

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

T. Hastings  
Xerox Corporation  
C. Manros  
Xerox Corporation  
~~December 6~~ January 8, 1999

9 Internet Printing Protocol/1.0: Implementer's Guide

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

11 Status of this Memo

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

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

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

21

22 Abstract

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

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 .....	21
67	2.2.2	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add Documents.....	23
68	2.2.2.1	Default "ipp-attribute-fidelity" if not supplied.....	23
69	2.2.2.2	Check that the Printer object is accepting jobs .....	23
70	2.2.2.3	Validate the values of the Job Template attributes .....	23
71	2.2.3	Algorithm for job validation .....	24
72	2.2.3.1	Check for conflicting Job Template attributes values.....	28
73	2.2.3.2	Decide whether to REJECT the request.....	29
74	2.2.3.3	For the Validate-Job operation, RETURN one of the success status codes .....	29
75	2.2.3.4	Create the Job object with attributes to support.....	29
76	2.2.3.5	Return one of the success status codes.....	30
77	2.2.3.6	Accept appended Document Content.....	31
78	2.2.3.7	Scheduling and Starting to Process the Job .....	31
79	2.2.3.8	Completing the Job .....	31
80	2.2.3.9	Destroying the Job after completion .....	31
81	2.2.3.10	Interaction with "ipp-attribute-fidelity" .....	31
82	2.3	Status codes returned by operation (Issue 1.50) .....	32
83	2.3.1	Printer Operations .....	32
84	2.3.1.1	Print-Job.....	32
85	2.3.1.2	Print-URI.....	34
86	2.3.1.3	Validate-Job .....	34
87	2.3.1.4	Create-Job .....	34
88	2.3.1.5	Get-Printer-Attributes .....	34
89	2.3.1.6	Get-Jobs .....	35
90	2.3.2	Job Operations.....	35
91	2.3.2.1	Send-Document.....	35
92	2.3.2.2	Send-URI .....	36
93	2.3.2.3	Cancel-Job.....	36
94	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) .....	<del>42</del> 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).....	<del>43</del> 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)?.....	<del>44</del> 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.....	<del>46</del> 45
118	3.3	Response Headers .....	<del>47</del> 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 Use of Chunked Requests with CGI Script Implementations.....	<del>49</del> 48
124	3.6.2.1	Implications for IPP .....	<del>50</del> 49
125	4	References .....	50
126	4.1	Authors' Address .....	51
127	5	Appendix C: Full Copyright Statement.....	51
128	6	Change History.....	52
129	6.1	Changes to produce the January 8, 1999 version from the December 6, 1998 version:.....	52
130	6.2	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 "~~printer~~operations-supported" attribute. If not, the Printer REJECTS the  
214 request 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)

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

426 Operation Responses

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 any single non-empty 'charset' value, REJECT/RETURN 'client-error-request-bad-request'.  
528 IF the value length is greater than ~~less than or equal to~~ 63 octets, REJECT/RETURN 'client-error-  
529 request-value-too-long'.

530 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-charset-  
531 not-supported".  
532

533 attributes-natural-language(naturalLanguage)

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

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

538 ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-  
539 language-supported" attribute.  
540

541 requesting-user-name

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

543 IF the value length is greater than~~less than or equal to~~ 255 octets, REJECT/RETURN 'client-error-  
544 request-value-too-long'.  
545 IF the IPP object can obtain a better authenticated name, use it instead.  
546

547 job-name(name)  
548 IF NOT any single 'name' value, REJECT/RETURN 'client-error-request-bad-request'.  
549 IF the value length is greater than~~less than or equal to~~ 255 octets, REJECT/RETURN 'client-error-  
550 request-value-too-long'.  
551 IF NOT supplied by the client, the Printer object creates a name from the document-name or document-  
552 uri.  
553

554 document-name (name)  
555 IF NOT any single 'name' value, REJECT/RETURN 'client-error-request-bad-request'.  
556 IF the value length is greater than~~less than or equal to~~ 255 octets, REJECT/RETURN 'client-error-  
557 request-value-too-long'.  
558

559 ipp-attribute-fidelity (boolean)  
560 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-error-  
561 bad-request'.  
562 IF NOT supplied by the client, the IPP object assumes the value 'false'.  
563

564 document-format (mimeType)  
565 IF NOT any single non-empty 'mimeType' value, REJECT/RETURN 'client-error-request-bad-  
566 request'.  
567 IF the value length is greater than~~less than or equal to~~ 255 octets, REJECT/RETURN 'client-error-  
568 request-value-too-long'.  
569 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN 'client-error-  
570 document-format-not-supported'.  
571 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's "document-  
572 format-default" attribute.  
573

574 document-uri (uri)  
575 IF NOT any single non-empty 'uri' value, REJECT/RETURN 'client-error-request-bad-request'.  
576 IF the value length is greater than~~less than or equal to~~ 1023 octets, REJECT/RETURN 'client-error-  
577 request-value-too-long'.  
578 IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.  
579 IF scheme is NOT in the Printer object's "reference-uri-schemes-supported" attribute,  
580 REJECT/RETURN 'client-error'-uri-scheme-not-supported'.  
581

582 last-document (boolean)

583 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-error-  
584 bad-request'.  
585

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

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

589 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-error-  
590 gone' status code, if keep track of recently deleted jobs.  
591

592 requested-attributes (1setOf keyword)

593 IF NOT any number of 'keyword' values, REJECT/RETURN 'client-error-request-bad-request'.

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

596 Ignore unsupported values which are the keyword names of unsupported attributes. Don't bother to  
597 copy such requested (unsupported) attributes to the Unsupported Attribute response group since the  
598 response will not return them.  
599

600 which-jobs (type2 keyword)

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

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

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

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

609 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.  
610

611 my-jobs (boolean)

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

614 IF NOT supplied by the client, the IPP object assumes the 'false' value.  
615

616 limit (integer(1:MAX))

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

619 IF NOT supplied by the client, the IPP object returns all jobs, no matter how many.  
620

621 -----  
622

623 2.2.1.6 Validate the values of the OPTIONAL Operation attributes

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

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

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

636 -----

637 document-natural-language (naturalLanguage)

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

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

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

644

645 compression (type3 keyword)

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

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

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

652

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

654 IF NOT any single 'integer' value equal to 4 octets,  
655 REJECT/RETURN 'client-error-bad-request'.

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

659

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

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

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

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

674 message (text(127))  
675 IF NOT any single 'text' value, REJECT/RETURN 'client-error-request-bad-request'.  
676 IF the value length is greater than~~less than or equal to~~ 127 octets,  
677 REJECT/RETURN 'client-error-request-value-too-long'.  
678

679 unknown or unsupported attribute  
680 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute  
681 syntax, REJECT/RETURN 'client-error-request-value-too-long'.  
682 ELSE copy the attribute and value to the Unsupported Attributes response group and change the  
683 attribute value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.  
684

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

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

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

705 2.2.2.1 Default "ipp-attribute-fidelity" if not supplied

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

708 2.2.2.2 Check that the Printer object is accepting jobs

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

711 2.2.2.3 Validate the values of the Job Template attributes

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

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

717 b) that the attribute syntax tag is correct for that attribute according to [IPP-MOD] Sections 4.2 to  
718 4.4.

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

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

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

- 730 1. reject the operation and return the 'client-error-bad syntax' error status code
- 731 2. accept the operation and use the first occurrence of the attribute



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

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

### 735 2.2.3 Algorithm for job validation

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

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

740 1. If U is multi-valued, validate each value X of U by performing the algorithm in [Table 2](#)~~Table-2~~  
741 with each value X. Each validation is separate from the standpoint of returning unsupported  
742 values.

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

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

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

755 3. If both U and V are single-valued, let X be U and Z be V and use the validation rules in [Table](#)  
756 ~~2~~[Table-2](#).

757 **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.

