such an attribute (combination) is needed to quickly indicate how busy a
1385 printer really is.

1386 2.8 Sending empty attribute groups (Issue 1.16)

1387 The [IPP-MOD] and [IPP-PRO] specifications RECOMMEND that a sender not send an empty attribute
1388 group in a request or a response. However, they REQUIRE a receiver to accept an empty attribute group as
1389 equivalent to the omission of that group. So a client SHOULD omit the Job Template Attributes group
1390 entirely in a create operation that is not supplying any Job Template attributes. Similarly, an IPP object
1391 SHOULD omit an empty Unsupported Attributes group if there are no unsupported attributes to be returned
1392 in a response.

1393 The [IPP-PRO] specification REQUIRES a receiver to be able to receive either an empty attribute group or
1394 an omitted attribute group and treat them equivalently. The term "receiver" means an IPP object for a
1395 request and a client for a response. The term "sender" means a client for a request and an IPP object for a
1396 response.

1397 There is an exception to the rule for Get-Jobs when there are no attributes to be returned. [ipp-pro]
1398 contains the following paragraph:

1399 The syntax allows an xxx-attributes-tag to be present when the xxx-attribute-sequence that follows
1400 is empty. The syntax is defined this way to allow for the response of Get-Jobs where no attributes
1401 are returned for some job-objects. Although it is RECOMMENDED that the sender not send an
1402 xxx-attributes-tag if there are no attributes (except in the Get-Jobs response just mentioned), the
1403 receiver MUST be able to decode such syntax.

1404 2.9 Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)

1405 In the Get-Printer-Attributes, Get-Jobs, or Get-Job-Attributes responses, the client cannot depend on getting
1406 unsupported attributes returned in the Unsupported Attributes group that the client requested, but are not
1407 supported by the IPP object. However, such unsupported requested attributes will not be returned in the
1408 Job Attributes or Printer Attributes group (since they are unsupported). Furthermore, the IPP object is
1409 REQUIRED to return the 'successful-ok-ignored-or-substituted-attributes' status code, so that the client
1410 knows that not all that was requested has been returned.

1411 2.10 Returning job-state in Print-Job response (Issue 1.30)

1412 An IPP client submits a small job via Print-Job. By the time the IPP printer/print server is putting together
1413 a response to the operation, the job has finished printing and been removed as an object from the print
1414 system. What should the job-state be in the response?

1415 The Model suggests that the Printer return a response before it even accepts the document content. The Job
1416 Object Attributes are returned only if the IPP object returns one of the success status codes. Then the job-
1417 state would always be "pending" or "pending-held".

1418 This issue comes up for the implementation of an IPP Printer object as a server that forwards jobs to
1419 devices that do not provide job status back to the server. If the server is reasonably certain that the job
1420 completed successfully, then it should return the job-state as 'completed'. Also the server can keep the job
1421 in its "job history" long after the job is no longer in the device. Then a user could query the server and see
1422 that the job was in the 'completed' state and completed as specified by the job's "time-at-completed" time
1423 which would be the same as the server submitted the job to the device.

1424 An alternative is for the server to respond to the client before or while sending the job to the device, instead
1425 of waiting until the server has finished sending the job to the device. In this case, the server can return the
1426 job's state as 'pending' with the 'job-outgoing' value in the job's "job-state-reasons" attribute.

1427 If the server doesn't know for sure whether the job completed successfully (or at all), it could return the
1428 (out-of-band) 'unknown' value.

1429 On the other hand, if the server is able to query the device and/or setup some sort of event notification that
1430 the device initiates when the job makes state transitions, then the server can return the current job state in
1431 the Print-Job response and in subsequent queries because the server knows what the job state is in the
1432 device (or can query the device).

1433 All of these alternatives depend on implementation of the server and the device.

1434 2.11 Flow controlling the data portion of a Print-Job request (Issue 1.22)

1435 A paused printer (or one that is stopped due to paper out or jam or spool space full or buffer space full, may
1436 flow control the data of a Print-Job operation (at the TCP/IP layer), so that the client is not able to send all
1437 the document data. Consequently, the Printer will not return a response until the condition is changed.

