

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

T. Hastings  
Xerox Corporation  
C. Manros  
Xerox Corporation  
February 12, 1999

3  
4  
5  
6  
7  
8  
9 Internet Printing Protocol/1.0: Implementer's Guide  
10 Copyright (C) The Internet Society (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

22 Abstract

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

30 The full set of IPP documents includes:

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

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

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

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

35 Mapping between LPD and IPP Protocols [IPP LPD]

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

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

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

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

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

53

## TABLE OF CONTENTS

54	1	Introduction .....	5
55	1.1	Conformance language .....	5
56	1.2	Other terminology .....	5
57	2	Model and Semantics .....	5
58	2.1	Summary of Operation Attributes.....	5
59	2.2	Suggested Operation Processing Steps for IPP Objects (Issue 1.21).....	9
60	2.2.1	Suggested Operation Processing Steps for all Operations .....	9
61	2.2.1.1	Validate version number .....	9
62	2.2.1.2	Validate operation identifier .....	10
63	2.2.1.3	Validate the request identifier .....	10
64	2.2.1.4	Validate attribute group and attribute presence and order .....	10
65	2.2.1.5	Validate the values of the REQUIRED Operation attributes.....	17
66	2.2.1.6	Validate the values of the OPTIONAL Operation attributes .....	20
67	2.2.2	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add Documents.....	22
69	2.2.2.1	Default "ipp-attribute-fidelity" if not supplied.....	22
70	2.2.2.2	Check that the Printer object is accepting jobs .....	23
71	2.2.2.3	Validate the values of the Job Template attributes .....	23
72	2.2.3	Algorithm for job validation .....	23
73	2.2.3.1	Check for conflicting Job Template attributes values.....	28
74	2.2.3.2	Decide whether to REJECT the request.....	28
75	2.2.3.3	For the Validate-Job operation, RETURN one of the success status codes .....	29
76	2.2.3.4	Create the Job object with attributes to support.....	29
77	2.2.3.5	Return one of the success status codes.....	30
78	2.2.3.6	Accept appended Document Content.....	31
79	2.2.3.7	Scheduling and Starting to Process the Job .....	31
80	2.2.3.8	Completing the Job .....	31
81	2.2.3.9	Destroying the Job after completion .....	31
82	2.2.3.10	Interaction with "ipp-attribute-fidelity" .....	31
83	2.3	Status codes returned by operation (Issue 1.50) .....	32
84	2.3.1	Printer Operations .....	32
85	2.3.1.1	Print-Job.....	32
86	2.3.1.2	Print-URI.....	33
87	2.3.1.3	Validate-Job .....	34
88	2.3.1.4	Create-Job .....	34
89	2.3.1.5	Get-Printer-Attributes .....	34
90	2.3.1.6	Get-Jobs .....	34
91	2.3.2	Job Operations.....	35
92	2.3.2.1	Send-Document.....	35
93	2.3.2.2	Send-URI .....	36
94	2.3.2.3	Cancel-Job.....	36
95	2.3.2.4	Get-Job-Attributes.....	37

96	2.4	Validate-Job .....	37
97	2.5	Case Sensitivity in URIs (issue 1.6).....	37
98	2.6	Character Sets, natural languages, and internationalization .....	38
99	2.6.1	Character set code conversion support (Issue 1.5).....	38
100	2.6.2	What charset to return when an unsupported charset is requested (Issue 1.19)?.....	39
101	2.6.3	Natural Language Override (NLO) (Issue 1.45).....	39
102	2.7	The "queued-job-count" Printer Description attribute .....	40
103	2.7.1	Why is "queued-job-count" RECOMMENDED (Issue 1.14)?.....	40
104	2.7.2	Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?.....	41
105	2.8	Sending empty attribute groups (Issue 1.16) .....	41
106	2.9	Returning unsupported attributes in Get-Xxxx responses (Issue 1.18) .....	41
107	2.10	Returning job-state in Print-Job response (Issue 1.30) .....	41
108	2.11	Flow controlling the data portion of a Print-Job request (Issue 1.22).....	42
109	2.12	Multi-valued attributes (Issue 1.31).....	42
110	2.13	Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)..	43
111	2.14	The 'none' value for empty sets (Issue 1.37).....	43
112	2.15	Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?.....	43
113	2.16	The "multiple-document-handling" Job Template attribute and support of multiple document jobs	
114		44	
115	3	Encoding and Transport.....	44
116	3.1	General Headers .....	45
117	3.2	Request Headers.....	45
118	3.3	Response Headers .....	46
119	3.4	Entity Headers.....	47
120	3.5	Optional support for HTTP/1.0.....	48
121	3.6	HTTP/1.1 Chunking.....	48
122	3.6.1	Disabling IPP Server Response Chunking.....	48
123	3.6.2	Warning About the Support of Chunked Requests.....	48
124	4	References .....	49
125	4.1	Authors' Address .....	50
126	5	Notices.....	<b>Error! Bookmark not defined.</b>
127	6	Change History.....	51
128	6.1	Changes to produce the February 12, 1999 version from the January 8, 1999 version:.....	51
129	6.2	Changes to produce the January 8, 1999 version from the December 6, 1998 version:.....	51
130	6.3	Changes to produce the December 6, 1998 version from the November 16, 1998 version: .....	52
131			

132

## 133 **1 Introduction**

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

### 141 1.1 Conformance language

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

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

### 153 1.2 Other terminology

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

## 156 **2 Model and Semantics**

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

### 158 2.1 Summary of Operation Attributes

159 Legend for the following table:

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

161 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
<b>Operation parameters--REQUIRED to be supplied by the sender</b>											
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
<b>Operation attributes--REQUIRED to be supplied by the sender</b>											
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	
<b>Operation attributes--RECOMMENDED to be supplied by the sender</b>											
job-name	R	R	R								
requesting-user-name	R	R	R	R	R		R	R	R	R	

162

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
<b>Operation attributes—OPTIONAL to be supplied by the sender</b>											
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.

163





164

## 165 2.2 Suggested Operation Processing Steps for IPP Objects (Issue 1.21)

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

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

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

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

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

### 192 2.2.1 Suggested Operation Processing Steps for all Operations

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

#### 196 2.2.1.1 Validate version number

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

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

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

#### 211 2.2.1.2 Validate operation identifier

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

#### 215 2.2.1.3 Validate the request identifier

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

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

#### 221 2.2.1.4 Validate attribute group and attribute presence and order

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

##### 223 2.2.1.4.1 Validate the presence and order of attribute groups

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

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

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

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

#### 236 2.2.1.4.2 Ignore unknown attribute groups in the expected position

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

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

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

#### 249 2.2.1.4.3 Validate the presence of a single occurrence of required Operation attributes

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

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

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

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

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

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

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

282

283

#### Operation Requests

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

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

287

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

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



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

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

430

431 Print-Job Response:

432 Print-URI Response:

433 Create-Job Response:

434 Send-Document Response:

435 Send-URI Response:

436 Group 1: Operation Attributes (R)

437 attributes-charset (R)

438 attributes-natural-language (R)

439 status-message (O\*)

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

441 <unsupported attributes> (R\*)

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

443 job-uri (R)

444 job-id (R)

445 job-state (R)

446 job-state-reasons (O\*)

447 job-state-message (O\*)

448 number-of-intervening-jobs (O\*)

449

450 Validate-Job Response:

451 Cancel-Job Response:

452 Group 1: Operation Attributes (R)

453 attributes-charset (R)

454 attributes-natural-language (R)

455 status-message (O\*)

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

457 <unsupported attributes> (R\*)

458

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

461

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



464  
465 Get-Printer-Attributes Response:  
466     Group 1: Operation Attributes (R)  
467         attributes-charset (R)  
468         attributes-natural-language (R)  
469         status-message (O\*)  
470     Group 2: Unsupported Attributes (R\*) (see Note 4)  
471         <unsupported attributes> (R\*)  
472     Group 3: Printer Object Attributes(R\*) (see Note 2)  
473         <requested attributes> (R\*)  
474

