

1 Internet Printing Protocol WG
2 INTERNET-DRAFT
3 draft-ietf-ipp-implementers-guide-v11-032.txt
4 Obsoletes RFC 2639
5 [Target category: informational]
6 Expires: January 17, 2002

T. Hastings
C. Manros
P. Zehler
Xerox Corporation
C. Kugler
IBM Printing Systems Co
H. Holst
i-data Printing Systems
~~July 17~~ ~~January 25, 2001~~

Internet Printing Protocol/1.1: Implementer's Guide

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

Status of this Memo

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

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

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

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

Abstract

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

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

33 ~~The full set of IPP documents includes:~~

34 ~~Design Goals for an Internet Printing Protocol [RFC2567]~~

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

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

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

38 ~~Mapping between LPD and IPP Protocols [RFC2569]~~

39 ~~The document, "Design Goals for an Internet Printing Protocol", takes a broad look at distributed~~
40 ~~printing functionality, and it enumerates real-life scenarios that help to clarify the features that need to~~
41 ~~be included in a printing protocol for the Internet. It identifies requirements for three types of users:~~
42 ~~end users, operators, and administrators. The design goal document calls out a subset of end user~~
43 ~~requirements that are satisfied in IPP/1.1. Operator and administrator requirements are out of scope for~~
44 ~~version 1.1.~~

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

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

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

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

57

58

58 **TABLE OF CONTENTS**

59	1	Introduction.....	6
60	1.1	Conformance language.....	6
61	1.2	Other terminology.....	7
62	1.3	Issues Raised from Interoperability Testing Events.....	7
63	2	IPP Objects.....	7
64	3	IPP Operations.....	9
65	3.1	Common Semantics.....	9
66	3.1.1	Summary of Operation Attributes	10
67	3.1.2	Suggested Operation Processing Steps for IPP Objects	15
68	3.1.2.1	Suggested Operation Processing Steps for all Operations	16
69	3.1.2.1.1	Validate version number.....	17
70	3.1.2.1.2	Validate operation identifier	17
71	3.1.2.1.3	Validate the request identifier.....	18
72	3.1.2.1.4	Validate attribute group and attribute presence and order	18
73	3.1.2.1.4.1	Validate the presence and order of attribute groups.....	18
74	3.1.2.1.4.2	Ignore unknown attribute groups in the expected position.....	18
75	3.1.2.1.4.3	Validate the presence of a single occurrence of required Operation attributes	19
76	3.1.2.1.5	Validate the values of the REQUIRED Operation attributes	25
77	3.1.2.1.6	Validate the values of the OPTIONAL Operation attributes	29
78	3.1.2.2	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add Documents	32
79	3.1.2.2.1	Default "ipp-attribute-fidelity" if not supplied.....	32
80	3.1.2.2.2	Check that the Printer object is accepting jobs.....	33
81	3.1.2.2.3	Validate the values of the Job Template attributes	33
82	3.1.2.3	Algorithm for job validation.....	33
83	3.1.2.3.1	Check for conflicting Job Template attributes values	38
84	3.1.2.3.2	Decide whether to REJECT the request	38
85	3.1.2.3.3	For the Validate-Job operation, RETURN one of the success status codes	40
86	3.1.2.3.4	Create the Job object with attributes to support.....	40
87	3.1.2.3.5	Return one of the success status codes	41
88	3.1.2.3.6	Accept appended Document Content	42
89	3.1.2.3.7	Scheduling and Starting to Process the Job.....	42
90	3.1.2.3.8	Completing the Job	42
91	3.1.2.3.9	Destroying the Job after completion	42
92	3.1.2.3.10	Interaction with "ipp-attribute-fidelity".....	43
93	3.1.2.3.11	Character set code conversion support	43
94	3.1.2.3.12	What charset to return when an unsupported charset is requested (Issue 1.19)?.....	44
95	3.1.2.3.13	Natural Language Override (NLO).....	44
96	3.1.3	Status codes returned by operation	45
97	3.1.3.1	Printer Operations.....	45
98	3.1.3.1.1	Print-Job.....	46
99	3.1.3.1.2	Print-URI.....	47
100	3.1.3.1.3	Validate-Job.....	48
101	3.1.3.1.4	Create-Job	48
102	3.1.3.1.5	Get-Printer-Attributes	48
103	3.1.3.1.6	Get-Jobs	49
104	3.1.3.1.7	Pause-Printer	49
105	3.1.3.1.8	Resume-Printer	50
106	3.1.3.1.8.1	What about Printers unable to change state due to an error condition?	50

108	3.1.3.1.8.2	How is "printer-state" handled on Resume-Printer?.....	51
109	3.1.3.1.9	Purge-Printer	51
110	3.1.3.2	Job Operations.....	51
111	3.1.3.2.1	Send-Document	51
112	3.1.3.2.2	Send-URI	52
113	3.1.3.2.3	Cancel-Job.....	52
114	3.1.3.2.4	Get-Job-Attributes	53
115	3.1.3.2.5	Hold-Job.....	54
116	3.1.3.2.6	Release-Job.....	54
117	3.1.3.2.7	Restart-Job	54
118	3.1.3.2.7.1	Can documents be added to a restarted job?.....	55
119	3.1.4	Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)	55
120	3.1.5	Sending empty attribute groups.....	55
121	3.2	Printer Operations	56
122	3.2.1	Print-Job operation	56
123	3.2.1.1	Flow controlling the data portion of a Print-Job request (Issue 1.22).....	56
124	3.2.1.2	Returning job-state in Print-Job response (Issue 1.30)	56
125	3.2.2	Get-Printer-Attributes operation	57
126	3.2.3	Get-Jobs operation.....	57
127	3.2.3.1	Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?.....	57
128	3.2.3.2	Why is there a "limit" attribute in the Get-Jobs operation?.....	57
129	3.2.4	Create-Job operation.....	58
130	3.3	Job Operations	58
131	3.3.1	Validate-Job	58
132	3.3.2	Restart-Job	58
133	4	Object Attributes.....	59
134	4.1	Attribute Syntax's.....	59
135	4.1.1	The 'none' value for empty sets (Issue 1.37)	59
136	4.1.2	Multi-valued attributes (Issue 1.31).....	59
137	4.1.3	Case Sensitivity in URIs (issue 1.6).....	59
138	4.1.4	Maximum length for xxxWithLanguage and xxxWithoutLanguage.....	60
139	4.2	Job Template Attributes	60
140	4.2.1	multiple-document-handling(type2 keyword)	60
141	4.2.1.1	Support of multiple document jobs.....	60
142	4.3	Job Description Attributes.....	61
143	4.3.1	Getting the date and time of day.....	61
144	4.4	Printer Description Attributes.....	61
145	4.4.1	queued-job-count (integer(0:MAX))	61
146	4.4.1.1	Why is "queued-job-count" RECOMMENDED (Issue 1.14)?.....	61
147	4.4.1.2	Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?.....	61
148	4.4.2	printer-current-time (dateTime).....	62
149	4.4.3	Printer-uri.....	62
150	4.5	Empty Jobs	62
151	5	Directory Considerations.....	63
152	5.1	General Directory Schema Considerations	63
153	5.2	IPP Printer with a DNS name	63
154	6	Security Considerations.....	63
155	6.1	Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)	
156		63	

157	7	Encoding and Transport	64
158	7.1	General Headers.....	65
159	7.2	Request Headers	66
160	7.3	Response Headers.....	67
161	7.4	Entity Headers	68
162	7.5	Optional support for HTTP/1.0.....	68
163	7.6	HTTP/1.1 Chunking.....	69
164	7.6.1	Disabling IPP Server Response Chunking	69
165	7.6.2	Warning About the Support of Chunked Requests	69
166	8	References	69
167	9	Authors' Address.....	71
168	10	Description of the Base IPP Documents	74
169	11	Full Copyright Statement	75
170			
171		TABLES	
172			
173		Table 1 - Summary of Printer operation attributes that sender MUST supply	10
174		Table 2 - Summary of Printer operation attributes that sender MAY supply.....	11
175		Table 3 - Summary of Job operation attributes that sender MUST supply.....	12
176		Table 4 - Summary of Job operation attributes that sender MAY supply.....	13
177		Table 5 - Printer operation response attributes.....	14
178		Table 6 - Examples of validating IPP version.....	17
179		Table 7 - Rules for validating single values X against Z.....	34
180			
181			

181
182
183

184 1 Introduction

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

189 [Design Goals for an Internet Printing Protocol \[RFC2567\]](#)
190 [Rationale for the Structure and Model and Protocol for the Internet Printing Protocol \[RFC2568\]](#)
191 [Internet Printing Protocol/1.1: Model and Semantics \[RFC2911\]](#)
192 [Internet Printing Protocol/1.1: Encoding and Transport \[RFC2910\]](#)
193 [Internet Printing Protocol/1.1: Implementer's Guide \(this document\)](#)
194 [Mapping between LPD and IPP Protocols \[RFC2569\]](#)

195
196 See section 10 for a description of these base IPP documents. Anyone reading these documents for the
197 first time is strongly encouraged to read the IPP documents in the above order.