1438 The Printer should not return a Print-Job response with an error code in any of these conditions, since either
1439 the printer will be resumed and/or the condition will be freed either by human intervention or as jobs print.

1440 In writing test scripts to test IPP Printers, the script must also be written not to expect a response, if the
1441 printer has been paused, until the printer is resumed, in order to work with all possible implementations.

1442 2.12 Multi-valued attributes (Issue 1.31)

1443 What is the attribute syntax for a multi-valued attribute? Since some attributes support values in more than
1444 one data type, such as "media", "job-hold-until", and "job-sheets", IPP semantics associate the attribute
1445 syntax with each value, not with the attribute as a whole. The protocol associates the attribute syntax tag
1446 with each value. Don't be fooled, just because the attribute syntax tag comes before the attribute keyword.
1447 All attribute values after the first have a zero length attribute keyword as the indication of a subsequent
1448 value of the same attribute.

1449 2.13 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)

1450 The following clarification was added to [ipp-mod] section 8.5:

1451 8.5 Queries on jobs submitted using non-IPP protocols

1452 If the device that an IPP Printer is representing is able to accept jobs using other job submission
1453 protocols in addition to IPP, it is RECOMMEND that such an implementation at least allow such
1454 "foreign" jobs to be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an
1455 implementation NEED NOT support all of the same IPP job attributes as for IPP jobs. The IPP
1456 object returns the 'unknown' out-of-band value for any requested attribute of a foreign job that is
1457 supported for IPP jobs, but not for foreign jobs.

1458 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such
1459 "foreign jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-
1460 Attributes and Cancel-Job. Such an implementation also needs to deal with the problem of
1461 authentication of such foreign jobs. One approach would be to treat all such foreign jobs as
1462 belonging to users other than the user of the IPP client. Another approach would be for the foreign
1463 job to belong to 'anonymous'. Only if the IPP client has been authenticated as an operator or
1464 administrator of the IPP Printer object, could the foreign jobs be queried by an IPP request.
1465 Alternatively, if the security policy is to allow users to query other users' jobs, then the foreign jobs
1466 would also be visible to an end-user IPP client using Get-Jobs and Get-Job-Attributes.

1467 Thus IPP MAY be implemented as a "universal" protocol that provides access to jobs submitted with any
1468 job submission protocol. As IPP becomes widely implemented, providing a more universal access makes
1469 sense.

1470 2.14 The 'none' value for empty sets (Issue 1.37)

1471 [ipp-mod] states that the 'none' value should be used as the value of a 1SetOf when the set is empty. In most
1472 cases, sets that are potentially empty contain keywords so the keyword 'none' is used, but for the 3
1473 finishings attributes, the values are enums and thus the empty set is represented by the enum 3. Currently
1474 there are no other attributes with 1SetOf values which can be empty and can contain values that are not
1475 keywords. This exception requires special code and is a potential place for bugs. It would have been better
1476 if we had chosen an out-of-band value, either "no-value" or some new value, such as 'none'. Since we
1477 didn't, implementations have to deal with the different representations of 'none', depending on the attribute
1478 syntax.

1479 2.15 Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?

1480 In [ipp-mod] section 3.2.6.1 'Get-Jobs Request', if the attribute 'my-jobs' is present and set to TRUE, MUST
1481 the 'requesting-user-name' attribute be there to, and if it's not present what should the IPP printer do?

1482 [ipp-mod] Section 8.3 describes the various cases of "requesting-user-name" being present or not for any
1483 operation. If the client does not supply a value for "requesting-user-name", the printer MUST assume that
1484 the client is supplying some anonymous name, such as "anonymous".

1485 2.16 The "multiple-document-handling" Job Template attribute and support of multiple document jobs

1486 ISSUE: IPP/1.0 is silent on which of the four effects an implementation would perform if it supports
1487 Create-Job, but does not support "multiple-document-handling".