758

759 If the value of the Printer object's "xxx-supported" attribute is 'no-value' (because the system administrator  
760 hasn't configured a value), the check always fails. If the check fails, the IPP object copies the attribute to



761 the Unsupported Attributes response group with its unsupported value. If the attribute contains more than  
762 one value, each value is checked and each unsupported value is separately copied, while supported values  
763 are not copied. If an IPP object doesn't recognize/support a Job Template attribute, i.e., there is no  
764 corresponding Printer object "xxx-supported" attribute, the IPP object treats the attribute as an unknown or  
765 unsupported attribute (see the last row in the table below).

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

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

776 -----

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

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

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

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

784 Map the value to the nearest supported value in the range 1:100 as specified by the number of discrete  
785 values indicated by the value of the Printer's "job-priority-supported" attribute. See the formula in  
786 [IPP-MOD] Section 4.2.1.  
787

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

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

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

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

794 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the  
795 unsupported value to the Unsupported Attributes response group.  
796

797 job-sheets (type3 keyword | name)

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

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

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

804 multiple-document-handling (type2 keyword)

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

811 copies (integer(1:MAX))

812 IF NOT a single 'integer' value with a length equal to 4 octets,  
813 REJECT/RETURN 'client-error-bad-request'.  
814 IF NOT in range of the Printer object's "copies-supported" attribute  
815 copy the attribute and the unsupported value to the Unsupported Attributes response group.  
816

817 finishings (1setOf type2 enum)

818 IF NOT an 'enum' value(s) each with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-  
819 request'.  
820 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the unsupported  
821 value(s), but not any supported values, to the Unsupported Attributes response group.  
822

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

824 IF NOT a 'rangeOfInteger' value(s) each with a length equal to 8 octets, REJECT/RETURN 'client-  
825 error-bad-request'.  
826 IF first value is greater than second value in any range, the ranges are not in ascending order, or ranges  
827 overlap, REJECT/RETURN 'client-error-bad-request'.  
828 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to the  
829 Unsupported Attributes response group and set the value to the "out-of-band" 'unsupported' value.  
830

831 sides (type2 keyword)

832 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-request-bad-request'.  
833 IF the value length is greater than~~with a length less than or equal to~~ 255 octets, REJECT/RETURN  
834 'client-error-request-value-too-long'.  
835 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported value  
836 to the Unsupported Attributes response group.  
837

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

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

841 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"  
842 attribute, copy the attribute and value to the Unsupported Attribute response group.  
843

844 orientation-requested (type2 enum)

845 IF NOT a single 'enum' value with a length equal to 4 octets,  
846 REJECT/RETURN 'client-error-bad-request'.  
847 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the  
848 unsupported value to the Unsupported Attributes response group.  
849

850 media (type3 keyword | name)

851 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-request-bad-request'.  
852 IF the with a value length less than or equal to is greater than 255 octets, REJECT/RETURN 'client-  
853 error-request-value-too-long'.  
854 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported value  
855 to the Unsupported Attributes response group.  
856

857 printer-resolution (resolution)

858 IF NOT a single 'resolution' value with a length equal to 9 octets,  
859 REJECT/RETURN 'client-error-bad-request'.  
860 IF NOT in the Printer object's "printer-resolution-supported~~multiple-document-handling-supported~~"  
861 attribute, copy the attribute and the unsupported value to the Unsupported Attributes response  
862 group.  
863

864 print-quality (type2 enum)

865 IF NOT a single 'enum' value with a length equal to 4 octets,  
866 REJECT/RETURN 'client-error-bad-request'.  
867 IF NOT in the Printer object's "print-quality-supported" attribute, copy the attribute and the  
868 unsupported value to the Unsupported Attributes response group.  
869

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

871 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute  
872 syntax,  
873 REJECT/RETURN 'client-error-bad-request' if the length of the attribute syntax is fixed or 'client-error-  
874 request-value-too-long' if the length of the attribute syntax is variable.  
875 ELSE copy the attribute and value to the Unsupported Attributes response group and change the  
876 attribute value to the "out-of-band" 'unsupported' value. Any remaining Job Template Attributes are  
877 either unknown or unsupported Job Template attributes and are validated algorithmically according  
878 to their attribute syntax for proper length (see below).

879 -----

880

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

887	Name	Octet length check for read-write attributes
888	-----	-----
889	'textWithLanguage	<= 1023 AND 'naturalLanguage' <= 63
890	'textWithoutLanguage'	<= 1023
891	'nameWithLanguage'	<= 255 AND 'naturalLanguage' <= 63
892	'nameWithoutLanguage'	<= 255
893	'keyword'	<= 255
894	'enum'	= 4
895	'uri'	<= 1023
896	'uriScheme'	<= 63
897	'charset'	<= 63
898	'naturalLanguage'	<= 63
899	'mimeType'	<= 255
900	'octetString'	<= 1023
901	'boolean'	= 1
902	'integer'	= 4
903	'rangeOfInteger'	= 8
904	'dateTime'	= 11
905	'resolution'	= 9
906	'lsetOf X'	
907		

### 908 2.2.3.1 Check for conflicting Job Template attributes values

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

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

## 922 2.2.3.2 Decide whether to REJECT the request

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

- 926 (1) 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes  
927 supplied by the client.
- 928 (2) 'client-error-attributes-or-values-not-supported' status code, otherwise.  
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 2.2.3.3 For the Validate-Job operation, RETURN one of the success status codes

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

- 936 (1) the "successful-ok" status code, if there are no unsupported or conflicting Job Template attributes or  
937 values.
- 938 (2) the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or  
939 values.
- 940 (3) the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template  
941 attributes or values.

942

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

## 947 2.2.3.4 Create the Job object with attributes to support

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

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

960

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

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

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

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

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

#### 992 2.2.3.5 Return one of the success status codes

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

- 994 (1) the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes or  
995 values.
- 996 (2) the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template  
997 attribute or values.
- 998 (3) the 'successful-ok-ignored-or-substituted-attributes' status code, if there are only unsupported Job  
999 Template attributes or values.



1000

1001 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this  
1002 step. If the unsupported Operation attribute was a serious error, the above already rejected the request in a  
1003 previous step. If control gets to this step with unsupported Operation attributes being returned, they are not  
1004 serious errors.

1005 The Printer object also returns Job status attributes that indicate the initial state of the Job ('pending',  
1006 'pending-held', 'processing', etc.), etc. See Print-Job Response, [IPP-MOD] section 3.2.1.2.

#### 1007 2.2.3.6 Accept appended Document Content

1008 The Printer object accepts the appended Document Content data and either starts it printing, or spools it for  
1009 later processing.

#### 1010 2.2.3.7 Scheduling and Starting to Process the Job

1011 The Printer object uses its own configuration and implementation specific algorithms for scheduling the  
1012 Job in the correct processing order. Once the Printer object begins processing the Job, the Printer changes  
1013 the Job's state to 'processing'. If the Printer object supports PDL override (the "pdl-override-supported"  
1014 attribute set to 'attempted'), the implementation does its best to see that IPP attributes take precedence over  
1015 embedded instructions in the document data.

#### 1016 2.2.3.8 Completing the Job

1017 The Printer object continues to process the Job until it can move the Job into the 'completed' state. If an  
1018 Cancel-Job operation is received, the implementation eventually moves the Job into the 'canceled' state. If  
1019 the system encounters errors during processing that do not allow it to progress the Job into a completed  
1020 state, the implementation halts all processing, cleans up any resources, and moves the Job into the 'aborted'  
1021 state.

#### 1022 2.2.3.9 Destroying the Job after completion

1023 Once the Job moves to the 'completed', 'aborted', or 'canceled' state, it is an implementation decision as to  
1024 when to destroy the Job object and release all associated resources. Once the Job has been destroyed, the  
1025 Printer would return either the "client-error-not-found" or "client-error-gone" status codes for operations  
1026 directed at that Job.