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

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

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

210 1.1 Conformance language

211 Usually, this document does not contain the terminology MUST, MUST NOT, MAY, NEED NOT,
212 SHOULD, SHOULD NOT, REQUIRED, and OPTIONAL. However, when those terms do appear in
213 this document, their intent is to repeat what the [RFC2911] and [RFC2910] documents require and
214 allow, rather than specifying additional conformance requirements. These terms are defined in section
215 12 on conformance terminology in [RFC2911], most of which is taken from RFC 2119 [RFC2119].

216 Implementers should read section 12 (APPENDIX A) in [RFC2911] in order to understand these
217 capitalized words. The words MUST, MUST NOT, and REQUIRED indicate what implementations
218 are required to support in a client or IPP object in order to be conformant to [RFC2911] and
219 [RFC2910]. MAY, NEED NOT, and OPTIONAL indicate was is merely allowed as an implementer
220 option. The verbs SHOULD and SHOULD NOT indicate suggested behavior, but which is not
221 required or disallowed, respectively, in order to conform to the specification.

222 1.2 Other terminology

223 This document uses other terms, such as "attributes", "operation", and "Printer" as defined in
224 [RFC2911] section 12. In addition, the term "sender" refers to the client that sends a request or an IPP
225 object that returns a response. The term "receiver" refers to the IPP object that receives a request and
226 to a client that receives a response.

227 1.3 Issues Raised from Interoperability Testing Events

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

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

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

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

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

242 <ftp://ftp.pwg.org/pub/pwg/ipp/Issues/Issues-raised-at-Bake-Off3.pdf>
243 <ftp://ftp.pwg.org/pub/pwg/ipp/Issues/Issues-raised-at-Bake-Off3.doc>
244 <ftp://ftp.pwg.org/pub/pwg/ipp/Issues/Issues-raised-at-Bake-Off3.txt>

245 2 IPP Objects

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

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

253

253 **3 IPP Operations**

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

256 **3.1 Common Semantics**

257 This section discusses semantics common to all operations.

258 3.1.1 Summary of Operation Attributes

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

Operation Attributes	Printer Operations						
	Requests						Response s
	PJ, VJ (R)	PU (O)	CJ (O)	GPA (R)	GJ (R)	PP, RP, PP (O+)	All Operatio ns
Operation parameters--REQUIRED to be supplied by the sender:							
operation-id	R	R	R	R	R	R	
status-code							R
request-id	R	R	R	R	R	R	R
version-number	R	R	R	R	R	R	R
Operation attributes--REQUIRED to be supplied by the sender:							
attributes-charset	R	R	R	R	R	R	R
attributes-natural- language	R	R	R	R	R	R	R
document-uri		R					
job-id*							
job-uri*							
last-document							
printer-uri	R	R	R	R	R	R	
Operation attributes--RECOMMENDED to be supplied by the sender:							
job-name	R	R	R				
requesting-user-name	R	R	R	R	R	R	

260

261

Legend:

262

PJ, VJ: Print-Job, Validate-Job

263

PU: Print-URI

264

CJ: Create-Job

265

GPA: Get-Printer-Attributes

266

GJ: Get-Jobs

267

PP, RP, PP: Pause-Printer, Resume-Printer, Purge-Printer

268

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

269

270

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

271

272

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

273

274

275

276

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

Operation Attributes	Printer Operations							Responses
	Requests							
	PJ, (R)	VJ	PU (O)	CJ (O)	GPA (R)	GJ (R)	PP, RP, PP (O+)	All Operati ons
Operation attributes--OPTIONAL to be supplied by the sender:								
status-message								O
detailed-status-message								O
document-access-error								O**
compression	R+		R+					
document-format	R		R		R			
document-name	O		O					
document-natural-language	O		O					
ipp-attribute-fidelity	R		R	R				
job-impressions	O		O	O				
job-k-octets	O		O	O				
job-media-sheets	O		O	O				
limit						R		
message								
my-jobs						R		
requested-attributes					R	R		
which-jobs						R		

277

Legend:

278

PJ, VJ: Print-Job, Validate-Job

279

PU: Print-URI

280

CJ: Create-Job

281

GPA: Get-Printer-Attributes

282

GJ: Get-Jobs

283

PP, RP, PP: Pause-Printer, Resume-Printer, Purge-Printer

284

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

286

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

288

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

289

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

291

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

292

293

293

294

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

Operation Attributes	Job Operations					
	Requests					Responses
	SD (O)	SU (O)	CJ (R)	GJA (R)	HJ, RJ, RJ (O+)	All Operations
Operation parameters--REQUIRED to be supplied by the sender:						
operation-id	R	R	R	R	R	
status-code						R
request-id	R	R	R	R	R	R
version-number	R	R	R	R	R	R
Operation attributes--REQUIRED to be supplied by the sender:						
attributes-charset	R	R	R	R	R	R
attributes-natural-language	R	R	R	R	R	R
document-uri		R				
job-id*	R	R	R	R	R	
job-uri*	R	R	R	R	R	
last-document	R	R				
printer-uri	R	R	R	R	R	
Operation attributes--RECOMMENDED to be supplied by the sender:						
job-name						
requesting-user-name	R	R	R	R	R	

295

Legend:

296

SD: Send-Document

297

SU: Send-URI

298

CJ: Cancel-Job

299

GJA: Get-Job-Attributes

300

HJ, RJ, RJ: Hold-Job, Release-Job, Restart-Job

301

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

302

303

304

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

305

306

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

307

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

308

309

310

311

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

Operation Attributes	Job Operations						
	Requests						Responses
	SD (O)	SU (O)	CJ (R)	GJA (R)	HJ, RJ, RJ (O+)	SD (O)	All Operati ons
Operation attributes--OPTIONAL to be supplied by the sender:							
status-message							O
detailed-status-message							O
document-access-error							O**
compression	R O	R O					
document-format	R	R					
document-name	O	O					
document-natural-language	O	O					
ipp-attribute-fidelity							
job-impressions							
job-k-octets							
job-media-sheets							
limit							
message			O		O	O	
job-hold-until					R		
my-jobs							
requested-attributes				R			
which-jobs							

Legend:

- 312 SD: Send-Document
313 SU: Send-URI
314 CJ: Cancel-Job
315 GJA: Get-Job-Attributes
316 HJ, RJ, RJ: Hold-Job, Release-Job, Restart-Job
317 R indicates a REQUIRED operation that MUST be supported by the IPP object (Printer or Job). For
318 attributes, R indicates that the attribute MUST be supported by the IPP object that supports the
319 associated operation.
320 O indicates an OPTIONAL operation or attribute that MAY be supported by the IPP object (Printer
321 or Job).
322 + indicates that this is not an IPP/1.0 feature, but is only a part of IPP/1.1 and future versions of IPP.
323 * "job-id" is REQUIRED only if used together with "printer-uri" to identify the target job; otherwise,
324 "job-uri" is REQUIRED.
325 ** "document-access-error" applies to the Send-URI operation only

326

326

327

Table 5 - Printer operation response attributes

Operation Attributes	Printer Operations						
	Response						
	PJ (R) SD (O)	VJ (R)	PU (O) SU (O)	CJ (O)	GPA (R)	GJ (R)	PP, RP, PP (O+)
job-uri	R		R	R			
job-id	R		R	R			
job-state	R		R	R			
job-state-reasons	R+		R+	R+			
number-of- intervening-jobs	O		O	O			
document-access- error+			O				

328

Legend:

329

330 PJ, SJ: Print-Job, Send-Document

331

VJ: Validate-Job

332

PU, SU: Print-URI, Send-URI

333

CJ: Create-Job

334

GPA: Get-Printer-Attributes

335

GJ: Get-Jobs

336

PP, RP, PP: Pause-Printer, Resume-Printer, Purge-Printer

337

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

338

339

340

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

341

342

343

343

344 **3.1.2 Suggested Operation Processing Steps for IPP Objects**

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

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

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

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

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

372

372 **3.1.2.1 Suggested Operation Processing Steps for all Operations**

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

376	IIG Sect #	Flow	IPP error status codes
377	-----	----	-----
378			
379		v	err
380	3.1.2.1.1	<Validate version>	--> server-error-version-not-
381			supported
382		ok	
383		v	err
384	3.1.2.1.2	<Validate operation>	--> server-error-operation-not-
385			supported
386		ok	
387		v	err
388	3.1.2.1.4.1-	<Validate presence>	--> client-error-bad-request
389	3.1.2.1.4.2	<of attributes>	
390		ok	
391		v	err
392	3.1.2.1.4.3	<Validate presence>	--> client-error-bad-request
393		<of operation attr>	
394		ok	
395		v	err
396	3.1.2.1.5	<Validate values of>	----> client-error-bad-request
397		<operation attrs>	client-error-request-value-
398			too-long
399		<(length, tag, range,>	
400		<multi-value)>	
401		ok	
402		v	err
403	3.1.2.1.5	<Validate values>	--> client-error-bad-request
404		<with supported values>	client-error-charset-not-
405			supported
406		ok	client-error-attributes-or-
407			values-
408			not-supported
409		v	err
410	3.1.2.1.6	<Validate optionally>	--> client-error-bad-request
411		<operation attr>	client-error-natural-language-
412			not-supported
413			client-error-request-value-
414			too-long
415			client-error-attributes-or-
416			values-not-supported
417			

418 3.1.2.1.1 Validate version number

419 Every request and every response contains the "version-number" attribute. The value of this attribute is
 420 the major and minor version number of the syntax and semantics that the client and IPP object is using,
 421 respectively. The "version-number" attribute remains in a fixed position across all future versions so
 422 that all clients and IPP object that support future versions can determine which version is being used.
 423 The IPP object checks to see if the major version number supplied in the request is supported. If not,
 424 the Printer object REJECTS the request and RETURNS the 'server-error-version-not-supported' status
 425 code in the response. The IPP object returns in the "version-number" response attribute the major and
 426 minor version for the error response. Thus the client can learn at least one major and minor version that
 427 the IPP object supports. The IPP object is encouraged to return the closest version number to the one
 428 supplied by the client.