1488 A fix to IPP/1.0 would be to require implementing all four values of "multiple-document-handling" if
1489 Create-Job is supported at all. Or at least 'single-document-new-sheet' and 'separate-documents-uncollated-
1490 copies'. In any case, an implementation that supports Create-Job SHOULD also support "multiple-
1491 document-handling". Support for all four values is RECOMMENDED, but at least the 'single-document-
1492 new-sheet' and 'separate-documents-uncollated-copies' values, along with the "multiple-document-
1493 handling-default" indicating the default behavior and "multiple-document-handling-supported" values. If
1494 an implementation spools the data, it should also support the 'separate-documents-collated-copies' value as
1495 well.

1496 3 Encoding and Transport

1497 This section discusses various aspects of IPP/1.0 Encoding and Transport [IPP-PRO].

1498 ~~The IPP layer doesn't have to deal with chunking. In the context of CGI scripts, the HTTP layer removes~~
1499 ~~any chunking information in the received data.~~

1500 A server is not required to send a response until after it has received the client's entire request. Hence, a
1501 client must not expect a response until after it has sent the entire request. However, we recommend that the
1502 server return a response as soon as possible if an error is detected while the client is still sending the data,
1503 rather than waiting until all of the data is received. Therefore, we also recommend that ~~A client MUST~~
1504 ~~NOT expect a response from an IPP server until after the client has sent the entire response. But~~ a client
1505 ~~MAY~~ listen for an error response that an IPP server MAY send before it receives all the data. In this case a
1506 client, if chunking the data, can send a premature zero-length chunk to end the request before sending all
1507 the data (and so the client can keep the connection open for other requests, rather than closing it). If the
1508 request is blocked for some reason, a client MAY determine the reason by opening another connection to
1509 query the server using Get-Printer-Attributes.

1510 In the following sections, there are a tables of all HTTP headers which describe their use in an IPP client or
1511 server. The following is an explanation of each column in these tables.

- 1512 • the "header" column contains the name of a header
1513 • the "request/client" column indicates whether a client sends the header.

- 1514 • the “request/ server” column indicates whether a server supports the header when received.
 1515 • the “response/ server” column indicates whether a server sends the header.
 1516 • the “response /client” column indicates whether a client supports the header when received.
 1517 • the “values and conditions” column specifies the allowed header values and the conditions for the
 1518 header to be present in a request/response.

1519 The table for “request headers” does not have columns for responses, and the table for “response headers”
 1520 does not have columns for requests.

1521 The following is an explanation of the values in the “request/client” and “response/ server” columns.

- 1522 • **must:** the client or server **MUST** send the header,
 1523 • **must-if:** the client or server **MUST** send the header when the condition described in the “values and
 1524 conditions” column is met,
 1525 • **may:** the client or server **MAY** send the header
 1526 • **not:** the client or server **SHOULD NOT** send the header. It is not relevant to an IPP
 1527 implementation.

1528 The following is an explanation of the values in the “response/client” and “request/ server” columns.

- 1529 • **must:** the client or server **MUST** support the header,
 1530 • **may:** the client or server **MAY** support the header
 1531 • **not:** the client or server **SHOULD NOT** support the header. It is not relevant to an IPP
 1532 implementation.

1533 3.1 General Headers

1534 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-	must-if	must	must-if	must	“chunked” only . Header MUST be present if Content-Length is

General-Header	Request		Response		Values and Conditions
	Client	Server	Server	Client	
Encoding					absent.
Upgrade	not	not	not	not	
Via	not	not	not	not	

1535 3.2 Request Headers

1536 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” 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

Request-Header	Client	Server	Request Values and Conditions and a "Proxy-Authenticate" header.
Range	not	not	
Referer	not	not	
User-Agent	not	not	

1537 3.3 Response Headers

1538 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	
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.

1539 3.4 Entity Headers

1540 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.

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

1541 3.5 Optional support for HTTP/1.0

1542 IPP implementations consist of an HTTP layer and an IPP layer. In the following discussion, the term
 1543 "client" refers to the HTTP client layer and the term "server" refers to the HTTP server layer. The
 1544 Encoding and Transport document [IPP-PRO] requires that HTTP 1.1 **MUST** be supported by all clients
 1545 and all servers. However, a client and/or a server implementation may choose to also support HTTP 1.0.