1027 Note: the Printer object SHOULD NOT re-use a "job-uri" or "job-id" value for a sufficiently long time  
1028 after a job has been destroyed, so that stale references kept by clients are less likely to access the wrong  
1029 (newer) job.

#### 1030 2.2.3.10 Interaction with "ipp-attribute-fidelity"

1031 Some Printer object implementations may support "ipp-attribute-fidelity" set to 'true' and "pdl-override-  
1032 supported" set to 'attempted' and yet still not be able to realize exactly what the client specifies in the create

1033 request. This is due to legacy decisions and assumptions that have been made about the role of job  
1034 instructions embedded within the document data and external job instructions that accompany the  
1035 document data and how to handle conflicts between such instructions. The inability to be 100% precise  
1036 about how a given implementation will behave is also compounded by the fact that the two special  
1037 attributes, "ipp-attribute-fidelity" and "pdl-override-supported", apply to the whole job rather than specific  
1038 values for each attribute. For example, some implementations may be able to override almost all Job  
1039 Template attributes except for "number-up".

## 1040 2.3 Status codes returned by operation (Issue 1.50)

1041 This section lists all status codes once in the first operation (Print-Job). Then it lists the status codes that  
1042 are different or specialized for subsequent operations under each operation.

### 1043 2.3.1 Printer Operations

#### 1044 2.3.1.1 Print-Job

1045 The Printer object **MUST** return one of the following "status-code" values for the indicated reason.  
1046 Whether all of the document data has been accepted or not before returning the success or error response  
1047 depends on implementation. See Section 14 for a more complete description of each status code.

1048 For the following success status codes, the Job object has been created and the "job-id", and "job-uri"  
1049 assigned and returned in the response:

1050     successful-ok: no request attributes were substituted or ignored.

1051     successful-ok-ignored-or-substituted-attributes: some supplied (1) attributes were ignored or (2)  
1052     unsupported attribute syntaxes or values were substituted with supported values or were ignored.  
1053     Unsupported attributes, attribute syntaxes, or values **MUST** be returned in the Unsupported  
1054     Attributes group of the response.

1055     successful-ok-conflicting-attributes: some supplied attribute values conflicted with the values of other  
1056     supplied attributes and were either substituted or ignored. Attributes or values which conflict with  
1057     other attributes and have been substituted or ignored **MUST** be returned in the Unsupported  
1058     Attributes group of the response as supplied by the client.  
1059

1060 [ipp-mod] section 3.1.6 Operation Status Codes and Messages states (Issue 1.19):

1061     If the Printer object supports the "status-message" operation attribute, it **SHOULD** use the  
1062     **REQUIRED** 'utf-8' charset to return a status message for the following error status codes (see  
1063     section 14): 'client-error-bad-request', 'client-error-charset-not-supported', 'server-error-internal-  
1064     error', 'server-error-operation-not-supported', and 'server-error-version-not-supported'. In this case,  
1065     it **MUST** set the value of the "attributes-charset" operation attribute to 'utf-8' in the error response.

1066 For the following error status codes, no job is created and no "job-id" or "job-uri" is returned:

1067     client-error-bad-request: The request syntax does not conform to the specification.



1068 client-error-forbidden: The request is being refused for authorization or authentication reasons. The  
1069 implementation security policy is to not reveal whether the failure is one of authentication or  
1070 authorization.

1071 client-error-not-authenticated: Either the request requires authentication information to be supplied or  
1072 the authentication information is not sufficient for authorization.

1073 client-error-not-authorized: The requester is not authorized to perform the request on the target object.

1074 client-error-not-possible: The request cannot be carried out because of the state of the system. See also  
1075 'server-error-not-accepting-jobs' status code which MUST take precedence if the Printer object's  
1076 "printer-accepting-jobs" attribute is 'false'.

1077 client-error-timeout: not applicable.

1078 client-error-not-found: the target object does not exist.

1079 client-error-gone: the target object no longer exists and no forwarding address is known.

1080 client-error-request-entity-too-large: the size of the request and/or print data exceeds the capacity of the  
1081 IPP Printer to process it.

1082 client-error-request-value-too-long: the size of request variable length attribute values, such as 'text'  
1083 and 'name' attribute syntaxes, exceed the maximum length specified in [IPP-MOD] for the attribute  
1084 and MUST be returned in the Unsupported Attributes Group.

1085 client-error-document-format-not-supported: the document format supplied is not supported. The  
1086 "document-format" attribute with the unsupported value MUST be returned in the Unsupported  
1087 Attributes Group. This error SHOULD take precedence over any other 'xxx-not-supported' error,  
1088 except 'client-error-charset-not-supported'.

1089 client-error-attributes-or-values-not-supported: one or more supplied attributes, attribute syntaxes, or  
1090 values are not supported and the client supplied the "ipp-attributes-fidelity" operation attribute with  
1091 a 'true' value. They MUST be returned in the Unsupported Attributes Group as explained below.

1092 client-error-uri-scheme-not-supported: not applicable.

1093 client-error-charset-not-supported: the charset supplied in the "attributes-charset" operation attribute is  
1094 not supported. The Printer's "configured-charset" MUST be returned in the response as the value of  
1095 the "attributes-charset" operation attribute and used for any 'text' and 'name' attributes returned in  
1096 the error response. This error SHOULD take precedence over any other error, unless the request  
1097 syntax is so bad that the client's supplied "attributes-charset" cannot be determined.

1098 client-error-conflicting-attributes: one or more supplied attribute values conflicted with each other and  
1099 the client supplied the "ipp-attributes-fidelity" operation attribute with a 'true' value. They MUST  
1100 be returned in the Unsupported Attributes Group as explained below.

1101 server-error-internal-error: an unexpected condition prevents the request from being fulfilled.

1102 server-error-operation-not-supported: not applicable (since Print-Job is REQUIRED).

1103 server-error-service-unavailable: the service is temporarily overloaded.

1104 server-error-version-not-supported: the version in the request is not supported. The "closest" version  
1105 number supported MUST be returned in the response.

1106 server-error-device-error: a device error occurred while receiving or spooling the request or document  
1107 data or the IPP Printer object can only accept one job at a time.

1108 server-error-temporary-error: a temporary error such as a buffer full write error, a memory overflow, or  
1109 a disk full condition occurred while receiving the request and/or the document data.

1110 server-error-not-accepting-jobs: the Printer object's "printer-is-not-accepting-jobs" attribute is 'false'.

1111 server-error-busy: the Printer is too busy processing jobs to accept another job at this time.

1112 server-error-job-canceled: the job has been canceled by an operator or the system while the client was  
1113 transmitting the document data.

## 1114 2.3.1.2 Print-URI

1115 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Print-  
1116 URI with the following specializations and differences. See Section 14 for a more complete description of  
1117 each status code.

1118       server-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation  
1119       attribute is not supported and is returned in the Unsupported Attributes group.

## 1120 2.3.1.3 Validate-Job

1121 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Validate-  
1122 Job. See Section 14 for a more complete description of each status code.

## 1123 2.3.1.4 Create-Job

1124 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Create-  
1125 Job with the following specializations and differences. See Section 14 for a more complete description of  
1126 each status code.

1127       server-error-operation-not-supported: the Create-Job operation is not supported.

## 1128 2.3.1.5 Get-Printer-Attributes

1129 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-  
1130 Printer-Attributes operation with the following specializations and differences. See Section 14 for a more  
1131 complete description of each status code.

1132 For the following success status codes, the requested attributes are returned in Group 3 in the response:

1133       successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested  
1134       attributes were unsupported.

1135       successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"  
1136       operation attribute MAY, but NEED NOT, be returned with the unsupported values.

1137       successful-ok-conflicting-attributes: same as Print-Job.

1138 For the error status codes, Group 3 is returned containing no attributes or is not returned at all:

1139       client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.