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

436 **Table 6 - Examples of validating IPP version**

Client supplies	Printer Accept Request?	Printer returns
1.0	yes (SHOULD)	1.1
1.0	no (SHOULD NOT)	1.1
1.1	yes (MUST)	1.1
1.2	yes (MUST)	1.2
1.3	yes (SHOULD)	1.2
1.3	no (SHOULD NOT)	1.2

437

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

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

443 3.1.2.1.2 Validate operation identifier

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

447 **3.1.2.1.3 Validate the request identifier**

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

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

454 **3.1.2.1.4 Validate attribute group and attribute presence and order**

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

456 **3.1.2.1.4.1 Validate the presence and order of attribute groups**

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

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

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

469 **3.1.2.1.4.2 Ignore unknown attribute groups in the expected position**

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

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

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

482 **3.1.2.1.4.3 Validate the presence of a single occurrence of required Operation attributes**

483 Client requests and IPP object responses contain Operation attributes that [RFC2911] Section 3
484 requires to be present. Attributes within a group may be in any order, except for the ordering of target,
485 charset, and natural languages attributes. These attributes **MUST** be first, and **MUST** be supplied in the
486 following order: charset, natural language, and then target. An IPP object verifies that the attributes that
487 Section 4 requires to be supplied by the client have been supplied in the request (attributes without an *
488 in the following tables). An asterisk (*) indicates groups and Operation attributes that the client may
489 omit in a request or an IPP object may omit in a response.

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

493 REJECTS the request and RETURNS the 'client-error-bad-request' status code

494 accepts the request and uses the first occurrence of the attribute no matter where it is

495 accepts the request and uses the last occurrence of the attribute no matter where it is

496 accept the request and assume some default value for the missing attribute

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

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

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

511 R indicates a **REQUIRED** attribute or operation that an IPP object **MUST** support

512 O indicates an **OPTIONAL** attribute or operation that an IPP object **NEED NOT** support

513 * indicates that a client MAY omit the attribute in a request and that an IPP object MAY omit
514 the attribute in a response. The absence of an * means that a client MUST supply the
515 attribute in a request and an IPP object MUST supply the attribute in a response.
516 + indicates that this is not a IPP/1.0 operation, but is only a part of IPP/1.1 and future versions
517 of IPP.

518

519 Operation Requests

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

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

523

524 Print-Job Request (R):

525 Group 1: Operation Attributes (R)
526 attributes-charset (R)
527 attributes-natural-language (R)
528 printer-uri (R)
529 requesting-user-name (R*)
530 job-name (R*)
531 ipp-attribute-fidelity (R*)
532 document-name (R*)
533 document-format (R*)
534 document-natural-language (O*)
535 compression (R~~O~~*)
536 job-k-octets (O*)
537 job-impressions (O*)
538 job-media-sheets (O*)
539 Group 2: Job Template Attributes (R*)
540 <Job Template attributes> (O*)
541 (see [RFC2911] Section 4.2)
542 Group 3: Document Content (R)
543 <document content>

544

545 Validate-Job Request (R):

546 Group 1: Operation Attributes (R)
547 attributes-charset (R)
548 attributes-natural-language (R)
549 printer-uri (R)
550 requesting-user-name (R*)
551 job-name (R*)
552 ipp-attribute-fidelity (R*)
553 document-name (R*)
554 document-format (R*)
555 document-natural-language (O*)
556 compression (R~~O~~*)
557 job-k-octets (O*)
558 job-impressions (O*)

559 job-media-sheets (O*)
560 Group 2: Job Template Attributes (R*)
561 <Job Template attributes> (O*)
562 (see [RFC2911] Section 4.2)
563
564 Print-URI Request (O):
565 Group 1: Operation Attributes (R)
566 attributes-charset (R)
567 attributes-natural-language (R)
568 printer-uri (R)
569 document-uri (R)
570 requesting-user-name (R*)
571 job-name (R*)
572 ipp-attribute-fidelity (R*)
573 document-name (R*)
574 document-format (R*)
575 document-natural-language (O*)
576 compression (R~~O~~*)
577 job-k-octets (O*)
578 job-impressions (O*)
579 job-media-sheets (O*)
580 Group 2: Job Template Attributes (R*)
581 <Job Template attributes> (O*) (see
582 (see [RFC2911] Section 4.2)
583
584 Create-Job Request (O):
585 Group 1: Operation Attributes (R)
586 attributes-charset (R)
587 attributes-natural-language (R)
588 printer-uri (R)
589 requesting-user-name (R*)
590 job-name (R*)
591 ipp-attribute-fidelity (R*)
592 job-k-octets (O*)
593 job-impressions (O*)
594 job-media-sheets (O*)
595 Group 2: Job Template Attributes (R*)
596 <Job Template attributes> (O*) (see
597 (see [RFC2911] Section 4.2)
598
599 Get-Printer-Attributes Request (R):
600 Group 1: Operation Attributes (R)
601 attributes-charset (R)
602 attributes-natural-language (R)
603 printer-uri (R)
604 requesting-user-name (R*)
605 requested-attributes (R*)
606 document-format (R*)
607
608 Get-Jobs Request (R):

609 Group 1: Operation Attributes (R)
610 attributes-charset (R)
611 attributes-natural-language (R)
612 printer-uri (R)
613 requesting-user-name (R*)
614 limit (R*)
615 requested-attributes (R*)
616 which-jobs (R*)
617 my-jobs (R*)
618
619 Send-Document Request (O):
620 Group 1: Operation Attributes (R)
621 attributes-charset (R)
622 attributes-natural-language (R)
623 (printer-uri & job-id) | job-uri (R)
624 last-document (R)
625 requesting-user-name (R*)
626 document-name (R*)
627 document-format (R*)
628 document-natural-language (O*)
629 compression (R~~Θ~~*)
630 Group 2: Document Content (R*)
631 <document content>
632
633 Send-URI Request (O):
634 Group 1: Operation Attributes (R)
635 attributes-charset (R)
636 attributes-natural-language (R)
637 (printer-uri & job-id) | job-uri (R)
638 last-document (R)
639 document-uri (R)
640 requesting-user-name (R*)
641 document-name (R*)
642 document-format (R*)
643 document-natural-language (O*)
644 compression (R~~Θ~~*)
645
646 Cancel-Job Request (R):
647 Release-Job Request (O+):
648 Group 1: Operation Attributes (R)
649 attributes-charset (R)
650 attributes-natural-language (R)
651 (printer-uri & job-id) | job-uri (R)
652 requesting-user-name (R*)
653 message (O*)
654
655 Get-Job-Attributes Request (R):
656 Group 1: Operation Attributes (R)
657 attributes-charset (R)
658 attributes-natural-language (R)

659 (printer-uri & job-id) | job-uri (R)
660 requesting-user-name (R*)
661 requested-attributes (R*)
662
663 Pause-Printer Request (O+):
664 Resume-Printer Request (O+):
665 Purge-Printer Request (O+):
666 Group 1: Operation Attributes (R)
667 attributes-charset (R)
668 attributes-natural-language (R)
669 printer-uri (R)
670 requesting-user-name (R*)
671
672 Hold-Job Request (O+):
673 Restart-Job Request (O+):
674 Group 1: Operation Attributes (R)
675 attributes-charset (R)
676 attributes-natural-language (R)
677 (printer-uri & job-id) | job-uri (R)
678 requesting-user-name (R*)
679 job-hold-until (R*)
680 message (O*)

681

682 Operation Responses

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

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

686

687 Print-Job Response (R):
688 Create-Job Response (O):
689 Send-Document Response (O):
690 Group 1: Operation Attributes (R)
691 attributes-charset (R)
692 attributes-natural-language (R)
693 status-message (O*)
694 detailed-status-message (O*)
695 Group 2: Unsupported Attributes (R*) (see Note 3)
696 <unsupported attributes> (R*)
697 Group 3: Job Object Attributes (R*) (see Note 2)
698 job-uri (R)
699 job-id (R)
700 job-state (R)
701 job-state-reasons (O* | R+)
702 job-state-message (O*)
703 number-of-intervening-jobs (O*)