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

477  
478 Get-Job-Attributes Response:  
479     Group 1: Operation Attributes (R)  
480         attributes-charset (R)  
481         attributes-natural-language (R)  
482         status-message (O\*)  
483     Group 2: Unsupported Attributes (R\*) (see Note 4)  
484         <unsupported attributes> (R\*)  
485     Group 3: Job Object Attributes(R\*) (see Note 2)  
486         <requested attributes> (R\*)  
487

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

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

#### 500 2.2.1.5 Validate the values of the REQUIRED Operation attributes

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

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

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

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

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

525

526 -----  
attributes-charset (charset)

527 IF NOT a single non-empty 'charset' value, REJECT/RETURN 'client-error-request-bad-request'.  
528 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-long'.  
529 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-charset-  
530 not-supported".  
531

532 attributes-natural-language(naturalLanguage)

533 IF NOT a single non-empty 'naturalLanguage' value, REJECT/RETURN 'client-error-request-bad-  
534 request'.  
535 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-long'.  
536 ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-  
537 language-supported" attribute. If the supplied value is not a member of the Printer object's  
538 "generated-natural-language-supported" attribute, use the Printer object's "natural-language-  
539 configured" value.  
540

541 requesting-user-name

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

543 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.  
544 IF the IPP object can obtain a better authenticated name, use it instead.  
545

546 job-name(name)

547 IF NOT a single 'name' value, REJECT/RETURN 'client-error-request-bad-request'.  
548 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.  
549 IF NOT supplied by the client, the Printer object creates a name from the document-name or document-  
550 uri.  
551

552 document-name (name)

553 IF NOT a single 'name' value, REJECT/RETURN 'client-error-request-bad-request'.  
554 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.  
555

556 ipp-attribute-fidelity (boolean)

557 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-  
558 request'.  
559 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'.  
560 IF NOT supplied by the client, the IPP object assumes the value 'false'.  
561

562 document-format (mimeMediaType)

563 IF NOT a single non-empty 'mimeMediaType' value, REJECT/RETURN 'client-error-request-bad-  
564 request'.  
565 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.  
566 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN 'client-error-  
567 document-format-not-supported'.  
568 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's "document-  
569 format-default" attribute.  
570

571 document-uri (uri)

572 IF NOT a single non-empty 'uri' value, REJECT/RETURN 'client-error-request-bad-request'.  
573 IF the value length is greater than 1023 octets, REJECT/RETURN 'client-error-request-value-too-long'.  
574 IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.  
575 IF scheme is NOT in the Printer object's "reference-uri-schemes-supported" attribute,  
576 REJECT/RETURN 'client-error-uri-scheme-not-supported'.  
577 The Printer object MAY check to see if the document exists and is accessible. If the document is not  
578 found or is not accessible, REJECT/RETURN 'client-error-not found'.

579 last-document (boolean)

580 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-  
581 request'.  
582 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'.  
583

584 job-id (integer(1:MAX))  
585 IF NOT an single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN  
586 'client-error-bad-request'.  
587 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-error-  
588 gone' status code, if keep track of recently deleted jobs.  
589

590 requested-attributes (1setOf keyword)  
591 IF NOT one or more 'keyword' values, REJECT/RETURN 'client-error-request-bad-request'.  
592 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.  
593 Ignore unsupported values which are the keyword names of unsupported attributes. Don't bother to  
594 copy such requested (unsupported) attributes to the Unsupported Attribute response group since the  
595 response will not return them.  
596

597 which-jobs (type2 keyword)  
598 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-request-bad-request'.  
599 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.  
600 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to the  
601 Unsupported Attributes response group and REJECT/RETURN 'client-error-attributes-or-values-  
602 not-supported'.  
603 Note: a Printer still supports the 'completed' value even if it keeps no completed/canceled/aborted jobs:  
604 by returning no jobs when so queried.  
605 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.  
606

607 my-jobs (boolean)  
608 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-  
609 request'.  
610 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'.  
611 IF NOT supplied by the client, the IPP object assumes the 'false' value.  
612

613 limit (integer(1:MAX))  
614 IF NOT a single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN  
615 'client-error-bad-request'.  
616 IF NOT supplied by the client, the IPP object returns all jobs, no matter how many.  
617

618 -----  
619

620 2.2.1.6 Validate the values of the OPTIONAL Operation attributes  
621 OPTIONAL Operation attributes are those that an IPP object MAY or MAY NOT support. An IPP object  
622 validates the values of the OPTIONAL attributes supplied by the client. The IPP object performs the same  
623 syntactic validation checks for each OPTIONAL attribute value as in Section 2.2.1.5. As in Section

624 2.2.1.5, if any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' or the  
625 'client-error-request-value-too-long' status code.

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

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

633 -----

634 document-natural-language (naturalLanguage)

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

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

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

640

641 compression (type3 keyword)

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

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

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

647

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

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

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

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

654

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

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

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

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

661

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

663 IF NOT a single 'integer' value equal to 4 octets,  
664 REJECT/RETURN 'client-error-bad-request'.  
665 IF NOT in the range of the Printer object's "job-media-sheets-supported" attribute, copy the attribute  
666 and the unsupported value to the Unsupported Attributes response group and REJECT/RETURN  
667 'client-error-attributes-or-values-not-supported'.  
668

669 message (text(127))

670 IF NOT a single 'text' value, REJECT/RETURN 'client-error-request-bad-request'.  
671 IF the value length is greater than 127 octets,  
672 REJECT/RETURN 'client-error-request-value-too-long'.  
673

674 unknown or unsupported attribute

675 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute  
676 syntax, REJECT/RETURN 'client-error-request-value-too-long'.  
677 ELSE copy the attribute and value to the Unsupported Attributes response group and change the  
678 attribute value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.  
679

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

## 695 2.2.2 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add 696 Documents

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

### 700 2.2.2.1 Default "ipp-attribute-fidelity" if not supplied

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

703 2.2.2.2 Check that the Printer object is accepting jobs

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

706 2.2.2.3 Validate the values of the Job Template attributes

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

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

712 b) that the attribute syntax tag is correct for that attribute according to [IPP-MOD] Sections 4.2 to  
713 4.4.

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

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

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

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

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

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

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

730 2.2.3 Algorithm for job validation

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

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

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

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

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

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

749 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.

750 **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.

751

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

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



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

769 -----

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

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

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

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

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

780

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

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

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

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

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

788

789 job-sheets (type3 keyword | name)

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

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

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

794

795 multiple-document-handling (type2 keyword)

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

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

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

800

801 copies (integer(1:MAX))

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

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

804 IF NOT in range of the Printer object's "copies-supported" attribute

805 copy the attribute and the unsupported value to the Unsupported Attributes response group.

806

807 finishings (1setOf type2 enum)

808 IF NOT an 'enum' value(s) each with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-  
809 request'.

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

812

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

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

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

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

820

821 sides (type2 keyword)

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

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

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

826

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

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

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

832

833 orientation-requested (type2 enum)

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

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

838

839 media (type3 keyword | name)

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

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

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

844

845 printer-resolution (resolution)

846 IF NOT a single 'resolution' value with a length equal to 9 octets,  
847 REJECT/RETURN 'client-error-bad-request'.  
848 IF NOT in the Printer object's "printer-resolution-supported" attribute, copy the attribute and the  
849 unsupported value to the Unsupported Attributes response group.  
850

851 print-quality (type2 enum)

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

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

858 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute  
859 syntax,  
860 REJECT/RETURN 'client-error-bad-request' if the length of the attribute syntax is fixed or 'client-error-  
861 request-value-too-long' if the length of the attribute syntax is variable.  
862 ELSE copy the attribute and value to the Unsupported Attributes response group and change the  
863 attribute value to the "out-of-band" 'unsupported' value. Any remaining Job Template Attributes are  
864 either unknown or unsupported Job Template attributes and are validated algorithmically according  
865 to their attribute syntax for proper length (see below).  
866 -----