- 1546 • This option means that a server may choose to communicate with a (non-conforming) client that only
 1547 supports HTTP 1.0. In such cases the server should not use any HTTP 1.1 specific parameters or
 1548 features and should respond using HTTP version number 1.0.
- 1549 • This option also means that a client may choose to communicate with a (non-conforming) server that
 1550 only supports HTTP 1.0. In such cases, if the server responds with an HTTP ‘unsupported version
 1551 number’ to an HTTP 1.1 request, the client should retry using HTTP version number 1.0.

1552 3.6 HTTP/1.1 Chunking

1553 3.6.1 Disabling IPP Server Response Chunking

1554 Clients **MUST** anticipate that the HTTP/1.1 server may chunk responses and **MUST** accept them in
 1555 responses. However, a (non-conforming) HTTP client that is unable to accept chunked responses may

1556 attempt to request an HTTP 1.1 server not to use chunking in its response to an operation by using the
1557 following HTTP header:

1558 TE: identity

1559 This mechanism should not be used by a server to disable a client from chunking a request, since chunking
1560 of document data is an important feature for clients to send long documents.

1561 3.6.2 Warning About the Support of Chunked Requests

1562 This section describes some problems with the use of chunked requests and HTTP/1.1 servers.

1563 The HTTP/1.1 standard [HTTP] requires that conforming servers support chunked requests for any method.
1564 However, in spite of this requirement, some HTTP/1.1 implementations support chunked responses in the
1565 GET method, but do not support chunked POST method requests. Some HTTP/1.1 implementations that
1566 support CGI scripts [CGI] and/or servlets [Servlet] require that the client supply a Content-Length. These
1567 implementations might reject a chunked POST method and return a 411 status code (Length Required),
1568 might attempt to buffer the request and run out of room returning a 413 status code (Request Entity Too
1569 Large), or might successfully accept the chunked request.

1570 Because of this lack of conformance of HTTP servers to the HTTP/1.1 standard, the IPP standard [IPP-
1571 PRO] REQUIRES that a conforming IPP Printer object implementation support chunked requests and that
1572 conforming clients accept chunked responses. Therefore, IPP object implementers are warned to seek
1573 HTTP server implementations that support chunked POST requests in order to conform to the IPP standard
1574 and/or use implementation techniques that support chunked POST requests.

1575 **4 References**

1576 [CGI]

1577 CGI/1.1 (<http://www.ietf.org/internet-drafts/draft-coar-cgi-v11-00.txt>).

1578 [HTTP]

1579 HTTP/1.1 (<http://www.ietf.org/internet-drafts/draft-ietf-http-v11-spec-rev-06.txt>)

1580 [IPP LPD]

1581 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", draft-
1582 ietf-ipp-lpd-ipp-map-04.txt, June 1998.

1583 [IPP-MOD]

1584 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
1585 Semantics", draft-ietf-ipp-model-11.txt, November, 1998.

1586 [IPP-PRO]

1587 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and
1588 Transport", draft-ietf-ipp-protocol-076.txt, ~~June~~November, 1998.

- 1589 [IPP-RAT]
1590 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
1591 draft-ietf-ipp-rat-043.txt, JuneNovember, 1998.
- 1592 [IPP-REQ]
1593 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-032.txt,
1594 JuneNovember, 1998.
- 1595 [RFC1123]
1596 Braden, S., "Requirements for Internet Hosts - Application and Support", RFC 1123, October, 1989.
- 1597 [RFC2026]
1598 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 1599 [RFC2068]
1600 R Fielding, et al, "Hypertext Transfer Protocol – HTTP/1.1" RFC 2068, January 1997.
- 1601 [rfc2119]
1602 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March
1603 1997.
- 1604 [RFC2396]
1605 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic
1606 Syntax", RFC 2396, August 1998.
- 1607 [Servlet]
1608 Servlet Specification Version 2.1 (<http://java.sun.com/products/servlet/2.1/index.html>).
- 1609 [SSL]
1610 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.
- 1611
- 1612 4.1 Authors' Address
- 1613 Thomas N. Hastings
1614 Xerox Corporation
1615 701 Aviation Blvd.
1616 El Segundo, CA 90245
1617 hastings@cp10.es.xerox.com
1618
- 1619 Carl-Uno Manros
1620 Xerox Corporation
1621 701 Aviation Blvd.
1622 El Segundo, CA 90245
1623 manros@cp10.es.xerox.com