704

705 Validate-Job Response (R):

706 Cancel-Job Response (R):
707 Hold-Job Response (O+):
708 Release-Job Response (O+):
709 Restart-Job Response (O+):
710 Group 1: Operation Attributes (R)
711 attributes-charset (R)
712 attributes-natural-language (R)
713 status-message (O*)
714 detailed-status-message (O*)
715 Group 2: Unsupported Attributes (R*) (see Note 3)
716 <unsupported attributes> (R*)
717
718 Print-URI Response (O):
719 Send-URI Response (O):
720 Group 1: Operation Attributes (R)
721 attributes-charset (R)
722 attributes-natural-language (R)
723 status-message (O*)
724 detailed-status-message (O*)
725 document-access-error (O*)
726 Group 2: Unsupported Attributes (R*) (see Note 3)
727 <unsupported attributes> (R*)
728 Group 3: Job Object Attributes (R*) (see Note 2)
729 job-uri (R)
730 job-id (R)
731 job-state (R)
732 job-state-reasons (O* | R+)
733 job-state-message (O*)
734 number-of-intervening-jobs (O*)
735
736 Get-Printer-Attributes Response (R):
737 Group 1: Operation Attributes (R)
738 attributes-charset (R)
739 attributes-natural-language (R)
740 status-message (O*)
741 detailed-status-message (O*)
742 Group 2: Unsupported Attributes (R*) (see Note 4)
743 <unsupported attributes> (R*)
744 Group 3: Printer Object Attributes (R*) (see Note 2)
745 <requested attributes> (R*)
746
747 Get-Jobs Response (R):
748 Group 1: Operation Attributes (R)
749 attributes-charset (R)
750 attributes-natural-language (R)
751 status-message (O*)
752 detailed-status-message (O*)
753 Group 2: Unsupported Attributes (R*) (see Note 4)
754 <unsupported attributes> (R*)
755 Group 3: Job Object Attributes (R*) (see Note 2, 5)

756 <requested attributes> (R*)
757
758 Get-Job-Attributes Response (R):
759 Group 1: Operation Attributes (R)
760 attributes-charset (R)
761 attributes-natural-language (R)
762 status-message (O*)
763 detailed-status-message (O*)
764 Group 2: Unsupported Attributes (R*) (see Note 4)
765 <unsupported attributes> (R*)
766 Group 3: Job Object Attributes (R*) (see Note 2)
767 <requested attributes> (R*)
768
769 Pause-Printer Response (O+):
770 Resume-Printer Response (O+):
771 Purge-Printer Response (O+):
772 Group 1: Operation Attributes (R)
773 attributes-charset (R)
774 attributes-natural-language (R)
775 status-message (O*)
776 detailed-status-message (O*)
777 Group 2: Unsupported Attributes (R*) (see Note 4)
778 <unsupported attributes> (R*)
779

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

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

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

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

788 3.1.2.1.5 **Validate the values of the REQUIRED Operation attributes**

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

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

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

- 795 b) that the attribute syntax tag is correct for that Operation attribute according to
796 [RFC2911] Section 3,
- 797 c) that the value is in the range specified for that Operation attribute according to
798 [RFC2911] Section 3,
- 799 d) that multiple values are supplied by the client only for operation attributes that are multi-
800 valued, i.e., that are 1setOf X according to [RFC2911] Section 3.

801

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

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

814

815

attributes-charset (charset)

- 816 IF NOT a single non-empty 'charset' value, REJECT/RETURN 'client-error-bad-request'.
817 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-
818 long'.
819 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-
820 charset-not-supported".

821

822

attributes-natural-language(naturalLanguage)

- 823 IF NOT a single non-empty 'naturalLanguage' value, REJECT/RETURN 'client-error-bad-
824 request'.
825 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-
826 long'.
827 ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-
828 language-supported" attribute. If the supplied value is not a member of the Printer
829 object's "generated-natural-language-supported" attribute, use the Printer object's
830 "natural-language-configured" value.

831

832

requesting-user-name

833 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
834 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
835 too-long'.
836 IF the IPP object can obtain a better-authenticated name, use it instead.
837
838 job-name(name)

839 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
840 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
841 too-long'.
842 IF NOT supplied by the client, the Printer object creates a name from the document-name or
843 document-uri.
844
845 document-name (name)

846 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
847 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
848 too-long'.
849
850 ipp-attribute-fidelity (boolean)

851 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-
852 error-bad-request'.
853 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-
854 long'.
855 IF NOT supplied by the client, the IPP object assumes the value 'false'.
856
857 document-format (mimeType)

858 IF NOT a single non-empty 'mimeType' value, REJECT/RETURN 'client-error-bad-
859 request'.
860 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
861 too-long'.
862 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN
863 'client-error-document-format-not-supported'.
864 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's
865 "document-format-default" attribute.
866
867 document-uri (uri)

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

872 If the client-supplied URI scheme is not supported, i.e. the value is not in the Printer object's
873 referenced-uri-scheme-supported" attribute, the Printer object MUST reject the request
874 and return the 'client-error-uri-scheme-not-supported' status code. The Printer object
875 MAY check to see if the document exists and is accessible. If the document is not found
876 or is not accessible, REJECT/RETURN 'client-error-not found'.
877 last-document (boolean)
878 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-
879 error-bad-request'.
880 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-
881 long'
882
883 job-id (integer(1:MAX))
884 IF NOT an single 'integer' value equal to 4 octets AND in the range 1 to MAX,
885 REJECT/RETURN 'client-error-bad-request'.
886 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-
887 error-gone' status code, if keep track of recently deleted jobs.
888
889 requested-attributes (1setOf keyword)
890 IF NOT one or more 'keyword' values, REJECT/RETURN 'client-error-bad-request'.
891 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
892 too-long'.
893 Ignore unsupported values, which are the keyword names of unsupported attributes. Don't
894 bother to copy such requested (unsupported) attributes to the Unsupported Attribute
895 response group since the response will not return them.
896
897 which-jobs (type2 keyword)
898 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
899 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-
900 too-long'.
901 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to
902 the Unsupported Attributes response group and REJECT/RETURN 'client-error-
903 attributes-or-values-not-supported'.
904 Note: a Printer still supports the 'completed' value even if it keeps no
905 completed/canceled/aborted jobs: by returning no jobs when so queried.
906 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.
907
908 my-jobs (boolean)
909 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-
910 error-bad-request'.
911 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-
912 long'.
913 IF NOT supplied by the client, the IPP object assumes the 'false' value.

914

915 limit (integer(1:MAX))

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

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

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

919

920

921

922 **3.1.2.1.6 Validate the values of the OPTIONAL Operation attributes**

923 OPTIONAL Operation attributes are those that an IPP object MAY support. An IPP object validates
 924 the values of the OPTIONAL attributes supplied by the client. The IPP object performs the same
 925 syntactic validation checks for each OPTIONAL attribute value as in Section 3.1.2.1.5. As in Section
 926 3.1.2.1.5, if any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request'
 927 or the 'client-error-request-value-too-long' status code.

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

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

935

936 document-natural-language (naturalLanguage)

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

938 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-
939 long'.940 IF NOT a value that the Printer object supports in document formats, (no corresponding "xxx-
941 supported" Printer attribute), REJECT/RETURN 'client-error-natural-language-not-
942 supported'.

943

944 compression (type3 keyword)

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

946 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
947 long'.948 IF NOT in the Printer object's "compression-supported" attribute, ~~copy the attribute and the~~
949 ~~unsupported value to the Unsupported Attributes response group and~~ REJECT/RETURN
950 'client-error-~~attributes or values~~compression-not-supported'.

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

955

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

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

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

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

960 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN

961 'client-error-attributes-or-values-not-supported'.

962

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

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

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

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

967 and the unsupported value to the Unsupported Attributes response group and

968 REJECT/RETURN 'client-error-attributes-or-values-not-supported'.

969

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

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

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

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

974 and the unsupported value to the Unsupported Attributes response group and

975 REJECT/RETURN 'client-error-attributes-or-values-not-supported'.

976

977 message (text(127))

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

979 IF the value length is greater than 127 octets,

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

981

982 unknown or unsupported attribute

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

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

987

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

003

003 **3.1.2.2 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and**
 004 **Add Documents**

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

009	IIG Sect #	Flow	IPP error status codes
010	-----	----	-----
011			
012		v	No
013	3.1.2.2.1	<ipp-attribute-fidelity>	-----+-----
014		<supplied?>	
015		Yes	
016			
017		ipp-attribute-fidelity = no	
018		<-----+-----	
019		v	No
020	3.1.2.2.2	<Printer is	--> server-error-not-accepting-jobs
021		<accepting jobs?>	
022		Yes	
023		v	err
024	3.1.2.3	<Validate values of>	--> client-error-bad-request
025		<Job template attributes>	client-error-request-value-too-
026			long
027		<(length, tag, range,>	
028		<multi-value)>	
029		ok	
030		v	err
031	3.1.2.3	<Validate values with>	--> client-error-bad-request
032		<supported values>	client-error-attributes-or-
033			values-not-supported
034		v	err
035	3.1.2.3.1	<Any conflicting>	--> client-error-conflicting-
036			attributes
037		<Job Template attr values>	client-error-attributes-or-
038			values-not-supported
039		v	

040 **3.1.2.2.1 Default "ipp-attribute-fidelity" if not supplied**

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

043 **3.1.2.2.2 Check that the Printer object is accepting jobs**

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

046 **3.1.2.2.3 Validate the values of the Job Template attributes**

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

- 050 a) that the length of each value is correct for the attribute syntax tag supplied by the client
051 according to [RFC2911] Section 4.1.
- 052 b) that the attribute syntax tag is correct for that attribute according to [RFC2911]
053 Sections 4.2 to 4.4.
- 054 c) that multiple values are supplied only for multi-valued attributes, i.e., that are 1setOf X
055 according to [RFC2911] Sections 4.2 to 4.4.