867

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

874	Name	Octet length check for read-write attributes
875	-----	-----
876	'textWithLanguage	<= 1023 AND 'naturalLanguage' <= 63
877	'textWithoutLanguage'	<= 1023
878	'nameWithLanguage'	<= 255 AND 'naturalLanguage' <= 63
879	'nameWithoutLanguage'	<= 255
880	'keyword'	<= 255
881	'enum'	= 4
882	'uri'	<= 1023
883	'uriScheme'	<= 63
884	'charset'	<= 63
885	'naturalLanguage'	<= 63
886	'mimeMediaType'	<= 255
887	'octetString'	<= 1023
888	'boolean'	= 1
889	'integer'	= 4
890	'rangeOfInteger'	= 8
891	'dateTime'	= 11
892	'resolution'	= 9
893	'lsetOf X'	
894		

### 895 2.2.3.1 Check for conflicting Job Template attributes values

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

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

### 909 2.2.3.2 Decide whether to REJECT the request

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

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

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

916

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

921 2.2.3.3 For the Validate-Job operation, RETURN one of the success status codes

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

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

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

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

929

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

934 2.2.3.4 Create the Job object with attributes to support

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

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

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

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

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

947

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

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

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

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

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

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

#### 979 2.2.3.5 Return one of the success status codes

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

- 981 (1) the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes or  
982 values.
- 983 (2) the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template  
984 attribute or values.
- 985 (3) the 'successful-ok-ignored-or-substituted-attributes' status code, if there are only unsupported Job  
986 Template attributes or values.

987 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this  
988 step. If the unsupported Operation attribute was a serious error, the above already rejected the request in a  
989 previous step. If control gets to this step with unsupported Operation attributes being returned, they are not  
990 serious errors.  
991

- 992 The Printer object also returns Job status attributes that indicate the initial state of the Job ('pending',  
993 'pending-held', 'processing', etc.), etc. See Print-Job Response, [IPP-MOD] section 3.2.1.2.
- 994 2.2.3.6 Accept appended Document Content
- 995 The Printer object accepts the appended Document Content data and either starts it printing, or spools it for  
996 later processing.
- 997 2.2.3.7 Scheduling and Starting to Process the Job
- 998 The Printer object uses its own configuration and implementation specific algorithms for scheduling the  
999 Job in the correct processing order. Once the Printer object begins processing the Job, the Printer changes  
1000 the Job's state to 'processing'. If the Printer object supports PDL override (the "pdl-override-supported"  
1001 attribute set to 'attempted'), the implementation does its best to see that IPP attributes take precedence over  
1002 embedded instructions in the document data.
- 1003 2.2.3.8 Completing the Job
- 1004 The Printer object continues to process the Job until it can move the Job into the 'completed' state. If an  
1005 Cancel-Job operation is received, the implementation eventually moves the Job into the 'canceled' state. If  
1006 the system encounters errors during processing that do not allow it to progress the Job into a completed  
1007 state, the implementation halts all processing, cleans up any resources, and moves the Job into the 'aborted'  
1008 state.
- 1009 2.2.3.9 Destroying the Job after completion
- 1010 Once the Job moves to the 'completed', 'aborted', or 'canceled' state, it is an implementation decision as to  
1011 when to destroy the Job object and release all associated resources. Once the Job has been destroyed, the  
1012 Printer would return either the "client-error-not-found" or "client-error-gone" status codes for operations  
1013 directed at that Job.
- 1014 Note: the Printer object SHOULD NOT re-use a "job-uri" or "job-id" value for a sufficiently long time  
1015 after a job has been destroyed, so that stale references kept by clients are less likely to access the wrong  
1016 (newer) job.
- 1017 2.2.3.10 Interaction with "ipp-attribute-fidelity"
- 1018 Some Printer object implementations may support "ipp-attribute-fidelity" set to 'true' and "pdl-override-  
1019 supported" set to 'attempted' and yet still not be able to realize exactly what the client specifies in the create  
1020 request. This is due to legacy decisions and assumptions that have been made about the role of job  
1021 instructions embedded within the document data and external job instructions that accompany the  
1022 document data and how to handle conflicts between such instructions. The inability to be 100% precise  
1023 about how a given implementation will behave is also compounded by the fact that the two special  
1024 attributes, "ipp-attribute-fidelity" and "pdl-override-supported", apply to the whole job rather than specific

1025 values for each attribute. For example, some implementations may be able to override almost all Job  
1026 Template attributes except for "number-up".

## 1027 2.3 Status codes returned by operation (Issue 1.50)

1028 This section lists all status codes once in the first operation (Print-Job). Then it lists the status codes that  
1029 are different or specialized for subsequent operations under each operation.

### 1030 2.3.1 Printer Operations

#### 1031 2.3.1.1 Print-Job

1032 The Printer object MUST return one of the following "status-code" values for the indicated reason.  
1033 Whether all of the document data has been accepted or not before returning the success or error response  
1034 depends on implementation. See Section 14 for a more complete description of each status code.

1035 For the following success status codes, the Job object has been created and the "job-id", and "job-uri"  
1036 assigned and returned in the response:

1037     successful-ok: no request attributes were substituted or ignored.

1038     successful-ok-ignored-or-substituted-attributes: some supplied (1) attributes were ignored or (2)  
1039         unsupported attribute syntaxes or values were substituted with supported values or were ignored.  
1040     Unsupported attributes, attribute syntaxes, or values MUST be returned in the Unsupported  
1041     Attributes group of the response.

1042     successful-ok-conflicting-attributes: some supplied attribute values conflicted with the values of other  
1043     supplied attributes and were either substituted or ignored. Attributes or values which conflict with  
1044     other attributes and have been substituted or ignored MUST be returned in the Unsupported  
1045     Attributes group of the response as supplied by the client.  
1046

1047 [ipp-mod] section 3.1.6 Operation Status Codes and Messages states (Issue 1.19):

1048     If the Printer object supports the "status-message" operation attribute, it SHOULD use the  
1049     REQUIRED 'utf-8' charset to return a status message for the following error status codes (see  
1050     section 14): 'client-error-bad-request', 'client-error-charset-not-supported', 'server-error-internal-  
1051     error', 'server-error-operation-not-supported', and 'server-error-version-not-supported'. In this case,  
1052     it MUST set the value of the "attributes-charset" operation attribute to 'utf-8' in the error response.

1053 For the following error status codes, no job is created and no "job-id" or "job-uri" is returned:

1054     client-error-bad-request: The request syntax does not conform to the specification.

1055     client-error-forbidden: The request is being refused for authorization or authentication reasons. The  
1056     implementation security policy is to not reveal whether the failure is one of authentication or  
1057     authorization.

1058     client-error-not-authenticated: Either the request requires authentication information to be supplied or  
1059     the authentication information is not sufficient for authorization.

1060     client-error-not-authorized: The requester is not authorized to perform the request on the target object.