1140       client-error-request-entity-too-large: same as Print-job, except that no print data is involved.

1141       client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes  
1142       MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.

1143       client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.

1144       server-error-operation-not-supported: not applicable (since Get-Printer-Attributes is REQUIRED).

1145       server-error-device-error: same as Print-Job, except that no document data is involved.

1146       server-error-temporary-error: same as Print-Job, except that no document data is involved.

1147       server-error-not-accepting-jobs: not applicable..

1148       server-error-busy: same as Print-Job, except the IPP object is too busy to accept even query requests.

1149       server-error-job-canceled: not applicable..

## 1150 2.3.1.6 Get-Jobs

1151 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-  
1152 Jobs operation with the following specializations and differences. See Section 14 for a more complete  
1153 description of each status code.

1154 For the following success status codes, the requested attributes are returned in Group 3 in the response:

1155       successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested  
1156       attributes were unsupported.

1157       successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"  
1158       operation attribute MAY, but NEED NOT, be returned with the unsupported values.

1159       successful-ok-conflicting-attributes: same as Print-Job.

1160 For any error status codes, Group 3 is returned containing no attributes or is not returned at all. The  
1161 following brief error status code descriptions contain unique information for use with Get-Jobs operation.  
1162 See section 14 for the other error status codes that apply uniformly to all operations:

1163       client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.

1164       client-error-request-entity-too-large: same as Print-job, except that no print data is involved.

1165       client-error-document-format-not-supported: not applicable.

1166       client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes  
1167       MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.

1168       client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.

1169       server-error-operation-not-supported: not applicable (since Get-Jobs is REQUIRED).

1170       server-error-device-error: same as Print-Job, except that no document data is involved.

1171       server-error-temporary-error: same as Print-Job, except that no document data is involved.

1172       server-error-not-accepting-jobs: not applicable.

1173       server-error-job-canceled: not applicable.

## 1174 2.3.2 Job Operations

## 1175 2.3.2.1 Send-Document

1176 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to the Get-  
1177 Printer-Attributes operation with the following specializations and differences. See Section 14 for a more  
1178 complete description of each status code.

1179 For the following success status codes, the document has been added to the specified Job object and the  
1180 job's "number-of-documents" attribute has been incremented:

1181       successful-ok: no request attributes were substituted or ignored (same as Print-Job).

1182       successful-ok-ignored-or-substituted-attributes: same as Print-Job.

1183       successful-ok-conflicting-attributes: same as Print-Job.

1184 For the error status codes, no document has been added to the Job object and the job's "number-of-  
1185 documents" attribute has not been incremented:

1186       client-error-not-possible: Same as Print-Job, except that the Printer's "printer-is-accepting-jobs"  
1187       attribute is not involved, so that the client is able to finish submitting a multi-document job after this

1188 attribute has been set to 'true'. Another condition is that the state of the job precludes Send-  
1189 Document, i.e., the job has already been closed out by the client. However, if the IPP Printer closed  
1190 out the job due to timeout, the 'client-error-timeout' error status SHOULD be returned instead.  
1191 client-error-timeout: This request was sent after the Printer closed the job, because it has not received a  
1192 Send-Document or Send-URI operation within the Printer's "multiple-operation-time-out" period .  
1193 client-error-request-entity-too-large: same as Print-Job.  
1194 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attributes-fidelity" operation  
1195 attribute is not involved..  
1196 server-error-operation-not-supported: the Send-Document request is not supported.  
1197 server-error-not-accepting-jobs: not applicable.  
1198 server-error-job-canceled: the job has been canceled by an operator or the system while the client was  
1199 transmitting the data.

### 1200 2.3.2.2 Send-URI

1201 All of the Print-Job status code descriptions in Section 3.2.1.2 Print-Job Response with the specializations  
1202 described for Send-Document are applicable to Send-URI. See Section 14 for a more complete description  
1203 of each status code.

1204 server-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation  
1205 attribute is not supported and the "document-uri" attribute MUST be returned in the Unsupported  
1206 Attributes group.

### 1207 2.3.2.3 Cancel-Job

1208 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Cancel-  
1209 Job with the following specializations and differences. See Section 14 for a more complete description of  
1210 each status code.

1211 For the following success status codes, the Job object is being canceled or has been canceled:

1212 successful-ok: no request attributes were substituted or ignored (same as Print-Job).  
1213 successful-ok-ignored-or-substituted-attributes: same as Print-Job.  
1214 successful-ok-conflicting-attributes: same as Print-Job.  
1215

1216 For any of the error status codes, the Job object has not been canceled or was previously canceled.

1217 client-error-not-possible: The request cannot be carried out because of the state of the Job object  
1218 ('completed', 'canceled', or 'aborted') or the state of the system.  
1219 client-error-not-found: the target Printer and/or Job object does not exist.  
1220 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding address is  
1221 known.  
1222 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.  
1223 client-error-document-format-not-supported: not applicable.  
1224 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes  
1225 and values MUST be ignored.  
1226 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-  
1227 jobs" attribute is not involved.

- 1228 server-error-operation-not-supported: not applicable (Cancel-Job is REQUIRED).
- 1229 server-error-device-error: same as Print-Job, except no document data is involved.
- 1230 server-error-temporary-error: same as Print-Job, except no document data is involved.
- 1231 server-error-not-accepting-jobs: not applicable..
- 1232 server-error-job-canceled: not applicable.
- 1233 2.3.2.4 Get-Job-Attributes
- 1234 All of the Print-Job status codes described in Section 3.2.1.2 Print-Job Response are applicable to Get-Job-  
1235 Attributes with the following specializations and differences. See Section 14 for a more complete  
1236 description of each status code.
- 1237 For the following success status codes, the requested attributes are returned in Group 3 in the response:
- 1238 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested  
1239 attributes were unsupported.
- 1240 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"  
1241 operation attribute MAY, but NEED NOT, be returned with the unsupported values.
- 1242 successful-ok-conflicting-attributes: same as Print-Job.
- 1243 For the error status codes, Group 3 is returned containing no attributes or is not returned at all.
- 1244 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.
- 1245 client-error-document-format-not-supported: not applicable.
- 1246 client-error-attributes-or-values-not-supported: not applicable.
- 1247 client-error-uri-scheme-not-supported: not applicable.
- 1248 client-error-conflicting-attributes: not applicable
- 1249 server-error-operation-not-supported: not applicable (since Get-Job-Attributes is REQUIRED).
- 1250 server-error-device-error: same as Print-Job, except no document data is involved.
- 1251 server-error-temporary-error: sane as Print-Job, except no document data is involved..
- 1252 server-error-not-accepting-jobs: not applicable.
- 1253 server-error-job-canceled: not applicable.
- 1254 2.4 Validate-Job
- 1255 The Validate-Job operation has been designed so that its implementation may be a part of the Print-Job  
1256 operation. Therefore, requiring Validate-Job is not a burden on implementers. Also it is useful for client's  
1257 to be able to count on its presence in all conformance implementations, so that the client can determine  
1258 before sending a long document, whether the job will be accepted by the IPP Printer or not.
- 1259 2.5 Case Sensitivity in URIs (issue 1.6)
- 1260 IPP client and server implementations must be aware of the diverse uppercase/lowercase nature of URIs.  
1261 RFC 2396 defines URL schemes and Host names as case insensitive but reminds us that the rest of the  
1262 URL may well demonstrate case sensitivity. When creating URL's for fields where the choice is  
1263 completely arbitrary, it is probably best to select lower case. However, this cannot be guaranteed and

1264 implementations MUST NOT rely on any fields being case-sensitive or case-insensitive in the URL beyond  
1265 the URL scheme and host name fields.