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

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

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

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

070 **3.1.2.3 Algorithm for job validation**

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

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

- 075 1. If U is multi-valued, validate each value X of U by performing the algorithm in Table 7 with each
076 value X. Each validation is separate from the standpoint of returning unsupported values.
077 Example: If U is "finishings" that the client supplies with 'staple', 'bind' values, then X takes on
078 the successive values: 'staple', then 'bind'
- 079 2. If V is multi-valued, validate X against each Z of V by performing the algorithm in Table 7 with
080 each value Z. If a value Z validates, the validation for the attribute value X succeeds. If it fails,
081 the algorithm is applied to the next value Z of V. If there are no more values Z of V, validation
082 fails. Example" If V is "sides-supported" with values: 'one-sided', 'two-sided-long', and 'two-
083 sided-short', then Z takes on the successive values: 'one-sided', 'two-sided-long', and 'two-sided-
084 short'. If the client supplies "sides" with 'two-sided-long', the first comparison fails ('one-sided' is
085 not equal to 'two-sided-long'), the second comparison succeeds ('two-sided-long' is equal to 'two-
086 sided-long'), and the third comparison ('two-sided-short' with 'two-sided-long') is not even
087 performed.
- 088 3. If both U and V are single-valued, let X be U and Z be V and use the validation rules in Table 7.

089 **Table 7 - Rules for validating single values X against Z**

Attribute syntax of X	attribute syntax of Z	validated if:
integer	rangeOfInteger	X is within the range of Z
uri	uriScheme	the uri scheme in X is equal to Z
any	boolean	the value of Z is TRUE
any	any	X and Z are of the same type and are equal.

090

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

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

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

108 -----

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

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

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

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

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

119

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

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

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

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

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

128

129 job-sheets (type3 keyword | name)

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

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

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

135

136 multiple-document-handling (type2 keyword)

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

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

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

142

143 copies (integer(1:MAX))

144 IF NOT a single 'integer' value with a length equal to 4 octets,
145 REJECT/RETURN 'client-error-bad-request'.
146 IF NOT in range of the Printer object's "copies-supported" attribute
147 copy the attribute and the unsupported value to the Unsupported Attributes response group.
148
149 finishings (1setOf type2 enum)

150 IF NOT an 'enum' value(s) each with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
151 request'.
152 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the
153 unsupported value(s), but not any supported values, to the Unsupported Attributes response
154 group.
155
156 page-ranges (1setOf rangeOfInteger(1:MAX))

157 IF NOT a 'rangeOfInteger' value(s) each with a length equal to 8 octets, REJECT/RETURN 'client-
158 error-bad-request'.
159 IF first value is greater than second value in any range, the ranges are not in ascending order, or
160 ranges overlap, REJECT/RETURN 'client-error-bad-request'.
161 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to
162 the Unsupported Attributes response group and set the value to the "out-of-band"
163 'unsupported' value.
164
165 sides (type2 keyword)

166 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
167 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
168 long'.
169 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported
170 value to the Unsupported Attributes response group.
171
172 number-up (integer(1:MAX))

173 IF NOT a single 'integer' value with a length equal to 4 octets,
174 REJECT/RETURN 'client-error-bad-request'.
175 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"
176 attribute, copy the attribute and value to the Unsupported Attribute response group.
177
178 orientation-requested (type2 enum)

179 IF NOT a single 'enum' value with a length equal to 4 octets,
180 REJECT/RETURN 'client-error-bad-request'.
181 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the
182 unsupported value to the Unsupported Attributes response group.
183
184 media (type3 keyword | name)

185 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
 186 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-
 187 long'.
 188 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported
 189 value to the Unsupported Attributes response group.
 190
 191 printer-resolution (resolution)

192 IF NOT a single 'resolution' value with a length equal to 9 octets,
 193 REJECT/RETURN 'client-error-bad-request'.
 194 IF NOT in the Printer object's "printer-resolution-supported" attribute, copy the attribute and the
 195 unsupported value to the Unsupported Attributes response group.
 196
 197 print-quality (type2 enum)

198 IF NOT a single 'enum' value with a length equal to 4 octets,
 199 REJECT/RETURN 'client-error-bad-request'.
 200 IF NOT in the Printer object's "print-quality-supported" attribute, copy the attribute and the
 201 unsupported value to the Unsupported Attributes response group.
 202
 203 unknown or unsupported attribute (i.e., there is no corresponding Printer object "xxx-supported"
 204 attribute)

205 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute
 206 syntax,
 207 REJECT/RETURN 'client-error-bad-request' if the length of the attribute syntax is fixed or 'client-
 208 error-request-value-too-long' if the length of the attribute syntax is variable.
 209 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
 210 attribute value to the "out-of-band" 'unsupported' value. Any remaining Job Template
 211 Attributes are either unknown or unsupported Job Template attributes and are validated
 212 algorithmically according to their attribute syntax for proper length (see below).
 213 -----

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

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

227	'keyword'	<= 255
228	'enum'	= 4
229	'uri'	<= 1023
230	'uriScheme'	<= 63
231	'charset'	<= 63
232	'naturalLanguage'	<= 63
233	'mimeType'	<= 255
234	'octetString'	<= 1023
235	'boolean'	= 1
236	'integer'	= 4
237	'rangeOfInteger'	= 8
238	'dateTime'	= 11
239	'resolution'	= 9
240	'lsetOf X'	

241

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

247 3.1.2.3.1 Check for conflicting Job Template attributes values

248 Once all the Operation and Job Template attributes have been checked individually, the Printer object
 249 SHOULD check for any conflicting values among all the supported values supplied by the client. For
 250 example, a Printer object might be able to staple and to print on transparencies, however due to physical
 251 stapling constraints, the Printer object might not be able to staple transparencies. The IPP object copies
 252 the supported attributes and their conflicting attribute values to the Unsupported Attributes response
 253 group. The Printer object only copies over those attributes that the Printer object either ignores or
 254 substitutes in order to resolve the conflict, and it returns the original values which were supplied by the
 255 client. For example suppose the client supplies "finishings" equals 'staple' and "media" equals
 256 'transparency', but the Printer object does not support stapling transparencies. If the Printer chooses to
 257 ignore the stapling request in order to resolve the conflict, the Printer objects returns "finishings" equal
 258 to 'staple' in the Unsupported Attributes response group. If any attributes are multi-valued, only the
 259 conflicting values of the attributes are copied.

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

261 3.1.2.3.2 Decide whether to REJECT the request

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

- 265 1. 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes
 266 supplied by the client.

267 2. 'client-error-attributes-or-values-not-supported' status code, otherwise.

268

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

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

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

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

- 280 - the Printer supports auto-sensing,
 - 281 - the request "document-format" operation attribute is 'application/octet-stream',
 - 282 - the Printer receives document data before responding,
 - 283 - the Printer auto-senses the document format before responding,
 - 284 - the sensed document format is not supported by the Printer
- 285 then the Printer should respond with 'client-error-document-format-not-supported' status.

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

- 288 - the client supplies a supported value for the "compression" operation attribute in the request
 - 289 - the Printer receives document data before responding,
 - 290 - the Printer attempts to decompress the document data before responding,
 - 291 - the document data cannot be decompressed using the algorithm specified by the "compression"
292 operation attribute
- 293 then the Printer should respond with 'client-error-compression-error' status.

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

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

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

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

309 **3.1.2.3.3 For the Validate-Job operation, RETURN one of the success status codes**

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

- 311 1. the "successful-ok" status code, if there are no unsupported or conflicting Job Template
312 attributes or values.
- 313 2. the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or
314 values.
- 315 3. the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template
316 attributes or values.

317

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

322 **3.1.2.3.4 Create the Job object with attributes to support**

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

- 324 1. creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and
325 initializes all of the job's other supported Job Description attributes.
- 326 2. removes all unsupported attributes from the Job object.
- 327 3. for each unsupported value, removes either the unsupported value or substitutes the
328 unsupported attribute value with some supported value. If an attribute has no values after
329 removing unsupported values from it, the attribute is removed from the Job object (so that the
330 normal default behavior at job processing time will take place for that attribute).
- 331 4. for each conflicting value, removes either the conflicting value or substitutes the conflicting
332 attribute value with some other supported value. If an attribute has no values after removing
333 conflicting values from it, the attribute is removed from the Job object (so that the normal
334 default behavior at job processing time will take place for that attribute).

335

336 If there were no attributes or values flagged as unsupported, or the value of 'ipp-attribute-fidelity' was
337 'false', the Printer object is able to accept the create request and create a new Job object. If the "ipp-
338 attribute-fidelity" attribute is set to 'true', the Job Template attributes that populate the new Job object
339 are necessarily all the Job Template attributes supplied in the create request. If the "ipp-attribute-
340 fidelity" attribute is set to 'false', the Job Template attributes that populate the new Job object are all the
341 client supplied Job Template attributes that are supported or that have value substitution. Thus, some
342 of the requested Job Template attributes will not appear in the Job object because the Printer object did
343 not support those attributes. The attributes that populate the Job object are persistently stored with the
344 Job object for that Job. A Get-Job-Attributes operation on that Job object will return only those
345 attributes that are persistently stored with the Job object.

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

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

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

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

367 **3.1.2.3.5 Return one of the success status codes**

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

- 369 1. the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes
370 or values.
- 371 2. the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template
372 attribute or values.