1061 client-error-not-possible: The request cannot be carried out because of the state of the system. See also  
1062 'server-error-not-accepting-jobs' status code which MUST take precedence if the Printer object's  
1063 "printer-accepting-jobs" attribute is 'false'.  
1064 client-error-timeout: not applicable.  
1065 client-error-not-found: the target object does not exist.  
1066 client-error-gone: the target object no longer exists and no forwarding address is known.  
1067 client-error-request-entity-too-large: the size of the request and/or print data exceeds the capacity of the  
1068 IPP Printer to process it.  
1069 client-error-request-value-too-long: the size of request variable length attribute values, such as 'text'  
1070 and 'name' attribute syntaxes, exceed the maximum length specified in [IPP-MOD] for the attribute  
1071 and MUST be returned in the Unsupported Attributes Group.  
1072 client-error-document-format-not-supported: the document format supplied is not supported. The  
1073 "document-format" attribute with the unsupported value MUST be returned in the Unsupported  
1074 Attributes Group. This error SHOULD take precedence over any other 'xxx-not-supported' error,  
1075 except 'client-error-charset-not-supported'.  
1076 client-error-attributes-or-values-not-supported: one or more supplied attributes, attribute syntaxes, or  
1077 values are not supported and the client supplied the "ipp-attributes-fidelity" operation attribute with  
1078 a 'true' value. They MUST be returned in the Unsupported Attributes Group as explained below.  
1079 client-error-uri-scheme-not-supported: not applicable.  
1080 client-error-charset-not-supported: the charset supplied in the "attributes-charset" operation attribute is  
1081 not supported. The Printer's "configured-charset" MUST be returned in the response as the value of  
1082 the "attributes-charset" operation attribute and used for any 'text' and 'name' attributes returned in  
1083 the error response. This error SHOULD take precedence over any other error, unless the request  
1084 syntax is so bad that the client's supplied "attributes-charset" cannot be determined.  
1085 client-error-conflicting-attributes: one or more supplied attribute values conflicted with each other and  
1086 the client supplied the "ipp-attributes-fidelity" operation attribute with a 'true' value. They MUST  
1087 be returned in the Unsupported Attributes Group as explained below.  
1088 server-error-internal-error: an unexpected condition prevents the request from being fulfilled.  
1089 server-error-operation-not-supported: not applicable (since Print-Job is REQUIRED).  
1090 server-error-service-unavailable: the service is temporarily overloaded.  
1091 server-error-version-not-supported: the version in the request is not supported. The "closest" version  
1092 number supported MUST be returned in the response.  
1093 server-error-device-error: a device error occurred while receiving or spooling the request or document  
1094 data or the IPP Printer object can only accept one job at a time.  
1095 server-error-temporary-error: a temporary error such as a buffer full write error, a memory overflow, or  
1096 a disk full condition occurred while receiving the request and/or the document data.  
1097 server-error-not-accepting-jobs: the Printer object's "printer-is-not-accepting-jobs" attribute is 'false'.  
1098 server-error-busy: the Printer is too busy processing jobs to accept another job at this time.  
1099 server-error-job-canceled: the job has been canceled by an operator or the system while the client was  
1100 transmitting the document data.

### 1101 2.3.1.2 Print-URI

1102 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Print-  
1103 URI with the following specializations and differences. See Section 14 for a more complete description of  
1104 each status code.

- 1105 server-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation  
1106 attribute is not supported and is returned in the Unsupported Attributes group.
- 1107 2.3.1.3 Validate-Job
- 1108 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Validate-  
1109 Job. See Section 14 for a more complete description of each status code.
- 1110 2.3.1.4 Create-Job
- 1111 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Create-  
1112 Job with the following specializations and differences. See Section 14 for a more complete description of  
1113 each status code.
- 1114 server-error-operation-not-supported: the Create-Job operation is not supported.
- 1115 2.3.1.5 Get-Printer-Attributes
- 1116 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-  
1117 Printer-Attributes operation with the following specializations and differences. See Section 14 for a more  
1118 complete description of each status code.
- 1119 For the following success status codes, the requested attributes are returned in Group 3 in the response:
- 1120 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested  
1121 attributes were unsupported.
- 1122 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"  
1123 operation attribute MAY, but NEED NOT, be returned with the unsupported values.
- 1124 successful-ok-conflicting-attributes: same as Print-Job.
- 1125 For the error status codes, Group 3 is returned containing no attributes or is not returned at all:
- 1126 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.
- 1127 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.
- 1128 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes  
1129 MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.
- 1130 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.
- 1131 server-error-operation-not-supported: not applicable (since Get-Printer-Attributes is REQUIRED).
- 1132 server-error-device-error: same as Print-Job, except that no document data is involved.
- 1133 server-error-temporary-error: same as Print-Job, except that no document data is involved.
- 1134 server-error-not-accepting-jobs: not applicable..
- 1135 server-error-busy: same as Print-Job, except the IPP object is too busy to accept even query requests.
- 1136 server-error-job-canceled: not applicable..
- 1137 2.3.1.6 Get-Jobs
- 1138 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-  
1139 Jobs operation with the following specializations and differences. See Section 14 for a more complete  
1140 description of each status code.

- 1141 For the following success status codes, the requested attributes are returned in Group 3 in the response:
- 1142     successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested  
1143     attributes were unsupported.
- 1144     successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"  
1145     operation attribute MAY, but NEED NOT, be returned with the unsupported values.
- 1146     successful-ok-conflicting-attributes: same as Print-Job.
- 1147 For any error status codes, Group 3 is returned containing no attributes or is not returned at all. The  
1148 following brief error status code descriptions contain unique information for use with Get-Jobs operation.  
1149 See section 14 for the other error status codes that apply uniformly to all operations:
- 1150     client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.  
1151     client-error-request-entity-too-large: same as Print-job, except that no print data is involved.  
1152     client-error-document-format-not-supported: not applicable.  
1153     client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes  
1154     MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.  
1155     client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.  
1156     server-error-operation-not-supported: not applicable (since Get-Jobs is REQUIRED).  
1157     server-error-device-error: same as Print-Job, except that no document data is involved.  
1158     server-error-temporary-error: same as Print-Job, except that no document data is involved.  
1159     server-error-not-accepting-jobs: not applicable.  
1160     server-error-job-canceled: not applicable.

## 1161 2.3.2 Job Operations

### 1162 2.3.2.1 Send-Document

- 1163 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-  
1164 Printer-Attributes operation with the following specializations and differences. See Section 14 for a more  
1165 complete description of each status code.
- 1166 For the following success status codes, the document has been added to the specified Job object and the  
1167 job's "number-of-documents" attribute has been incremented:
- 1168     successful-ok: no request attributes were substituted or ignored (same as Print-Job).  
1169     successful-ok-ignored-or-substituted-attributes: same as Print-Job.  
1170     successful-ok-conflicting-attributes: same as Print-Job.
- 1171 For the error status codes, no document has been added to the Job object and the job's "number-of-  
1172 documents" attribute has not been incremented:
- 1173     client-error-not-possible: Same as Print-Job, except that the Printer's "printer-is-accepting-jobs"  
1174     attribute is not involved, so that the client is able to finish submitting a multi-document job after this  
1175     attribute has been set to 'true'. Another condition is that the state of the job precludes Send-  
1176     Document, i.e., the job has already been closed out by the client. However, if the IPP Printer closed  
1177     out the job due to timeout, the 'client-error-timeout' error status SHOULD be returned instead.  
1178     client-error-timeout: This request was sent after the Printer closed the job, because it has not received a  
1179     Send-Document or Send-URI operation within the Printer's "multiple-operation-time-out" period .  
1180     client-error-request-entity-too-large: same as Print-Job.

- 1181 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attributes-fidelity" operation  
1182 attribute is not involved..
- 1183 server-error-operation-not-supported: the Send-Document request is not supported.
- 1184 server-error-not-accepting-jobs: not applicable.
- 1185 server-error-job-canceled: the job has been canceled by an operator or the system while the client was  
1186 transmitting the data.
- 1187 2.3.2.2 Send-URI
- 1188 All of the Print-Job status code descriptions in Section 3.2.1.2 Print-Job Response with the specializations  
1189 described for Send-Document are applicable to Send-URI. See Section 14 for a more complete description  
1190 of each status code.
- 1191 server-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation  
1192 attribute is not supported and the "document-uri" attribute MUST be returned in the Unsupported  
1193 Attributes group.
- 1194 2.3.2.3 Cancel-Job
- 1195 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Cancel-  
1196 Job with the following specializations and differences. See Section 14 for a more complete description of  
1197 each status code.
- 1198 For the following success status codes, the Job object is being canceled or has been canceled:
- 1199 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
- 1200 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
- 1201 successful-ok-conflicting-attributes: same as Print-Job.
- 1202
- 1203 For any of the error status codes, the Job object has not been canceled or was previously canceled.
- 1204 client-error-not-possible: The request cannot be carried out because of the state of the Job object  
1205 ('completed', 'canceled', or 'aborted') or the state of the system.
- 1206 client-error-not-found: the target Printer and/or Job object does not exist.
- 1207 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding address is  
1208 known.
- 1209 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
- 1210 client-error-document-format-not-supported: not applicable.
- 1211 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes  
1212 and values MUST be ignored.
- 1213 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-  
1214 jobs" attribute is not involved.
- 1215 server-error-operation-not-supported: not applicable (Cancel-Job is REQUIRED).
- 1216 server-error-device-error: same as Print-Job, except no document data is involved.
- 1217 server-error-temporary-error: same as Print-Job, except no document data is involved.
- 1218 server-error-not-accepting-jobs: not applicable..
- 1219 server-error-job-canceled: not applicable.