1266 The reason that the IPP specification does not make any restrictions on URIs, is so that implementations of  
1267 IPP may use off-the-shelf components that conform to the standards that define URIs, such as RFC 2396  
1268 and the HTTP/1.1 specifications [RFC2068]. See these specifications for rules of matching, comparison,  
1269 and case-sensitivity.

1270 It is also recommended that that System Administrators and implementations avoid creating URLs for  
1271 different printers that differ only in their case. For example, don't have Printer1 and printer1 as two  
1272 different IPP Printers.

1273 The HTTP/1.1 specification [RFC2068] contains more details on comparing URLs.

1274 2.6 Character Sets, natural languages, and internationalization

1275 This section discusses character set support, natural language support and internationalization.

1276 2.6.1 Character set code conversion support (Issue 1.5)

1277 IPP clients and IPP objects are REQUIRED to support UTF-8. They MAY support additional charsets. It  
1278 is RECOMMENDED that an IPP object also support US-ASCII, since many clients support US-ASCII,  
1279 and indicate that UTF-8 and US-ASCII are supported by populating the Printer's "charset-supported" with  
1280 'utf-8' and 'us-ascii' values. An IPP object is required to code covert with as little loss as possible between  
1281 the charsets that it supports, as indicated in the Printer's "charsets-supported" attribute.

1282 How should the server handle the situation where the "attributes-charset" of the response itself is "us-ascii",  
1283 but one or more attributes in that response is in the "utf-8" format?

1284 Example: Consider a case where a client sends a Print-Job request with "utf-8" as the value of "attributes-  
1285 charset" and with the "job-name" attribute supplied. Later another client submits a Get-Job-Attribute or  
1286 Get-Jobs request. This second request contains the "attributes-charset" with value "us-ascii" and  
1287 "requested-attributes" attribute with exactly one value "job-name".

1288 According to the IPP-Mod document (section 3.1.4.2), the value of the "attributes-charset" for the response  
1289 of the second request must be "us-ascii" since that is the charset specified in the request. The "job-name"  
1290 value, however, is in "utf-8" format. Should the request be rejected even though both "utf-8" and "us-ascii"  
1291 charsets are supported by the server? or should the "job-name" value be converted to "us-ascii" and return  
1292 "successful-ok-conflicting-attributes" (0x0002) as the status code?

1293 Answer: An IPP object that supports both utf-8 (REQUIRED) and us-ascii, the second paragraph of  
1294 section 3.1.4.2 applies so that the IPP object MUST accept the request, perform code set conversion  
1295 between these two charsets with "the highest fidelity possible" and return 'successful-ok', rather than a  
1296 warning 'successful-ok-conflicting-attributes, or an error. The printer will do the best it can to convert  
1297 between each of the character sets that it supports--even if that means providing a string of question marks  
1298 because none of the characters are representable in US ASCII. If it can't perform such conversion, it



1299 MUST NOT advertise us-ascii as a value of its "attributes-charset-supported" and MUST reject any request  
1300 that requests 'us-ascii'.

1301 One IPP object implementation strategy is to convert all request text and name values to a Unicode internal  
1302 representation. This is 16-bit and virtually universal. Then convert to the specified operation attributes-  
1303 charset on output.

1304 Also it would be smarter for a client to ask for 'utf-8', rather than 'us-ascii' and throw away characters that it  
1305 doesn't understand, rather than depending on the code conversion of the IPP object.

1306 2.6.2 What charset to return when an unsupported charset is requested (Issue 1.19)?

1307 Section 3.1.4.1 Request Operation attributes was clarified in November 1998 as follows:

1308 All clients and IPP objects MUST support the 'utf-8' charset [RFC2044] and MAY support  
1309 additional charsets provided that they are registered with IANA [IANA-CS]. If the Printer object  
1310 does not support the client supplied charset value, the Printer object MUST reject the request, set  
1311 the "attributes-charset" to 'utf-8' in the response, and return the 'client-error-charset-not-supported'  
1312 status code and any 'text' or 'name' attributes using the 'utf-8' charset.

1313 Since the client and IPP object MUST support UTF-8, returning any text or name attributes in UTF-8 when  
1314 the client requests a charset that is not supported should allow the client to display the text or name.

1315 Since such an error is a client error, rather than a user error, the client should check the status code first so  
1316 that it can avoid displaying any other returned 'text' and 'name' attributes that are not in the charset  
1317 requested.

1318 Furthermore, [ipp-mod] section 14.1.4.14 client-error-charset-not-supported (0x040D) was clarified in  
1319 November 1998 as follows:

1320 For any operation, if the IPP Printer does not support the charset supplied by the client in the  
1321 "attributes-charset" operation attribute, the Printer MUST reject the operation and return this status  
1322 and any 'text' or 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1).

1323 2.6.3 Natural Language Override (NLO) (Issue 1.45)

1324 The 'text' and 'name' attributes each have two forms. One has an implicit natural language, and the other  
1325 has an explicit natural language. The 'textWithoutLanguage' and 'textWithLanguage' are the two 'text'  
1326 forms. The 'nameWithoutLanguage' and 'nameWithLanguage' are the two 'name' forms. If a receiver (IPP  
1327 object or IPP client) supports an attribute with attribute syntax 'text', it MUST support both forms in a  
1328 request and a response. A sender (IPP client or IPP object) MAY send either form for any such attribute.  
1329 When a sender sends a WithoutLanguage form, the implicit natural language is specified in the "attributes-  
1330 natural-language" operation attribute which all senders MUST include in every request and response.

1331 When a sender sends a WithLanguage form, it MAY be different from the implicit natural language  
1332 supplied by the sender or it MAY be the same. The receiver MUST treat either form equivalently.

1333 There is an implementation decision for senders, whether to always send the WithLanguage forms or use  
1334 the WithoutLanguage form when the attribute's natural language is the same as the request or response.  
1335 The former approach makes the sender implementation simpler. The latter approach is more efficient on  
1336 the wire and allows inter-working with non-conforming receivers that fail to support the WithLanguage  
1337 forms. As each approach have advantages, the choice is completely up to the implementer of the sender.

1338 Furthermore, when a client receives a 'text' or 'name' job attribute that it had previously supplied, that client  
1339 MUST NOT expect to see the attribute in the same form, i.e., in the same WithoutLanguage or  
1340 WithLanguage form as the client supplied when it created the job. The IPP object is free to transform the  
1341 attribute from the WithLanguage form to the WithoutLanguage form and vice versa, as long as the natural  
1342 language is preserved. However, in order to meet this latter requirement, it is usually simpler for the IPP  
1343 object implementation to store the natural language explicitly with the attribute value, i.e., to store using an  
1344 internal representation that resembles the WithLanguage form.

1345 The IPP Printer MUST copy the natural language of a job, i.e., the value of the "attributes-natural-  
1346 language" operation attribute supplied by the client in the create operation, to the Job object as a Job  
1347 Description attribute, so that a client is able to query it. In returning a Get-Job-Attributes response, the IPP  
1348 object MAY return one of three natural language values in the response's "attributes-natural-language"  
1349 operation attribute: (1) that requested by the requester, (2) the natural language of the job, or (3) the  
1350 configured natural language of the IPP Printer, if the requested language is not supported by the IPP  
1351 Printer.

1352 This "attributes-natural-language" Job Description attribute is useful for an IPP object implementation that  
1353 prints start sheets in the language of the user who submitted the job. This same Job Description attribute is  
1354 useful to a multi-lingual operator who has to communicate with different job submitters in different natural  
1355 languages. This same Job Description attribute is expected to be used in the future to generate notification  
1356 messages in the natural language of the job submitter.

1357 Early drafts of [IPP-MOD] contained a job-level natural language override (NLO) for the Get-Jobs  
1358 response. A job-level (NLO) is an (unrequested) Job Attribute which then specified the implicit natural  
1359 language for any other WithoutLanguage job attributes returned in the response for that job.  
1360 Interoperability testing of early implementations showed that no one was implementing the job-level NLO  
1361 in Get-Job responses. So the job-level NLO was eliminated from the Get-Jobs response. This  
1362 simplification makes all requests and responses consistent in that the implicit natural language for any  
1363 WithoutLanguage 'text' or 'name' form is always supplied in the request's or response's "attributes-natural-  
1364 language" operation attribute.