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

375

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

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

382 **3.1.2.3.6 Accept appended Document Content**

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

385 **3.1.2.3.7 Scheduling and Starting to Process the Job**

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

391 **3.1.2.3.8 Completing the Job**

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

397 **3.1.2.3.9 Destroying the Job after completion**

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

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

405 3.1.2.3.10 Interaction with "ipp-attribute-fidelity"

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

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

417 3.1.2.3.11 Character set code conversion support

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

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

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

430 According to the RFC2911 document (section 3.1.4.2), the value of the "attributes-charset" for the
431 response of the second request must be "us-ascii" since that is the charset specified in the request. The
432 "job-name" value, however, is in "utf-8" format. Should the request be rejected even though both "utf-
433 8" and "us-ascii" charsets are supported by the server? or should the "job-name" value be converted to
434 "us-ascii" and return "successful-ok-conflicting-attributes" (0x0002) as the status code?

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

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

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

448 **3.1.2.3.12 What charset to return when an unsupported charset is requested (Issue 1.19)?**

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

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

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

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

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

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

466 **3.1.2.3.13 Natural Language Override (NLO)**

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

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

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

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

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

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

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

510 **3.1.3 Status codes returned by operation**

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

514 **3.1.3.1 Printer Operations**

515 **3.1.3.1.1 Print-Job**

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

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

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

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

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

531
532 [RFC2911] section 3.1.6 Operation Status Codes and Messages states:

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

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

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

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

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

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

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

551 client-error-timeout: not applicable.

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

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

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

556 client-error-request-value-too-long: the size of request variable length attribute values, such as
557 'text' and 'name' attribute syntax's, exceed the maximum length specified in [RFC2911] for the
558 attribute and MUST be returned in the Unsupported Attributes Group.

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

563 client-error-attributes-or-values-not-supported: one or more supplied attributes, attribute
564 syntax's, or values are not supported and the client supplied the "ipp-attributes-fidelity"
565 operation attribute with a 'true' value. They MUST be returned in the Unsupported
566 Attributes Group as explained below.

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

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

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

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

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

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

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

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

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

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

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

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

592 3.1.3.1.2 Print-URI

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

596 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation
597 attribute is not supported and is returned in the Unsupported Attributes group.
598 server-error-operation-not-supported: the Print-URI operation is not supported.
599

600 3.1.3.1.3 Validate-Job

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

603 3.1.3.1.4 Create-Job

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

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

611 3.1.3.1.5 Get-Printer-Attributes

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

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

616 successful-ok: no operation attributes or values were substituted or ignored (same as Print-Job) and
617 no requested attributes were unsupported.
618 successful-ok-ignored-or-substituted-attributes: The "requested-attributes" operation attribute
619 MAY, but NEED NOT, be returned with the unsupported values.
620 successful-ok-conflicting-attributes: same as Print-Job.

621
622 For the error status codes, Group 3 is returned containing no attributes or is not returned at all:

623 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any
624 requests.
625 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.
626 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation
627 attributes and/or values MUST be ignored and an appropriate success code returned (see above).
628 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not
629 involved.
630 server-error-operation-not-supported: not applicable (since Get-Printer-Attributes is REQUIRED).
631 server-error-device-error: same as Print-Job, except that no document data is involved.

632 server-error-temporary-error: same as Print-Job, except that no document data is involved.
633 server-error-not-accepting-jobs: not applicable.
634 server-error-busy: same as Print-Job, except the IPP object is too busy to accept even query
635 requests.
636 server-error-job-canceled: not applicable.

637 **3.1.3.1.6 Get-Jobs**

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

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

642 successful-ok: same as Get-Printer-Attributes (see section 3.1.3.1.5).
643 successful-ok-ignored-or-substituted-attributes: same as Get-Printer-Attributes (see section
644 3.1.3.1.5).
645 successful-ok-conflicting-attributes: same as Get-Printer-Attributes (see section 3.1.3.1.5).
646

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

650 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any
651 requests.
652 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.
653 client-error-document-format-not-supported: not applicable.
654 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation
655 attributes and/or values MUST be ignored and an appropriate success code returned (see
656 above).
657 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not
658 involved.
659 server-error-operation-not-supported: not applicable (since Get-Jobs is REQUIRED).
660 server-error-device-error: same as Print-Job, except that no document data is involved.
661 server-error-temporary-error: same as Print-Job, except that no document data is involved.
662 server-error-not-accepting-jobs: not applicable.
663 server-error-job-canceled: not applicable.

664 **3.1.3.1.7 Pause-Printer**

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

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

670 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
671 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
672 successful-ok-conflicting-attributes: same as Print-Job.

673

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

676 client-error-not-possible: not applicable.
677 client-error-not-found: the target Printer object does not exist.
678 client-error-gone: the target Printer object no longer exists and no forwarding address is known.
679 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
680 client-error-document-format-not-supported: not applicable.
681 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-
682 accepting-jobs" attribute is not involved.
683 server-error-operation-not-supported: the Pause-Printer operation is not supported.
684 server-error-device-error: not applicable.
685 server-error-temporary-error: same as Print-Job, except no document data is involved.
686 server-error-not-accepting-jobs: not applicable.
687 server-error-job-canceled: not applicable.

688 **3.1.3.1.8 Resume-Printer**

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

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

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

696

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

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

699

700 **3.1.3.1.8.1 What about Printers unable to change state due to an error condition?**

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

705 The Resume-Printer operation must clear the 'paused' or 'moving-to-paused' 'printer-state-message'.
706 The operation must return a 'successful-ok' status code.

707 **3.1.3.1.8.2** **How is "printer-state" handled on Resume-Printer?**

708

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

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

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

- 713 1. 'idle' - no additional jobs and no error conditions present
- 714 2. 'processing' - job available and no error conditions present
- 715 3. current state (i.e. no change) an error condition is present (e.g. media jam)

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

719 **3.1.3.1.9** **Purge-Printer**

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

723 For the following success status codes, the Printer object purges all it's jobs.

- 724 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
- 725 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
- 726 successful-ok-conflicting-attributes: same as Print-Job.

727

728 For any of the error status codes, the Printer object does not purge any jobs.

729 server-error-operation-not-supported: the Purge-Printer operation is not supported.

730 **3.1.3.2** **Job Operations**

731 **3.1.3.2.1** **Send-Document**

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

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

737 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
738 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
739 successful-ok-conflicting-attributes: same as Print-Job.

740

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

743 client-error-not-possible: Same as Print-Job, except that the Printer's "printer-is-accepting-jobs"
744 attribute is not involved, so that the client is able to finish submitting a job that was created
745 with a Create-Job operation after this attribute has been set to 'true'. Another condition is
746 that the state of the job precludes Send-Document, i.e., the job has already been closed out
747 by the client. However, if the IPP Printer closed out the job due to timeout, the 'client-error-
748 timeout' error status SHOULD be returned instead.

749 client-error-timeout: This request was sent after the Printer closed the job, because it has not
750 received a Send-Document or Send-URI operation within the Printer's "multiple-operation-
751 time-out" period .

752 client-error-request-entity-too-large: same as Print-Job.

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

755 server-error-operation-not-supported: the Send-Document request is not supported.

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

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

759 **3.1.3.2.2 Send-URI**

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

763 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri"
764 operation attribute is not supported and the "document-uri" attribute MUST be returned in
765 the Unsupported Attributes group.

766 server-error-operation-not-supported: the Send-URI operation is not supported.

767

768 **3.1.3.2.3 Cancel-Job**

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

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

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

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

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

776

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

778 client-error-not-possible: The request cannot be carried out because of the state of the Job
779 object ('completed', 'canceled', or 'aborted') or the state of the system.

780 client-error-not-found: the target Printer and/or Job object does not exist.

781 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding
782 address is known.

783 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.

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

785 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation
786 attributes and values MUST be ignored.

787 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-
788 accepting-jobs" attribute is not involved.

789 server-error-operation-not-supported: not applicable (Cancel-Job is REQUIRED).

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

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

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

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

794 **3.1.3.2.4 Get-Job-Attributes**

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

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

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

800 successful-ok-ignored-or-substituted-attributes: same as Get-Printer-Attributes (see section
801 3.1.3.1.5).

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

803

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

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

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

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

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

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

813 client-error-conflicting-attributes: not applicable

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

815 server-error-device-error: same as Print-Job, except no document data is involved.
816 server-error-temporary-error: sane as Print-Job, except no document data is involved..
817 server-error-not-accepting-jobs: not applicable.
818 server-error-job-canceled: not applicable.

819 3.1.3.2.5 Hold-Job

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

823 For the following success status codes, the Job object is being held or has been held:

824 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
825 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
826 successful-ok-conflicting-attributes: same as Print-Job.

827

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

829 client-error-not-possible: The request cannot be carried out because of the state of the Job
830 object ('completed', 'canceled', or 'aborted') or the state of the system.
831 client-error-not-found: the target Printer and/or Job object does not exist.
832 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding
833 address is known.
834 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
835 client-error-document-format-not-supported: not applicable.
836 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-
837 accepting-jobs" attribute is not involved.
838 server-error-operation-not-supported: the Hold-Job operation is not supported.
839 server-error-device-error: not applicable.
840 server-error-temporary-error: same as Print-Job, except no document data is involved.
841 server-error-not-accepting-jobs: not applicable.
842 server-error-job-canceled: not applicable.

843 3.1.3.2.6 Release-Job

844 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
845 specialization's described for Hold-Job are applicable to Release-Job. See Section 13 in [RFC2911] for
846 a more complete description of each status code.