## 1220 2.3.2.4 Get-Job-Attributes

1221 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Get-Job-  
1222 Attributes with the following specializations and differences. See Section 14 for a more complete  
1223 description of each status code.

1224 For the following success status codes, the requested attributes are returned in Group 3 in the response:

1225     successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested  
1226     attributes were unsupported.

1227     successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"  
1228     operation attribute MAY, but NEED NOT, be returned with the unsupported values.

1229     successful-ok-conflicting-attributes: same as Print-Job.

1230 For the error status codes, Group 3 is returned containing no attributes or is not returned at all.

1231     client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.

1232     client-error-document-format-not-supported: not applicable.

1233     client-error-attributes-or-values-not-supported: not applicable.

1234     client-error-uri-scheme-not-supported: not applicable.

1235     client-error-conflicting-attributes: not applicable

1236     server-error-operation-not-supported: not applicable (since Get-Job-Attributes is REQUIRED).

1237     server-error-device-error: same as Print-Job, except no document data is involved.

1238     server-error-temporary-error: sane 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 2.4 Validate-Job

1242 The Validate-Job operation has been designed so that its implementation may be a part of the Print-Job  
1243 operation. Therefore, requiring Validate-Job is not a burden on implementers. Also it is useful for client's  
1244 to be able to count on its presence in all conformance implementations, so that the client can determine  
1245 before sending a long document, whether the job will be accepted by the IPP Printer or not.

## 1246 2.5 Case Sensitivity in URIs (issue 1.6)

1247 IPP client and server implementations must be aware of the diverse uppercase/lowercase nature of URIs.

1248 RFC 2396 defines URL schemes and Host names as case insensitive but reminds us that the rest of the

1249 URL may well demonstrate case sensitivity. When creating URL's for fields where the choice is

1250 completely arbitrary, it is probably best to select lower case. However, this cannot be guaranteed and

1251 implementations MUST NOT rely on any fields being case-sensitive or case-insensitive in the URL beyond

1252 the URL scheme and host name fields.

1253 The reason that the IPP specification does not make any restrictions on URIs, is so that implementations of  
1254 IPP may use off-the-shelf components that conform to the standards that define URIs, such as RFC 2396  
1255 and the HTTP/1.1 specifications [RFC2068]. See these specifications for rules of matching, comparison,  
1256 and case-sensitivity.

- 1257 It is also recommended that System Administrators and implementations avoid creating URLs for different  
1258 printers that differ only in their case. For example, don't have Printer1 and printer1 as two different IPP  
1259 Printers.
- 1260 The HTTP/1.1 specification [RFC2068] contains more details on comparing URLs.
- 1261 2.6 Character Sets, natural languages, and internationalization
- 1262 This section discusses character set support, natural language support and internationalization.
- 1263 2.6.1 Character set code conversion support (Issue 1.5)
- 1264 IPP clients and IPP objects are REQUIRED to support UTF-8. They MAY support additional charsets. It  
1265 is RECOMMENDED that an IPP object also support US-ASCII, since many clients support US-ASCII,  
1266 and indicate that UTF-8 and US-ASCII are supported by populating the Printer's "charset-supported" with  
1267 'utf-8' and 'us-ascii' values. An IPP object is required to code covert with as little loss as possible between  
1268 the charsets that it supports, as indicated in the Printer's "charsets-supported" attribute.
- 1269 How should the server handle the situation where the "attributes-charset" of the response itself is "us-ascii",  
1270 but one or more attributes in that response is in the "utf-8" format?
- 1271 Example: Consider a case where a client sends a Print-Job request with "utf-8" as the value of "attributes-  
1272 charset" and with the "job-name" attribute supplied. Later another client submits a Get-Job-Attribute or  
1273 Get-Jobs request. This second request contains the "attributes-charset" with value "us-ascii" and  
1274 "requested-attributes" attribute with exactly one value "job-name".
- 1275 According to the IPP-Mod document (section 3.1.4.2), the value of the "attributes-charset" for the response  
1276 of the second request must be "us-ascii" since that is the charset specified in the request. The "job-name"  
1277 value, however, is in "utf-8" format. Should the request be rejected even though both "utf-8" and "us-ascii"  
1278 charsets are supported by the server? or should the "job-name" value be converted to "us-ascii" and return  
1279 "successful-ok-conflicting-attributes" (0x0002) as the status code?
- 1280 Answer: An IPP object that supports both utf-8 (REQUIRED) and us-ascii, the second paragraph of  
1281 section 3.1.4.2 applies so that the IPP object MUST accept the request, perform code set conversion  
1282 between these two charsets with "the highest fidelity possible" and return 'successful-ok', rather than a  
1283 warning 'successful-ok-conflicting-attributes, or an error. The printer will do the best it can to convert  
1284 between each of the character sets that it supports--even if that means providing a string of question marks  
1285 because none of the characters are representable in US ASCII. If it can't perform such conversion, it  
1286 MUST NOT advertise us-ascii as a value of its "attributes-charset-supported" and MUST reject any request  
1287 that requests 'us-ascii'.
- 1288 One IPP object implementation strategy is to convert all request text and name values to a Unicode internal  
1289 representation. This is 16-bit and virtually universal. Then convert to the specified operation attributes-  
1290 charset on output.

1291 Also it would be smarter for a client to ask for 'utf-8', rather than 'us-ascii' and throw away characters that it  
1292 doesn't understand, rather than depending on the code conversion of the IPP object.

1293 2.6.2 What charset to return when an unsupported charset is requested (Issue 1.19)?

1294 Section 3.1.4.1 Request Operation attributes was clarified in November 1998 as follows:

1295 All clients and IPP objects MUST support the 'utf-8' charset [RFC2044] and MAY support  
1296 additional charsets provided that they are registered with IANA [IANA-CS]. If the Printer object  
1297 does not support the client supplied charset value, the Printer object MUST reject the request, set  
1298 the "attributes-charset" to 'utf-8' in the response, and return the 'client-error-charset-not-supported'  
1299 status code and any 'text' or 'name' attributes using the 'utf-8' charset.

1300 Since the client and IPP object MUST support UTF-8, returning any text or name attributes in UTF-8 when  
1301 the client requests a charset that is not supported should allow the client to display the text or name.

1302 Since such an error is a client error, rather than a user error, the client should check the status code first so  
1303 that it can avoid displaying any other returned 'text' and 'name' attributes that are not in the charset  
1304 requested.

1305 Furthermore, [ipp-mod] section 14.1.4.14 client-error-charset-not-supported (0x040D) was clarified in  
1306 November 1998 as follows:

1307 For any operation, if the IPP Printer does not support the charset supplied by the client in the  
1308 "attributes-charset" operation attribute, the Printer MUST reject the operation and return this status  
1309 and any 'text' or 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1).

1310 2.6.3 Natural Language Override (NLO) (Issue 1.45)

1311 The 'text' and 'name' attributes each have two forms. One has an implicit natural language, and the other  
1312 has an explicit natural language. The 'textWithoutLanguage' and 'textWithLanguage' are the two 'text'  
1313 forms. The 'nameWithoutLanguage' and 'nameWithLanguage' are the two 'name' forms. If a receiver (IPP  
1314 object or IPP client) supports an attribute with attribute syntax 'text', it MUST support both forms in a  
1315 request and a response. A sender (IPP client or IPP object) MAY send either form for any such attribute.  
1316 When a sender sends a WithoutLanguage form, the implicit natural language is specified in the "attributes-  
1317 natural-language" operation attribute which all senders MUST include in every request and response.

1318 When a sender sends a WithLanguage form, it MAY be different from the implicit natural language  
1319 supplied by the sender or it MAY be the same. The receiver MUST treat either form equivalently.

1320 There is an implementation decision for senders, whether to always send the WithLanguage forms or use  
1321 the WithoutLanguage form when the attribute's natural language is the same as the request or response.  
1322 The former approach makes the sender implementation simpler. The latter approach is more efficient on  
1323 the wire and allows inter-working with non-conforming receivers that fail to support the WithLanguage  
1324 forms. As each approach have advantages, the choice is completely up to the implementer of the sender.