1624 **5 ~~Appendix C: Full Copyright Statement~~ Notices**

1625 The IETF takes no position regarding the validity or scope of any intellectual property or other rights that
1626 might be claimed to pertain to the implementation or use of the technology described in this document or
1627 the extent to which any license under such rights might or might not be available; neither does it represent
1628 that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to
1629 rights in standards-track and standards-related documentation can be found in BCP-11 [BCP-11]. Copies of
1630 claims of rights made available for publication and any assurances of licenses to be made available, or the
1631 result of an attempt made to obtain a general license or permission for the use of such proprietary rights by
1632 implementers or users of this specification can be obtained from the IETF Secretariat.

1633 The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications,
1634 or other proprietary rights which may cover technology that may be required to practice this standard.
1635 Please address the information to the IETF Executive Director.

1636 Copyright (C) The Internet Society (~~1998~~1999). All Rights Reserved

1637 This document and translations of it may be copied and furnished to others, and derivative works that
1638 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
1639 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
1640 and this paragraph are included on all such copies and derivative works. However, this document itself
1641 may not be modified in any way, such as by removing the copyright notice or references to the Internet
1642 Society or other Internet organizations, except as needed for the purpose of developing Internet standards
1643 in which case the procedures for copyrights defined in the Internet Standards process must be followed, or
1644 as required to translate it into languages other than English.

1645 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its
1646 successors or assigns.

1647 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET
1648 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,
1649 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE
1650 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
1651 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1652 **6 Change History**

1653 The change history is in reverse chronological order:

1654 6.1 Changes to produce the February 12, 1999 version from the January 8, 1999 version:

- 1655 1. Section 2.2.1.5: added check for document not found or accessible in Print-URI and Send-URI
- 1656 2. Section 3.6.2: Clarified that the IPP standard requires that servers MUST accept chunked requests
1657 and that clients MUST accept chunked responses, in spite of the lack of conformance of HTTP
1658 servers to the HTTP/1.1 requirement to support chunking.

- 1659 ~~1.1~~6.2 Changes to produce the January 8, 1999 version from the December 6, 1998 version:
- 1660 1. Added section 3.6.2: Warning About the Use of Chunked Requests with CGI Script
1661 Implementations
- 1662 2. Section 2.2.1.2: changed "printer-operations-supported" to "operations-supported".
- 1663 3. Section 2.2.1.6: changed "job-media-supported" to "job-media-sheets-supported"
- 1664 4. Section 2.2.3: separated the validation checks for variable length attributes into two separate tests:
1665 one for correct attribute syntax and one for correct length.
- 1666 5. Section 2.2.3: changed "multiple-document-handling-supported" to "printer-resolution-supported"
- 1667 6. Section 2.6.1: recommended that an IPP object also support US-ASCII charset.
- 1668 7. Section 3: Clarified that a server is not required to send a response until after it has received the
1669 client's entire request, but recommend that the server return a response as soon as possible if an
1670 error is detected while the client is still sending the data, rather than waiting until all of the data is
1671 received. Also recommended that a client listen for an error response that an IPP server MAY send
1672 before it receives all the data.
- 1673 6.3 Changes to produce the December 6, 1998 version from the November 16, 1998 version:
- 1674 Included all of the remaining agreed issues raised before the November 16, 1998 production of the Internet-
1675 Drafts for IPP/1.0 that included adding explanations to the Implementers Guide.
- 1676