1365 2.7 The "queued-job-count" Printer Description attribute

1366 2.7.1 Why is "queued-job-count" RECOMMENDED (Issue 1.14)?

1367 The reason that "queued-job-count" is RECOMMENDED, is that some clients look at that attribute alone  
1368 when summarizing the status of a list of printers, instead of doing a Get-Jobs to determine the number of  
1369 jobs in the queue. Implementations that fail to support the "queued-job-count" will cause that client to  
1370 display 0 jobs when there are actually queued jobs.



1371 We would have made it a REQUIRED Printer attribute, but some implementations had already been  
1372 completed before the issue was raised, so making it a SHOULD was a compromise.

1373 2.7.2 Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?

1374 The "queued-job-count" is not a good measure of how busy the printer is when there are held jobs. A  
1375 future registration could be to add a "held-job-count" (or an "active-job-count") Printer Description  
1376 attribute if experience shows that such an attribute (combination) is needed to quickly indicate how busy a  
1377 printer really is.

1378 2.8 Sending empty attribute groups (Issue 1.16)

1379 The [IPP-MOD] and [IPP-PRO] specifications RECOMMEND that a sender not send an empty attribute  
1380 group in a request or a response. However, they REQUIRE a receiver to accept an empty attribute group as  
1381 equivalent to the omission of that group. So a client SHOULD omit the Job Template Attributes group  
1382 entirely in a create operation that is not supplying any Job Template attributes. Similarly, an IPP object  
1383 SHOULD omit an empty Unsupported Attributes group if there are no unsupported attributes to be returned  
1384 in a response.

1385 The [IPP-PRO] specification REQUIRES a receiver to be able to receive either an empty attribute group or  
1386 an omitted attribute group and treat them equivalently. The term "receiver" means an IPP object for a  
1387 request and a client for a response. The term "sender" means a client for a request and an IPP object for a  
1388 response.

1389 There is an exception to the rule for Get-Jobs when there are no attributes to be returned. [ipp-pro]  
1390 contains the following paragraph:

1391       The syntax allows an xxx-attributes-tag to be present when the xxx-attribute-sequence that follows  
1392       is empty. The syntax is defined this way to allow for the response of Get-Jobs where no attributes  
1393       are returned for some job-objects. Although it is RECOMMENDED that the sender not send an  
1394       xxx-attributes-tag if there are no attributes (except in the Get-Jobs response just mentioned), the  
1395       receiver MUST be able to decode such syntax.

1396 2.9 Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)

1397 In the Get-Printer-Attributes, Get-Jobs, or Get-Job-Attributes responses, the client cannot depend on getting  
1398 unsupported attributes returned in the Unsupported Attributes group that the client requested, but are not  
1399 supported by the IPP object. However, such unsupported requested attributes will not be returned in the  
1400 Job Attributes or Printer Attributes group (since they are unsupported). Furthermore, the IPP object is  
1401 REQUIRED to return the 'successful-ok-ignored-or-substituted-attributes' status code, so that the client  
1402 knows that not all that was requested has been returned.

## 1403 2.10 Returning job-state in Print-Job response (Issue 1.30)

1404 An IPP client submits a small job via Print-Job. By the time the IPP printer/print server is putting together  
1405 a response to the operation, the job has finished printing and been removed as an object from the print  
1406 system. What should the job-state be in the response?

1407 The Model suggests that the Printer return a response before it even accepts the document content. The Job  
1408 Object Attributes are returned only if the IPP object returns one of the success status codes. Then the job-  
1409 state would always be "pending" or "pending-held".

1410 This issue comes up for the implementation of an IPP Printer object as a server that forwards jobs to  
1411 devices that do not provide job status back to the server. If the server is reasonably certain that the job  
1412 completed successfully, then it should return the job-state as 'completed'. Also the server can keep the job  
1413 in its "job history" long after the job is no longer in the device. Then a user could query the server and see  
1414 that the job was in the 'completed' state and completed as specified by the job's "time-at-completed" time  
1415 which would be the same as the server submitted the job to the device.

1416 An alternative is for the server to respond to the client before or while sending the job to the device, instead  
1417 of waiting until the server has finished sending the job to the device. In this case, the server can return the  
1418 job's state as 'pending' with the 'job-outgoing' value in the job's "job-state-reasons" attribute.

1419 If the server doesn't know for sure whether the job completed successfully (or at all), it could return the  
1420 (out-of-band) 'unknown' value.

1421 On the other hand, if the server is able to query the device and/or setup some sort of event notification that  
1422 the device initiates when the job makes state transitions, then the server can return the current job state in  
1423 the Print-Job response and in subsequent queries because the server knows what the job state is in the  
1424 device (or can query the device).

1425 All of these alternatives depend on implementation of the server and the device.

## 1426 2.11 Flow controlling the data portion of a Print-Job request (Issue 1.22)

1427 A paused printer (or one that is stopped due to paper out or jam or spool space full or buffer space full, may  
1428 flow control the data of a Print-Job operation (at the TCP/IP layer), so that the client is not able to send all  
1429 the document data. Consequently, the Printer will not return a response until the condition is changed.

1430 The Printer should not return a Print-Job response with an error code in any of these conditions, since either  
1431 the printer will be resumed and/or the condition will be freed either by human intervention or as jobs print.

1432 In writing test scripts to test IPP Printers, the script must also be written not to expect a response, if the  
1433 printer has been paused, until the printer is resumed, in order to work with all possible implementations.

## 1434 2.12 Multi-valued attributes (Issue 1.31)

1435 What is the attribute syntax for a multi-valued attribute? Since some attributes support values in more than  
1436 one data type, such as "media", "job-hold-until", and "job-sheets", IPP semantics associate the attribute  
1437 syntax with each value, not with the attribute as a whole. The protocol associates the attribute syntax tag  
1438 with each value. Don't be fooled, just because the attribute syntax tag comes before the attribute keyword.  
1439 All attribute values after the first have a zero length attribute keyword as the indication of a subsequent  
1440 value of the same attribute.

## 1441 2.13 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)

1442 The following clarification was added to [ipp-mod] section 8.5:

## 1443 8.5 Queries on jobs submitted using non-IPP protocols

1444 If the device that an IPP Printer is representing is able to accept jobs using other job submission  
1445 protocols in addition to IPP, it is RECOMMEND that such an implementation at least allow such  
1446 "foreign" jobs to be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an  
1447 implementation NEED NOT support all of the same IPP job attributes as for IPP jobs. The IPP  
1448 object returns the 'unknown' out-of-band value for any requested attribute of a foreign job that is  
1449 supported for IPP jobs, but not for foreign jobs.

1450 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such  
1451 "foreign jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-  
1452 Attributes and Cancel-Job. Such an implementation also needs to deal with the problem of  
1453 authentication of such foreign jobs. One approach would be to treat all such foreign jobs as  
1454 belonging to users other than the user of the IPP client. Another approach would be for the foreign  
1455 job to belong to 'anonymous'. Only if the IPP client has been authenticated as an operator or  
1456 administrator of the IPP Printer object, could the foreign jobs be queried by an IPP request.  
1457 Alternatively, if the security policy is to allow users to query other users' jobs, then the foreign jobs  
1458 would also be visible to an end-user IPP client using Get-Jobs and Get-Job-Attributes.

1459 Thus IPP MAY be implemented as a "universal" protocol that provides access to jobs submitted with any  
1460 job submission protocol. As IPP becomes widely implemented, providing a more universal access makes  
1461 sense.