1325 Furthermore, when a client receives a 'text' or 'name' job attribute that it had previously supplied, that client  
1326 MUST NOT expect to see the attribute in the same form, i.e., in the same WithoutLanguage or  
1327 WithLanguage form as the client supplied when it created the job. The IPP object is free to transform the  
1328 attribute from the WithLanguage form to the WithoutLanguage form and vice versa, as long as the natural  
1329 language is preserved. However, in order to meet this latter requirement, it is usually simpler for the IPP  
1330 object implementation to store the natural language explicitly with the attribute value, i.e., to store using an  
1331 internal representation that resembles the WithLanguage form.

1332 The IPP Printer MUST copy the natural language of a job, i.e., the value of the "attributes-natural-  
1333 language" operation attribute supplied by the client in the create operation, to the Job object as a Job  
1334 Description attribute, so that a client is able to query it. In returning a Get-Job-Attributes response, the IPP  
1335 object MAY return one of three natural language values in the response's "attributes-natural-language"  
1336 operation attribute: (1) that requested by the requester, (2) the natural language of the job, or (3) the  
1337 configured natural language of the IPP Printer, if the requested language is not supported by the IPP  
1338 Printer.

1339 This "attributes-natural-language" Job Description attribute is useful for an IPP object implementation that  
1340 prints start sheets in the language of the user who submitted the job. This same Job Description attribute is  
1341 useful to a multi-lingual operator who has to communicate with different job submitters in different natural  
1342 languages. This same Job Description attribute is expected to be used in the future to generate notification  
1343 messages in the natural language of the job submitter.

1344 Early drafts of [IPP-MOD] contained a job-level natural language override (NLO) for the Get-Jobs  
1345 response. A job-level (NLO) is an (unrequested) Job Attribute which then specified the implicit natural  
1346 language for any other WithoutLanguage job attributes returned in the response for that job.  
1347 Interoperability testing of early implementations showed that no one was implementing the job-level NLO  
1348 in Get-Job responses. So the job-level NLO was eliminated from the Get-Jobs response. This  
1349 simplification makes all requests and responses consistent in that the implicit natural language for any  
1350 WithoutLanguage 'text' or 'name' form is always supplied in the request's or response's "attributes-natural-  
1351 language" operation attribute.

1352 2.7 The "queued-job-count" Printer Description attribute

1353 2.7.1 Why is "queued-job-count" RECOMMENDED (Issue 1.14)?

1354 The reason that "queued-job-count" is RECOMMENDED, is that some clients look at that attribute alone  
1355 when summarizing the status of a list of printers, instead of doing a Get-Jobs to determine the number of  
1356 jobs in the queue. Implementations that fail to support the "queued-job-count" will cause that client to  
1357 display 0 jobs when there are actually queued jobs.

1358 We would have made it a REQUIRED Printer attribute, but some implementations had already been  
1359 completed before the issue was raised, so making it a SHOULD was a compromise.



1360 2.7.2 Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?

1361 The "queued-job-count" is not a good measure of how busy the printer is when there are held jobs. A  
1362 future registration could be to add a "held-job-count" (or an "active-job-count") Printer Description  
1363 attribute if experience shows that such an attribute (combination) is needed to quickly indicate how busy a  
1364 printer really is.

1365 2.8 Sending empty attribute groups (Issue 1.16)

1366 The [IPP-MOD] and [IPP-PRO] specifications RECOMMEND that a sender not send an empty attribute  
1367 group in a request or a response. However, they REQUIRE a receiver to accept an empty attribute group as  
1368 equivalent to the omission of that group. So a client SHOULD omit the Job Template Attributes group  
1369 entirely in a create operation that is not supplying any Job Template attributes. Similarly, an IPP object  
1370 SHOULD omit an empty Unsupported Attributes group if there are no unsupported attributes to be returned  
1371 in a response.

1372 The [IPP-PRO] specification REQUIRES a receiver to be able to receive either an empty attribute group or  
1373 an omitted attribute group and treat them equivalently. The term "receiver" means an IPP object for a  
1374 request and a client for a response. The term "sender" means a client for a request and an IPP object for a  
1375 response.

1376 There is an exception to the rule for Get-Jobs when there are no attributes to be returned. [ipp-pro]  
1377 contains the following paragraph:

1378       The syntax allows an xxx-attributes-tag to be present when the xxx-attribute-sequence that follows  
1379       is empty. The syntax is defined this way to allow for the response of Get-Jobs where no attributes  
1380       are returned for some job-objects. Although it is RECOMMENDED that the sender not send an  
1381       xxx-attributes-tag if there are no attributes (except in the Get-Jobs response just mentioned), the  
1382       receiver MUST be able to decode such syntax.

1383 2.9 Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)

1384 In the Get-Printer-Attributes, Get-Jobs, or Get-Job-Attributes responses, the client cannot depend on getting  
1385 unsupported attributes returned in the Unsupported Attributes group that the client requested, but are not  
1386 supported by the IPP object. However, such unsupported requested attributes will not be returned in the  
1387 Job Attributes or Printer Attributes group (since they are unsupported). Furthermore, the IPP object is  
1388 REQUIRED to return the 'successful-ok-ignored-or-substituted-attributes' status code, so that the client  
1389 knows that not all that was requested has been returned.

1390 2.10 Returning job-state in Print-Job response (Issue 1.30)

1391 An IPP client submits a small job via Print-Job. By the time the IPP printer/print server is putting together  
1392 a response to the operation, the job has finished printing and been removed as an object from the print  
1393 system. What should the job-state be in the response?

1394 The Model suggests that the Printer return a response before it even accepts the document content. The Job  
1395 Object Attributes are returned only if the IPP object returns one of the success status codes. Then the job-  
1396 state would always be "pending" or "pending-held".

1397 This issue comes up for the implementation of an IPP Printer object as a server that forwards jobs to  
1398 devices that do not provide job status back to the server. If the server is reasonably certain that the job  
1399 completed successfully, then it should return the job-state as 'completed'. Also the server can keep the job  
1400 in its "job history" long after the job is no longer in the device. Then a user could query the server and see  
1401 that the job was in the 'completed' state and completed as specified by the job's "time-at-completed" time  
1402 which would be the same as the server submitted the job to the device.

1403 An alternative is for the server to respond to the client before or while sending the job to the device, instead  
1404 of waiting until the server has finished sending the job to the device. In this case, the server can return the  
1405 job's state as 'pending' with the 'job-outgoing' value in the job's "job-state-reasons" attribute.

1406 If the server doesn't know for sure whether the job completed successfully (or at all), it could return the  
1407 (out-of-band) 'unknown' value.

1408 On the other hand, if the server is able to query the device and/or setup some sort of event notification that  
1409 the device initiates when the job makes state transitions, then the server can return the current job state in  
1410 the Print-Job response and in subsequent queries because the server knows what the job state is in the  
1411 device (or can query the device).

1412 All of these alternatives depend on implementation of the server and the device.

#### 1413 2.11 Flow controlling the data portion of a Print-Job request (Issue 1.22)

1414 A paused printer (or one that is stopped due to paper out or jam or spool space full or buffer space full, may  
1415 flow control the data of a Print-Job operation (at the TCP/IP layer), so that the client is not able to send all  
1416 the document data. Consequently, the Printer will not return a response until the condition is changed.

1417 The Printer should not return a Print-Job response with an error code in any of these conditions, since either  
1418 the printer will be resumed and/or the condition will be freed either by human intervention or as jobs print.

1419 In writing test scripts to test IPP Printers, the script must also be written not to expect a response, if the  
1420 printer has been paused, until the printer is resumed, in order to work with all possible implementations.

#### 1421 2.12 Multi-valued attributes (Issue 1.31)

1422 What is the attribute syntax for a multi-valued attribute? Since some attributes support values in more than  
1423 one data type, such as "media", "job-hold-until", and "job-sheets", IPP semantics associate the attribute  
1424 syntax with each value, not with the attribute as a whole. The protocol associates the attribute syntax tag  
1425 with each value. Don't be fooled, just because the attribute syntax tag comes before the attribute keyword.  
1426 All attribute values after the first have a zero length attribute keyword as the indication of a subsequent  
1427 value of the same attribute.