847 server-error-operation-not-supported: the Release-Job operation is not supported.

848 3.1.3.2.7 Restart-Job

849 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
850 specialization's described for Hold-Job are applicable to Restart-Job. See Section 13 in [RFC2911] for
851 a more complete description of each status code.

852 server-error-operation-not-supported: the Restart-Job operation is not supported.
853

854 3.1.3.2.7.1 Can documents be added to a restarted job?

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

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

864 3.1.4 Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)

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

871 3.1.5 Sending empty attribute groups

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

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

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

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

889 3.2 Printer Operations

890 3.2.1 Print-Job operation

891 3.2.1.1 Flow controlling the data portion of a Print-Job request (Issue 1.22)

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

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

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

901 3.2.1.2 Returning job-state in Print-Job response (Issue 1.30)

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

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

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

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

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

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

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

924 **3.2.2 Get-Printer-Attributes operation**

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

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

930 **3.2.3 Get-Jobs operation**

931 **3.2.3.1 Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?**

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

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

938 **3.2.3.2 Why is there a "limit" attribute in the Get-Jobs operation?**

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

948 3.2.4 Create-Job operation

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

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

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

965 3.3 Job Operations

966 3.3.1 Validate-Job

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

971 3.3.2 Restart-Job

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

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

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

981 **4 Object Attributes**

982 **4.1 Attribute Syntax's**

983 **4.1.1 The 'none' value for empty sets (Issue 1.37)**

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

992 **4.1.2 Multi-valued attributes (Issue 1.31)**

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

999 **4.1.3 Case Sensitivity in URIs (issue 1.6)**

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

006 The reason that the IPP specification does not make any restrictions on URIs, is so that implementations
007 of IPP may use off-the-shelf components that conform to the standards that define URIs, such as RFC
008 2396 and the HTTP/1.1 specifications [RFC2616]. See these specifications for rules of matching,
009 comparison, and case-sensitivity.

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

013 Example of equivalent URI's

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

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

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

017 Example of equivalent URI's using the IPP scheme

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

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

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

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

022 **4.1.4 Maximum length for xxxWithLanguage and xxxWithoutLanguage**

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

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

030 **4.2 Job Template Attributes**

031 **4.2.1 multiple-document-handling(type2 keyword)**

032 **4.2.1.1 Support of multiple document jobs**

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

041 4.3 Job Description Attributes

042 4.3.1 Getting the date and time of day

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

- 046 1. A Printer can get time from an NTP timeserver if there's one reachable on the network . See
047 RFC 1305. Also DHCP option 32 in RFC 2132 returns the IP address of the NTP server.
- 048 2. Get the date and time at startup from a human operator
- 049 3. Have an operator set the date and time using a web administrative interface
- 050 4. Get the date and time from incoming HTTP requests, though the problems of spoofing need
051 to be considered. Perhaps comparing several HTTP requests could reduce the chances of spoofing.
- 052 5. Internal date time clock battery driven.
- 053 6. Query "<http://tycho.usno.navy.mil/cgi-bin/timer.pl>"

054 4.4 Printer Description Attributes

055 4.4.1 queued-job-count (integer(0:MAX))

056 4.4.1.1 Why is "queued-job-count" RECOMMENDED (Issue 1.14)?

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

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

063 4.4.1.2 Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?

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

068 4.4.2 printer-current-time (dateTime)

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

- 071 1. an internal date time clock,
- 072 2. from the operator at startup using the console,
- 073 3. from an operator using an administrative web page,
- 074 4. from HTTP headers supplied in client requests,
- 075 5. use HTTP to query "<http://tycho.usno.navy.mil/cgi-bin/timer.pl>"
- 076 6. from the network, using NTP [RFC1305] or DHCP option 32 [RFC2132] that returns the IP
077 address of the NTP server.

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

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

083 4.4.3 Printer-uri

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

085 A forgiving printer implementation would not reject the operation. But the implementation has its rights
086 to reject a printer or job operation if the operational attribute printer-uri is not a value of the printer-uri-
087 supported. The printer might not be improperly configured. The request obviously reached the printer.
088 The printer could treat the printer-uri as the logical equivalent of a value in the printer-uri-supported. It
089 would be implementation dependent for which value, and associated security policy, would apply. This
090 does also apply to a job object specified with a printer-uri and job-id, or with a job-uri. See section 4.1.3
091 for how to compare URI's.

092 4.5 Empty Jobs

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

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

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

104 5 Directory Considerations

105 5.1 General Directory Schema Considerations

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

108 The SLP printer template is defined in the "Definition of the Printer Abstract Service Type v2.0"
109 document [svrloc-printer]. The LDAP printer template is defined in the "Internet Printing Protocol
110 (IPP): LDAP Schema for Printer Services" document [ldap-printer]. Both documents systematically
111 add "printer-" to any attribute that doesn't already start with "printer-" in order to keep the printer
112 directory attributes distinct from other directory attributes. Also, instead of using "printer-uri-
113 supported", "uri-authentication-supported", and "uri-security-supported", they use a "printer-xri-
114 supported" attribute with special syntax to contain all of the same information in a single attribute.

115 5.2 IPP Printer with a DNS name

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

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

119 6 Security Considerations

120 The security considerations given in [RFC2911] Section 8 "Security Considerations" all apply to this
121 document. In addition, the following sub-sections describes security consideration that have arisen as a
122 result of implementation testing. ~~This section corresponds to the RFC2911 Section 8 "Security~~
123 ~~Considerations.~~

124 6.1 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)

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

126 8.5 Queries on jobs submitted using non-IPP protocols
127 If the device that an IPP Printer is representing is able to accept jobs using other job submission
128 protocols in addition to IPP, it is RECOMMEND that such an implementation at least allow such
129 "foreign" jobs to be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an
130 implementation NEED NOT support all of the same IPP job attributes as for IPP jobs. The IPP
131 object returns the 'unknown' out-of-band value for any requested attribute of a foreign job that is
132 supported for IPP jobs, but not for foreign jobs.

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

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

145 7 Encoding and Transport

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

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

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

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

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

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

- 174 - the "header" column contains the name of a header
- 175 - the "request/client" column indicates whether a client sends the header.
- 176 - the "request/ server" column indicates whether a server supports the header when received.
- 177 - the "response/ server" column indicates whether a server sends the header.
- 178 - the "response /client" column indicates whether a client supports the header when received.
- 179 - the "values and conditions" column specifies the allowed header values and the conditions for the
180 header to be present in a request/response.

181
182 The table for "request headers" does not have columns for responses, and the table for "response
183 headers" does not have columns for requests.

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

- 185 - **must:** the client or server **MUST** send the header,
- 186 - **must-if:** the client or server **MUST** send the header when the condition described in the "values
187 and conditions" column is met,
- 188 - **may:** the client or server **MAY** send the header
- 189 - **not:** the client or server **SHOULD NOT** send the header. It is not relevant to an IPP
190 implementation.

191
192 The following is an explanation of the values in the "response/client" and "request/ server" columns.

- 193 - **must:** the client or server **MUST** support the header,
- 194 - **may:** the client or server **MAY** support the header
- 195 - **not:** the client or server **SHOULD NOT** support the header. It is not relevant to an IPP
196 implementation.

197 7.1 General Headers

198 The following is a table for the general headers.

General-Header	Request		Response		Values and Conditions
	Client	Server	Server	Client	
Cache-Control	must	not	must	not	"no-cache" only
Connection	must-if	must	must-if	must	"close" only. Both client and server SHOULD keep a connection for the duration of a sequence of operations. The client and server MUST include this header for the last operation in such a sequence.
Date	may	may	must	may	per RFC 1123 [RFC1123] from RFC 2616 [RFC2616]
Pragma	must	not	must	not	"no-cache" only
Transfer-Encoding	must-if	must	must-if	must	"chunked" only . Header MUST be present if Content-Length is absent.
Upgrade	not	not	not	not	
Via	not	not	not	not	

199 **7.2 Request Headers**

200 The following is a table for the request headers.

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

Request-Header	Client	Server	Request Values and Conditions
Accept	may	must	"application/ipp" only. This value is the default if the client omits it
Accept-Charset	not	not	Charset information is within the application/ipp entity
Accept-Encoding	may	must	empty and per RFC 2616 [RFC2616] and IANA registry for content-codings
Accept-Language	not	not	language information is within the application/ipp entity
Authorization	must-if	must	per RFC 2616. A client MUST send this header when it receives a 401 "Unauthorized" response and does not receive a "Proxy-Authenticate" header.
From	not	not	per RFC 2616. Because RFC recommends sending this header only with the user's approval, it is not very useful
Host	must	must	per RFC 2616
If-Match	not	not	
If-Modified-Since	not	not	
If-None-Match	not	not	
If-Range	not	not	
If-Unmodified-Since	not	not	
Max-Forwards	not	not	
Proxy- Authorization	must-if	not	per RFC 2616. A client MUST send this header when it receives a 401 "Unauthorized" response and a "Proxy-Authenticate" header.
Range	not	not	
Referrer	not	not	
User-Agent	not	not	

201 7.3 Response Headers

202 The following is a table for the request headers.

Response-Header	Server	Client	Response Values and Conditions
Accept-Ranges	not	not	
Age	not	not	
Location	must-if	may	per RFC 2616. When URI needs redirection.
Proxy-Authenticate	not	must	per RFC 2616
Public	may	may	per RFC 2616
Retry-After	may	may	per RFC 2616
Server	not	not	
Vary	not	not	
Warning	may	may	per RFC 2616
WWW-Authenticate	must-if	must	per RFC 2616. When a server needs to authenticate a client.