## 1462 2.14 The 'none' value for empty sets (Issue 1.37)

1463 [ipp-mod] states that the 'none' value should be used as the value of a 1SetOf when the set is empty. In most  
1464 cases, sets that are potentially empty contain keywords so the keyword 'none' is used, but for the 3  
1465 finishings attributes, the values are enums and thus the empty set is represented by the enum 3. Currently  
1466 there are no other attributes with 1SetOf values which can be empty and can contain values that are not  
1467 keywords. This exception requires special code and is a potential place for bugs. It would have been better  
1468 if we had chosen an out-of-band value, either "no-value" or some new value, such as 'none'. Since we  
1469 didn't, implementations have to deal with the different representations of 'none', depending on the attribute  
1470 syntax.

1471 2.15 Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?

1472 In [ipp-mod] section 3.2.6.1 'Get-Jobs Request', if the attribute 'my-jobs' is present and set to TRUE, MUST  
1473 the 'requesting-user-name' attribute be there to, and if it's not present what should the IPP printer do?

1474 [ipp-mod] Section 8.3 describes the various cases of "requesting-user-name" being present or not for any  
1475 operation. If the client does not supply a value for "requesting-user-name", the printer MUST assume that  
1476 the client is supplying some anonymous name, such as "anonymous".

1477 2.16 The "multiple-document-handling" Job Template attribute and support of multiple document jobs

1478 ISSUE: IPP/1.0 is silent on which of the four effects an implementation would perform if it supports  
1479 Create-Job, but does not support "multiple-document-handling".

1480 A fix to IPP/1.0 would be to require implementing all four values of "multiple-document-handling" if  
1481 Create-Job is supported at all. Or at least 'single-document-new-sheet' and 'separate-documents-uncollated-  
1482 copies'. In any case, an implementation that supports Create-Job SHOULD also support "multiple-  
1483 document-handling". Support for all four values is RECOMMENDED, but at least the 'single-document-  
1484 new-sheet' and 'separate-documents-uncollated-copies' values, along with the "multiple-document-  
1485 handling-default" indicating the default behavior and "multiple-document-handling-supported" values. If  
1486 an implementation spools the data, it should also support the 'separate-documents-collated-copies' value as  
1487 well.

### 1488 3 Encoding and Transport

1489 This section discusses various aspects of IPP/1.0 Encoding and Transport [IPP-PRO].

1490 ~~The IPP layer doesn't have to deal with chunking. In the context of CGI scripts, the HTTP layer removes~~  
1491 ~~any chunking information in the received data.~~

1492 A server is not required to send a response until after it has received the client's entire request. Hence, a  
1493 client must not expect a response until after it has sent the entire request. However, we recommend that the  
1494 server return a response as soon as possible if an error is detected while the client is still sending the data,  
1495 rather than waiting until all of the data is received. Therefore, we also recommend that A client MUST  
1496 NOT expect a response from an IPP server until after the client has sent the entire response. But a client  
1497 MAY listen for an error response that an IPP server MAY send before it receives all the data. In this case a  
1498 client, if chunking the data, can send a premature zero-length chunk to end the request before sending all  
1499 the data (and so the client can keep the connection open for other requests, rather than closing it). If the  
1500 request is blocked for some reason, a client MAY determine the reason by opening another connection to  
1501 query the server using Get-Printer-Attributes.

1502 In the following sections, there are a tables of all HTTP headers which describe their use in an IPP client or  
1503 server. The following is an explanation of each column in these tables.

- 1504     • the "header" column contains the name of a header  
1505     • the "request/client" column indicates whether a client sends the header.

- 1506       • the “request/ server” column indicates whether a server supports the header when received.  
 1507       • the “response/ server” column indicates whether a server sends the header.  
 1508       • the “response /client” column indicates whether a client supports the header when received.  
 1509       • the “values and conditions” column specifies the allowed header values and the conditions for the  
 1510       header to be present in a request/response.

1511       The table for “request headers” does not have columns for responses, and the table for “response headers”  
 1512       does not have columns for requests.

1513       The following is an explanation of the values in the “request/client” and “response/ server” columns.

- 1514       • **must:** the client or server **MUST** send the header,  
 1515       • **must-if:** the client or server **MUST** send the header when the condition described in the “values and  
 1516       conditions” column is met,  
 1517       • **may:** the client or server **MAY** send the header  
 1518       • **not:** the client or server **SHOULD NOT** send the header. It is not relevant to an IPP  
 1519       implementation.

1520       The following is an explanation of the values in the “response/client” and “request/ server” columns.

- 1521       • **must:** the client or server **MUST** support the header,  
 1522       • **may:** the client or server **MAY** support the header  
 1523       • **not:** the client or server **SHOULD NOT** support the header. It is not relevant to an IPP  
 1524       implementation.

### 1525   3.1   General Headers

1526       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-	must-if	must	must-if	must	“chunked” only . Header <b>MUST</b> be present if Content-Length is

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

## 1527 3.2 Request Headers

1528 The following is a table for the request headers.

Request-Header	Client	Server	Request Values and Conditions
Accept	may	must	“application/ipp” only. This value is the default if the client omits it
Accept-Charset	not	not	Charset information is within the application/ipp entity
Accept-Encoding	may	must	empty and per RFC 2068 [RFC2068] and IANA registry for content-codings
Accept-Language	not	not	language information is within the application/ipp entity
Authorization	must-if	must	per RFC 2068. A client MUST send this header when it receives a 401 “Unauthorized” response and does not receive a “Proxy-Authenticate” header.
From	not	not	per RFC 2068. Because RFC recommends sending this header only with the user’s approval, it is not very useful
Host	must	must	per RFC 2068
If-Match	not	not	
If-Modified-Since	not	not	
If-None-Match	not	not	
If-Range	not	not	
If-Unmodified-Since	not	not	
Max-Forwards	not	not	
Proxy-Authorization	must-if	not	per RFC 2068. A client MUST send this header when it receives a 401 “Unauthorized” response

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

## 1529 3.3 Response Headers

1530 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.

## 1531 3.4 Entity Headers

1532 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.



Entity-Header	Request		Response		Values and Conditions
	Client	Server	Server	Client	
Content-Language	not	not	not	not	Application/ipp handles language
Content-Length	must-if	must	must-if	must	the length of the message-body per RFC 2068. Header <b>MUST</b> be present if Transfer-Encoding is absent..
Content-Location	not	not	not	not	
Content-MD5	may	may	may	may	per RFC 2068
Content-Range	not	not	not	not	
Content-Type	must	must	must	must	“application/ipp” only
ETag	not	not	not	not	
Expires	not	not	not	not	
Last-Modified	not	not	not	not	

### 1533 3.5 Optional support for HTTP/1.0

1534 IPP implementations consist of an HTTP layer and an IPP layer. In the following discussion, the term  
 1535 "client" refers to the HTTP client layer and the term "server" refers to the HTTP server layer. The  
 1536 Encoding and Transport document [IPP-PRO] requires that HTTP 1.1 **MUST** be supported by all clients  
 1537 and all servers. However, a client and/or a server implementation may choose to also support HTTP 1.0.

1538 • This option means that a server may choose to communicate with a (non-conforming) client that only  
 1539 supports HTTP 1.0. In such cases the server should not use any HTTP 1.1 specific parameters or  
 1540 features and should respond using HTTP version number 1.0.

1541 • This option also means that a client may choose to communicate with a (non-conforming) server that  
 1542 only supports HTTP 1.0. In such cases, if the server responds with an HTTP ‘unsupported version  
 1543 number’ to an HTTP 1.1 request, the client should retry using HTTP version number 1.0.