1428 2.13 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)

1429 The following clarification was added to [ipp-mod] section 8.5:

1430 8.5 Queries on jobs submitted using non-IPP protocols

1431 If the device that an IPP Printer is representing is able to accept jobs using other job submission  
1432 protocols in addition to IPP, it is RECOMMEND that such an implementation at least allow such  
1433 "foreign" jobs to be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an  
1434 implementation NEED NOT support all of the same IPP job attributes as for IPP jobs. The IPP  
1435 object returns the 'unknown' out-of-band value for any requested attribute of a foreign job that is  
1436 supported for IPP jobs, but not for foreign jobs.

1437 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such  
1438 "foreign jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-  
1439 Attributes and Cancel-Job. Such an implementation also needs to deal with the problem of  
1440 authentication of such foreign jobs. One approach would be to treat all such foreign jobs as  
1441 belonging to users other than the user of the IPP client. Another approach would be for the foreign  
1442 job to belong to 'anonymous'. Only if the IPP client has been authenticated as an operator or  
1443 administrator of the IPP Printer object, could the foreign jobs be queried by an IPP request.  
1444 Alternatively, if the security policy is to allow users to query other users' jobs, then the foreign jobs  
1445 would also be visible to an end-user IPP client using Get-Jobs and Get-Job-Attributes.

1446 Thus IPP MAY be implemented as a "universal" protocol that provides access to jobs submitted with any  
1447 job submission protocol. As IPP becomes widely implemented, providing a more universal access makes  
1448 sense.

1449 2.14 The 'none' value for empty sets (Issue 1.37)

1450 [ipp-mod] states that the 'none' value should be used as the value of a 1SetOf when the set is empty. In most  
1451 cases, sets that are potentially empty contain keywords so the keyword 'none' is used, but for the 3  
1452 finishings attributes, the values are enums and thus the empty set is represented by the enum 3. Currently  
1453 there are no other attributes with 1SetOf values which can be empty and can contain values that are not  
1454 keywords. This exception requires special code and is a potential place for bugs. It would have been better  
1455 if we had chosen an out-of-band value, either "no-value" or some new value, such as 'none'. Since we  
1456 didn't, implementations have to deal with the different representations of 'none', depending on the attribute  
1457 syntax.

1458 2.15 Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?

1459 In [ipp-mod] section 3.2.6.1 'Get-Jobs Request', if the attribute 'my-jobs' is present and set to TRUE, MUST  
1460 the 'requesting-user-name' attribute be there to, and if it's not present what should the IPP printer do?

1461 [ipp-mod] Section 8.3 describes the various cases of "requesting-user-name" being present or not for any  
1462 operation. If the client does not supply a value for "requesting-user-name", the printer MUST assume that  
1463 the client is supplying some anonymous name, such as "anonymous".

1464 2.16 The "multiple-document-handling" Job Template attribute and support of multiple document jobs

1465 ISSUE: IPP/1.0 is silent on which of the four effects an implementation would perform if it supports  
1466 Create-Job, but does not support "multiple-document-handling".

1467 A fix to IPP/1.0 would be to require implementing all four values of "multiple-document-handling" if  
1468 Create-Job is supported at all. Or at least 'single-document-new-sheet' and 'separate-documents-uncollated-  
1469 copies'. In any case, an implementation that supports Create-Job SHOULD also support "multiple-  
1470 document-handling". Support for all four values is RECOMMENDED, but at least the 'single-document-  
1471 new-sheet' and 'separate-documents-uncollated-copies' values, along with the "multiple-document-  
1472 handling-default" indicating the default behavior and "multiple-document-handling-supported" values. If  
1473 an implementation spools the data, it should also support the 'separate-documents-collated-copies' value as  
1474 well.

### 1475 3 Encoding and Transport

1476 This section discusses various aspects of IPP/1.0 Encoding and Transport [IPP-PRO].

1477 A server is not required to send a response until after it has received the client's entire request. Hence, a  
1478 client must not expect a response until after it has sent the entire request. However, we recommend that the  
1479 server return a response as soon as possible if an error is detected while the client is still sending the data,  
1480 rather than waiting until all of the data is received. Therefore, we also recommend that a client listen for an  
1481 error response that an IPP server MAY send before it receives all the data. In this case a client, if chunking  
1482 the data, can send a premature zero-length chunk to end the request before sending all the data (and so the  
1483 client can keep the connection open for other requests, rather than closing it). If the request is blocked for  
1484 some reason, a client MAY determine the reason by opening another connection to query the server using  
1485 Get-Printer-Attributes.

1486 In the following sections, there are a tables of all HTTP headers which describe their use in an IPP client or  
1487 server. The following is an explanation of each column in these tables.

- 1488 • the "header" column contains the name of a header
- 1489 • the "request/client" column indicates whether a client sends the header.
- 1490 • the "request/ server" column indicates whether a server supports the header when received.
- 1491 • the "response/ server" column indicates whether a server sends the header.
- 1492 • the "response /client" column indicates whether a client supports the header when received.
- 1493 • the "values and conditions" column specifies the allowed header values and the conditions for the  
1494 header to be present in a request/response.

1495 The table for "request headers" does not have columns for responses, and the table for "response headers"  
1496 does not have columns for requests.

1497 The following is an explanation of the values in the "request/client" and "response/ server" columns.

- 1498 • **must:** the client or server MUST send the header,

- 1499       • **must-if:** the client or server **MUST** send the header when the condition described in the “values and  
1500       conditions” column is met,  
1501       • **may:** the client or server **MAY** send the header  
1502       • **not:** the client or server **SHOULD NOT** send the header. It is not relevant to an IPP  
1503       implementation.

1504       The following is an explanation of the values in the “response/client” and “request/ server” columns.

- 1505       • **must:** the client or server **MUST** support the header,  
1506       • **may:** the client or server **MAY** support the header  
1507       • **not:** the client or server **SHOULD NOT** support the header. It is not relevant to an IPP  
1508       implementation.

### 1509    3.1   General Headers

1510       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 <b>SHOULD</b> keep a connection for the duration of a sequence of operations. The client and server <b>MUST</b> include this header for the last operation in such a sequence.
Date	may	may	must	may	per RFC 1123 [RFC1123] from RFC 2068 [RFC2068]
Pragma	must	not	must	not	“no-cache” only
Transfer-Encoding	must-if	must	must-if	must	“chunked” only . Header <b>MUST</b> be present if Content-Length is absent.
Upgrade	not	not	not	not	
Via	not	not	not	not	

### 1511    3.2   Request Headers

1512       The following is a table for the request headers.

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

<b>Request-Header</b>	<b>Client</b>	<b>Server</b>	<b>Request Values and Conditions</b>
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 and a “Proxy-Authenticate” header.
Range	not	not	
Referer	not	not	
User-Agent	not	not	

## 1513 3.3 Response Headers

1514 The following is a table for the request headers.

<b>Response-Header</b>	<b>Server</b>	<b>Client</b>	<b>Response Values and Conditions</b>
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.

## 1515 3.4 Entity Headers

1516 The following is a table for the entity headers.

<b>Entity-Header</b>	<b>Request</b>		<b>Response</b>		<b>Values and Conditions</b>
	<b>Client</b>	<b>Server</b>	<b>Server</b>	<b>Client</b>	
Allow	not	not	not	not	
Content-Base	not	not	not	not	
Content-Encoding	may	must	must	must	per RFC 2068 and IANA registry for content codings.
Content-Language	not	not	not	not	Application/ipp handles language
Content-Length	must-if	must	must-if	must	the length of the message-body per RFC 2068. Header MUST be present if Transfer-Encoding is absent..
Content-Location	not	not	not	not	
Content-MD5	may	may	may	may	per RFC 2068
Content-Range	not	not	not	not	

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

### 1517 3.5 Optional support for HTTP/1.0

1518 IPP implementations consist of an HTTP layer and an IPP layer. In the following discussion, the term  
 1519 "client" refers to the HTTP client layer and the term "server" refers to the HTTP server layer. The  
 1520 Encoding and Transport document [IPP-PRO] requires that HTTP 1.1 **MUST** be supported by all clients  
 1521 and all servers. However, a client and/or a server implementation may choose to also support HTTP 1.0.