203 7.4 Entity Headers

204 The following is a table for the entity headers.

Entity-Header	Request		Response		Values and Conditions
	Client	Server	Server	Client	
Allow	not	not	not	not	
Content-Base	not	not	not	not	
Content-Encoding	may	must	must	must	per RFC 2616 and IANA registry for content codings.
Content-Language	not	not	not	not	Application/ipp handles language
Content-Length	must-if	must	must-if	must	the length of the message-body per RFC 2616. Header MUST be present if Transfer-Encoding is absent..
Content-Location	not	not	not	not	
Content-MD5	may	may	may	may	per RFC 2616
Content-Range	not	not	not	not	
Content-Type	must	must	must	must	"application/ipp" only
ETag	not	not	not	not	
Expires	not	not	not	not	
Last-Modified	not	not	not	not	

205 7.5 Optional support for HTTP/1.0

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

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

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

217 **7.6 HTTP/1.1 Chunking**

218 **7.6.1 Disabling IPP Server Response Chunking**

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

223 TE: identity

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

226 **7.6.2 Warning About the Support of Chunked Requests**

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

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

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

240 **8 References**

241 [CGI]

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

- 243 [\[IANA-CS\]](#)
244 [IANA Registry of Coded Character Sets: \[ftp://ftp.iana.org/in-notes/iana/assignments/character-sets\]\(http://ftp.iana.org/in-notes/iana/assignments/character-sets\)](http://ftp.iana.org/in-notes/iana/assignments/character-sets)
- 245 [ldap-printer]
246 Fleming, P., Jones, K., Lewis, H., McDonald, I., "Internet Printing Protocol (IPP): LDAP Schema
247 for Printer Services", <draft-ietf-ipp-ldap-printer-schema-04.txt>, work in progress, [April](#)
248 [27December 20](#), 2000.
- 249 [RFC793]
250 J. Postel, "Transmission Control Protocol", RFC 793.
- 251 [RFC1123]
252 Braden, S., "Requirements for Internet Hosts - Application and Support", RFC 1123, October, 1989.
- 253 [RFC2026]
254 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 255 [RFC2119]
256 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March
257 1997.
- 258 [RFC2396]
259 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic
260 Syntax", RFC 2396, August 1998.
- 261 [RFC2565]
262 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
263 Semantics", RFC 2566, April 1999.
- 264 [RFC2566]
265 Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.0: Encoding and
266 Transport", RFC 2565, April 1999.
- 267 [RFC2567]
268 Wright, D., "Design Goals for an Internet Printing Protocol", [draft-ietf-ipp-req-03.txt](#) [RFC 2567](#),
269 [November, 1998](#) [April 1999](#).
- 270 [RFC2568]
271 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
272 RFC 2568, April 1999.
- 273 [RFC2569]
274 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC
275 2569, April 1999.
- 276 [RFC2616]

- 277 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext
278 Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
- 279 [RFC2910]
280 Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.0: Encoding and
281 Transport", RFC 2910, September, 2000.
- 282 [RFC2911]
283 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
284 Semantics", RFC 2911, September, 2000.
- 285 [Servlet]
286 Servlet Specification Version 2.1 (<http://java.sun.com/products/servlet/2.1/index.html>).
- 287 [svrloc-printer]
288 St. Pierre, P., Isaacson, S., McDonald, I., "Definition of the Printer Abstract Service Type v2.0",
289 <http://www.isi.edu/in-notes/iana/assignments/svrloc-templates/printer.2.0.en> (IANA Registered, May
290 [27, 2000](#))<~~draft-ietf-svrloc-printer-scheme-06.txt~~>, work in progress, March 8, 2000.
- 291 [SSL]
292 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.

293 9 Authors' Address

294
295 Thomas N. Hastings
296 Xerox Corporation
297 701 Aviation Blvd.
298 El Segundo, CA 90245
299 hastings@cp10.es.xerox.com
300

301 Carl-Uno Manros
302 Xerox Corporation
303 701 Aviation Blvd.
304 El Segundo, CA 90245
305 cmanros@cp10.es.xerox.com
306

307 Carl Kugler
308 Mail Stop 003G
309 IBM Printing Systems Co
310 6300 Diagonal Hwy
311 Boulder CO 80301
312 Kugler@us.ibm.com
313

314 Henrik Holst

315 i-data Printing Systems
316 Vadstrupvej 35-43
317 2880 Bagsvaerd, Denmark
318 hh@I-data.com

319
320 Peter Zehler
321 Xerox Corporation
322 800 Philips Road
323 Webster, NY 14580
324 peter.zehler@usa.xerox.com

325
326 IPP Web Page: <http://www.pwg.org/ipp/>
327 IPP Mailing List: ipp@pwg.org

328
329 To subscribe to the ipp mailing list, send the following email:

330 1) send it to majordomo@pwg.org
331 2) leave the subject line blank
332 3) put the following two lines in the message body:
333 subscribe ipp
334 end

335
336 Implementers of this specification document are encouraged to join the IPP Mailing List in order to
337 participate in any discussions of clarification issues and review of registration proposals for additional
338 attributes and values. In order to reduce spam the mailing list rejects mail from non-subscribers, so you
339 must subscribe to the mailing list in order to send a question or comment to the mailing list.

340
341 Other Participants:

Chuck Adams - Tektronix	Shivaun Albright - HP
Stefan Andersson - Axis	Jeff Barnett - IBM
Ron Bergman - Hitachi Koki Imaging Systems	Dennis Carney - IBM
Keith Carter - IBM	Angelo Caruso - Xerox
Rajesh Chawla - TR Computing Solutions	Nancy Chen - Okidata
Josh Cohen - Microsoft	Jeff Copeland - QMS
Andy Davidson - Tektronix	Roger deBry - IBM
Maulik Desai - Auco	Mabry Dozier - QMS
Lee Farrell - Canon Information Systems	Satoshi Fujitami - Ricoh
Steve Gebert - IBM	Sue Gleeson - Digital
Charles Gordon - Osicom	Brian Grimshaw - Apple
Jerry Hadsell - IBM	Richard Hart - Digital
Tom Hastings - Xerox	Henrik Holst - I-data
Stephen Holmstead	Zhi-Hong Huang - Zenographics
Scott Isaacson - Novell	Babek Jahromi - Microsoft
Sven Johnson - Xerox	David Kellerman - Northlake Software
Robert Kline - TrueSpectra	Charles Kong - Panasonic
Carl Kugler - IBM	Dave Kuntz - Hewlett-Packard
Takami Kurono - Brother	Rick Landau - Digital
Scott Lawrence - Agranot Systems	Greg LeClair - Epson
Dwight Lewis - Lexmark	Harry Lewis - IBM
Tony Liao - Vivid Image	Roy Lomicka - Digital
Pete Loya - HP	Ray Lutz - Cognisys
Mike MacKay - Novell, Inc.	David Manchala - Xerox
Carl-Uno Manros - Xerox	Jay Martin - Underscore
Stan McConnell - Xerox	Larry Masinter - Xerox
Sandra Matts - Hewlett Packard	Peter Michalek - Shinesoft
Ira McDonald - High North Inc.	Mike Moldovan - G3 Nova
Tetsuya Morita - Ricoh	Yuichi Niwa - Ricoh
Pat Nogay - IBM	Ron Norton - Printronics
Hugo Parra, Novell	Bob Pentecost - Hewlett-Packard
Patrick Powell - Astart Technologies	Jeff Rackowitz - Intermec
Eric Random - Peerless	Rob Rhoads - Intel
Xavier Riley - Xerox	Gary Roberts - Ricoh
David Roach - Unisys	Stuart Rowley - Kyocera
Yuji Sasaki - Japan Computer Industry	Richard Schneider - Epson
Kris Schoff - HP	Katsuaki Sekiguchi - Canon
Bob Setterbo - Adobe	Gail Songer - Peerless
Hideki Tanaka - Canon	Devon Taylor - Novell, Inc.
Mike Timperman - Lexmark	Atsushi Uchino - Epson
Shigeru Ueda - Canon	Bob Von Andel - Allegro Software
William Wagner - NetSilicon/DPI	Jim Walker - DAZEL
Chris Wellens - Interworking Labs	Trevor Wells - Hewlett Packard

Craig Whittle - Sharp Labs	Rob Whittle - Novell, Inc.
Jasper Wong - Xionics	Don Wright - Lexmark
Michael Wu - Heidelberg Digital	Rick Yardumian - Xerox
Michael Yeung - Toshiba	Lloyd Young - Lexmark
Atsushi Yuki - Kyocera	Peter Zehler - Xerox
William Zhang- Canon Information Systems	Frank Zhao - Panasonic
Steve Zilles - Adobe	Rob Zirnstein - Canon Information Systems

342

343

344 10 Description of the Base IPP Documents

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

346 Design Goals for an Internet Printing Protocol [RFC2567]

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

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

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

350 Mapping between LPD and IPP Protocols [RFC2569]

351

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

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

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

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

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

373 **11 Full Copyright Statement**

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

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

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

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

391 **Acknowledgement**

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