### 1544 3.6 HTTP/1.1 Chunking

#### 1545 3.6.1 Disabling IPP Server Response Chunking

1546 Clients **MUST** anticipate that the HTTP/1.1 server may chunk responses and **MUST** accept them in  
 1547 responses. However, a (non-conforming) HTTP client that is unable to accept chunked responses may

1548 attempt to request an HTTP 1.1 server not to use chunking in its response to an operation by using the  
1549 following HTTP header:

1550           TE: identity

1551 This mechanism should not be used by a server to disable a client from chunking a request, since chunking  
1552 of document data is an important feature for clients to send long documents.

### 1553 3.6.2 Warning About the Use of Chunked Requests with CGI Script Implementations

1554 This section describes some problems with the use of chunked requests and IPP servers that are  
1555 implemented using CGI [CGI] scripts. About chunked POST in HTTP/1.1:

- 1556           1. All HTTP/1.1 [HTTP] applications (IPP recipients) that receive entities MUST accept the  
1557 "chunked" transfer-coding, thus allowing this mechanism to be used for messages when the  
1558 message length cannot be determined in advance.
- 1559           2. However, an origin server MAY refuse to accept a request without a defined Content-Length by  
1560 responding with status code 411 (Length Required).
- 1561           3. The Content-Length header field MUST NOT be sent if a Transfer-Encoding header field is  
1562 present.
- 1563           4. An origin server acting as a CGI 1.1 gateway for a request MUST determine and set the  
1564 CONTENT\_LENGTH metavariabale.
- 1565           5. There is currently nothing in the HTTP, CGI, or servlet [Servlet] specs to guarantee that origin  
1566 servers will remove the Transfer-Encoding before passing a request body to a plug-in, servlet,  
1567 (Fast)CGI, or across any other gateway boundary.

1568 Origin servers supporting CGI 1.1 have two options when receiving a POST request with "Transfer-  
1569 Encoding: chunked" for a CGI 1.1 resource:

- 1570           1. Reject the request with 411 (Length Required).
- 1571           2. Filter and buffer the request to determine CONTENT\_LENGTH before passing the decoded  
1572 request body to the CGI application. If the buffered request grows too large, the server MAY reject  
1573 the request with status code 413 (Request Entity Too Large) and the server MAY close the  
1574 connection to prevent the client from continuing the request.

1575 Origin servers supporting the Servlet API 2.1 [Servlet] have three options when receiving a POST request  
1576 with "Transfer-Encoding: chunked" for a servlet resource:

- 1577           1. Reject the request with 411 (Length Required).
- 1578           2. Filter and buffer the request to determine CONTENT\_LENGTH before passing the decoded  
1579 request body to the servlet. If the buffered request grows too large, the server MAY reject the

1580 request with status code 413 (Request Entity Too Large) and the server MAY close the connection  
1581 to prevent the client from continuing the request.

1582 3. Pass a filtered input stream to the servlet and filter the request body on-the-fly to remove the  
1583 chunked transfer-coding. Indicate the end of the request body with EOF (end of file) on the servlet  
1584 input stream.

### 1585 3.6.2.1 Implications for IPP

1586 Chunking takes place in the transport layer, and is not part of the IPP protocol itself. In the context of CGI  
1587 scripts, the HTTP layer either rejects a chunked POST request with 411 or removes any chunking  
1588 information in the received data and supplies CONTENT\_LENGTH. The CGI/1.1 spec doesn't explicitly  
1589 state that the HTTP server is required to decode the transfer-coding before passing the request body to the  
1590 CGI application, but this behavior is virtually guaranteed by the massive install base of old CGI scripts in  
1591 the world.

1592 The HTTP/1.1 standard does not guarantee that an origin server will accept chunked requests, regardless of  
1593 the resource identified in the request.

## 1594 **4 References**

### 1595 [CGI]

1596 CGI/1.1 (<http://www.ietf.org/internet-drafts/draft-coar-cgi-v11-00.txt>).

### 1597 [HTTP]

1598 HTTP/1.1 (<http://www.ietf.org/internet-drafts/draft-ietf-http-v11-spec-rev-06.txt>)

### 1599 [IPP LPD]

1600 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", draft-  
1601 ietf-ipp-lpd-ipp-map-04.txt, June 1998.

### 1602 [IPP-MOD]

1603 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and  
1604 Semantics", draft-ietf-ipp-model-11.txt, November, 1998.

### 1605 [IPP-PRO]

1606 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and  
1607 Transport", draft-ietf-ipp-pro-06.txt, June, 1998.

### 1608 [IPP-RAT]

1609 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",  
1610 draft-ietf-ipp-rat-03.txt, June, 1998.

### 1611 [IPP-REQ]

1612 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-02.txt, June, 1998.

- 1613 [RFC1123]  
1614 Braden, S., "Requirements for Internet Hosts - Application and Support", RFC 1123, October, 1989.
- 1615 [RFC2068]  
1616 R Fielding, et al, "Hypertext Transfer Protocol – HTTP/1.1" RFC 2068, January 1997.
- 1617 [rfc2119]  
1618 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March  
1619 1997.
- 1620 [RFC2396]  
1621 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic  
1622 Syntax", RFC 2396, August 1998.
- 1623 [Servlet]  
1624 [Servlet Specification Version 2.1 \(http://java.sun.com/products/servlet/2.1/index.html\)](http://java.sun.com/products/servlet/2.1/index.html).
- 1625 [SSL]  
1626 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.
- 1627

#### 1628 4.1 Authors' Address

1629 Thomas N. Hastings  
1630 Xerox Corporation  
1631 701 Aviation Blvd.  
1632 El Segundo, CA 90245  
1633 [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)  
1634  
1635 Carl-Uno Manros  
1636 Xerox Corporation  
1637 701 Aviation Blvd.  
1638 El Segundo, CA 90245  
1639 [manros@cp10.es.xerox.com](mailto:manros@cp10.es.xerox.com)

### 1640 **5 Appendix C: Full Copyright Statement**

1641 Copyright (C) The Internet Society (1998). All Rights Reserved

1642 This document and translations of it may be copied and furnished to others, and derivative works that  
1643 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and  
1644 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice  
1645 and this paragraph are included on all such copies and derivative works. However, this document itself  
1646 may not be modified in any way, such as by removing the copyright notice or references to the Internet  
1647 Society or other Internet organizations, except as needed for the purpose of developing Internet standards

1648 in which case the procedures for copyrights defined in the Internet Standards process must be followed, or  
1649 as required to translate it into languages other than English.

1650 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its  
1651 successors or assigns.

1652 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET  
1653 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,  
1654 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE  
1655 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED  
1656 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## 1657 **6 Change History**

1658 The change history is in reverse chronological order:

1659 ~~4.1~~6.1 Changes to produce the January 8, 1999 version from the December 6, 1998 version:

- 1660 1. Added section 3.6.2: Warning About the Use of Chunked Requests with CGI Script  
1661 Implementations
- 1662 2. Section 2.2.1.2: changed "printer-operations-supported" to "operations-supported".
- 1663 3. Section 2.2.1.6: changed "job-media-supported" to "job-media-sheets-supported"
- 1664 4. Section 2.2.3: separated the validation checks for variable length attributes into two separate tests:  
1665 one for correct attribute syntax and one for correct length.
- 1666 5. Section 2.2.3: changed "multiple-document-handling-supported" to "printer-resolution-supported"
- 1667 6. Section 2.6.1: recommended that an IPP object also support US-ASCII charset.
- 1668 7. Section 3: Clarified that a server is not required to send a response until after it has received the  
1669 client's entire request, but recommend that the server return a response as soon as possible if an  
1670 error is detected while the client is still sending the data, rather than waiting until all of the data is  
1671 received. Also recommended that a client listen for an error response that an IPP server MAY send  
1672 before it receives all the data.

1673 ~~6.1~~6.2 Changes to produce the December 6, 1998 version from the November 16, 1998 version:

1674 Included all of the remaining agreed issues raised before the November 16, 1998 production of the Internet-  
1675 Drafts for IPP/1.0 that included adding explanations to the Implementers Guide.

1676