- 1522 • This option means that a server may choose to communicate with a (non-conforming) client that only  
 1523 supports HTTP 1.0. In such cases the server should not use any HTTP 1.1 specific parameters or  
 1524 features and should respond using HTTP version number 1.0.
- 1525 • This option also means that a client may choose to communicate with a (non-conforming) server that  
 1526 only supports HTTP 1.0. In such cases, if the server responds with an HTTP ‘unsupported version  
 1527 number’ to an HTTP 1.1 request, the client should retry using HTTP version number 1.0.

### 1528 3.6 HTTP/1.1 Chunking

#### 1529 3.6.1 Disabling IPP Server Response Chunking

1530 Clients **MUST** anticipate that the HTTP/1.1 server may chunk responses and **MUST** accept them in  
 1531 responses. However, a (non-conforming) HTTP client that is unable to accept chunked responses may  
 1532 attempt to request an HTTP 1.1 server not to use chunking in its response to an operation by using the  
 1533 following HTTP header:

1534       TE: identity

1535 This mechanism should not be used by a server to disable a client from chunking a request, since chunking  
 1536 of document data is an important feature for clients to send long documents.

#### 1537 3.6.2 Warning About the Support of Chunked Requests

1538 This section describes some problems with the use of chunked requests and HTTP/1.1 servers.

1539 The HTTP/1.1 standard [HTTP] requires that conforming servers support chunked requests for any method.  
 1540 However, in spite of this requirement, some HTTP/1.1 implementations support chunked responses in the



1541 GET method, but do not support chunked POST method requests. Some HTTP/1.1 implementations that  
1542 support CGI scripts [CGI] and/or servlets [Servlet] require that the client supply a Content-Length. These  
1543 implementations might reject a chunked POST method and return a 411 status code (Length Required),  
1544 might attempt to buffer the request and run out of room returning a 413 status code (Request Entity Too  
1545 Large), or might successfully accept the chunked request.

1546 Because of this lack of conformance of HTTP servers to the HTTP/1.1 standard, the IPP standard [IPP-  
1547 PRO] REQUIRES that a conforming IPP Printer object implementation support chunked requests and that  
1548 conforming clients accept chunked responses. Therefore, IPP object implementers are warned to seek  
1549 HTTP server implementations that support chunked POST requests in order to conform to the IPP standard  
1550 and/or use implementation techniques that support chunked POST requests.

## 1551 4 References

1552 [CGI]

1553 CGI/1.1 (<http://www.ietf.org/internet-drafts/draft-coar-cgi-v11-00.txt>).

1554 [HTTP]

1555 HTTP/1.1 (<http://www.ietf.org/internet-drafts/draft-ietf-http-v11-spec-rev-06.txt>)

1556 [IPP LPD]

1557 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", draft-  
1558 ietf-ipp-lpd-ipp-map-04.txt, June 1998.

1559 [IPP-MOD]

1560 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and  
1561 Semantics", draft-ietf-ipp-model-11.txt, November, 1998.

1562 [IPP-PRO]

1563 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and  
1564 Transport", draft-ietf-ipp-protocol-07.txt, November, 1998.

1565 [IPP-RAT]

1566 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",  
1567 draft-ietf-ipp-rat-04.txt, November, 1998.

1568 [IPP-REQ]

1569 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-03.txt, November,  
1570 1998.

1571 [RFC1123]

1572 Braden, S., "Requirements for Internet Hosts - Application and Support", RFC 1123, October, 1989.

1573 [RFC2026]

1574 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

1575 [RFC2068]

1576 R Fielding, et al, "Hypertext Transfer Protocol – HTTP/1.1" RFC 2068, January 1997.

1577 [rfc2119]  
1578 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March  
1579 1997.

1580 [RFC2396]  
1581 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic  
1582 Syntax", RFC 2396, August 1998.

1583 [Servlet]  
1584 Servlet Specification Version 2.1 (<http://java.sun.com/products/servlet/2.1/index.html>).

1585 [SSL]  
1586 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.

1587

1588 4.1 Authors' Address

1589 Thomas N. Hastings  
1590 Xerox Corporation  
1591 701 Aviation Blvd.  
1592 El Segundo, CA 90245  
1593 [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)

1594  
1595 Carl-Uno Manros  
1596 Xerox Corporation  
1597 701 Aviation Blvd.  
1598 El Segundo, CA 90245  
1599 [manros@cp10.es.xerox.com](mailto:manros@cp10.es.xerox.com)

## 1600 5 Notices

1601 The IETF takes no position regarding the validity or scope of any intellectual property or other rights that  
1602 might be claimed to pertain to the implementation or use of the technology described in this document or  
1603 the extent to which any license under such rights might or might not be available; neither does it represent  
1604 that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to  
1605 rights in standards-track and standards-related documentation can be found in BCP-11[BCP-11]. Copies of  
1606 claims of rights made available for publication and any assurances of licenses to be made available, or the  
1607 result of an attempt made to obtain a general license or permission for the use of such proprietary rights by  
1608 implementers or users of this specification can be obtained from the IETF Secretariat.

1609 The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications,  
1610 or other proprietary rights which may cover technology that may be required to practice this standard.  
1611 Please address the information to the IETF Executive Director.

1612 Copyright (C) The Internet Society (1999). All Rights Reserved

1613 This document and translations of it may be copied and furnished to others, and derivative works that  
1614 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and  
1615 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice  
1616 and this paragraph are included on all such copies and derivative works. However, this document itself  
1617 may not be modified in any way, such as by removing the copyright notice or references to the Internet  
1618 Society or other Internet organizations, except as needed for the purpose of developing Internet standards  
1619 in which case the procedures for copyrights defined in the Internet Standards process must be followed, or  
1620 as required to translate it into languages other than English.

1621 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its  
1622 successors or assigns.

1623 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET  
1624 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,  
1625 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE  
1626 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED  
1627 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## 1628 **6 Change History**

1629 The change history is in reverse chronological order:

1630 6.1 Changes to produce the February 12, 1999 version from the January 8, 1999 version:

- 1631 1. Section 2.2.1.5: added check for document not found or accessible in Print-URI and Send-URI
- 1632 2. Section 3.6.2: Clarified that the IPP standard requires that servers **MUST** accept chunked requests  
1633 and that clients **MUST** accept chunked responses, in spite of the lack of conformance of HTTP  
1634 servers to the HTTP/1.1 requirement to support chunking.

1635 6.2 Changes to produce the January 8, 1999 version from the December 6, 1998 version:

- 1636 1. Added section 3.6.2: Warning About the Use of Chunked Requests with CGI Script  
1637 Implementations
- 1638 2. Section 2.2.1.2: changed "printer-operations-supported" to "operations-supported".
- 1639 3. Section 2.2.1.6: changed "job-media-supported" to "job-media-sheets-supported"
- 1640 4. Section 2.2.3: separated the validation checks for variable length attributes into two separate tests:  
1641 one for correct attribute syntax and one for correct length.
- 1642 5. Section 2.2.3: changed "multiple-document-handling-supported" to "printer-resolution-supported"
- 1643 6. Section 2.6.1: recommended that an IPP object also support US-ASCII charset.

1644 7. Section 3: Clarified that a server is not required to send a response until after it has received the  
1645 client's entire request, but recommend that the server return a response as soon as possible if an  
1646 error is detected while the client is still sending the data, rather than waiting until all of the data is  
1647 received. Also recommended that a client listen for an error response that an IPP server MAY send  
1648 before it receives all the data.

1649 6.3 Changes to produce the December 6, 1998 version from the November 16, 1998 version:

1650 Included all of the remaining agreed issues raised before the November 16, 1998 production of the Internet-  
1651 Drafts for IPP/1.0 that included adding explanations to the Implementers Guide.

1652