

1 INTERNET-DRAFT ~~return job attributes in Send Document and Send URI needs to be indicated~~
2 draft-ietf-ipp-implementers-guide-v11-010.txt

3
4 T. Hastings
Xerox Corporation
5 C. Manros
6 Xerox Corporation
7 C. Kugler
8 IBM Printing Systems Co
9 H. Holst
10 i-data Printing Systems
11 P.Zehler
12 Xerox Corporation
13 ~~September~~ May 927, 19992000
14

15 Internet Printing Protocol/1.1: Implementer's Guide

16 Copyright (C) The Internet Society (~~1999~~2000). All Rights Reserved.

17 Status of this Memo

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

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

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

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

27
28 Abstract

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

37 The full set of IPP documents includes:

38 Design Goals for an Internet Printing Protocol [RFC2567]

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

40 Internet Printing Protocol/1.1: Model and Semantics [IPP-MOD]

41 Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]

42 Mapping between LPD and IPP Protocols [RFC2569]

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

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

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

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

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

60

61 TABLE OF CONTENTS

| | | |
|-----|---|----------|
| 62 | 1 INTRODUCTION | 6 |
| 63 | 1.1 CONFORMANCE LANGUAGE..... | 6 |
| 64 | 1.2 OTHER TERMINOLOGY..... | 6 |
| 65 | 1.3 ISSUES RAISED FROM INTEROPERABILITY BAKE OFFS..... | 6 |
| 66 | 2 IPP OBJECTS | 7 |
| 67 | 3 IPP OPERATIONS | 8 |
| 68 | 3.1 COMMON SEMANTICS..... | 8 |
| 69 | 3.1.1 <i>Summary of Operation Attributes</i> | 8 |
| 70 | 3.1.2 <i>Suggested Operation Processing Steps for IPP Objects</i> | 12 |
| 71 | 3.1.2.1 Suggested Operation Processing Steps for all Operations..... | 13 |
| 72 | 3.1.2.1.1 Validate version number..... | 13 |
| 73 | 3.1.2.1.2 Validate operation identifier..... | 14 |
| 74 | 3.1.2.1.3 Validate the request identifier..... | 15 |
| 75 | 3.1.2.1.4 Validate attribute group and attribute presence and order..... | 15 |
| 76 | 3.1.2.1.4.1 Validate the presence and order of attribute groups..... | 15 |
| 77 | 3.1.2.1.4.2 Ignore unknown attribute groups in the expected position..... | 15 |
| 78 | 3.1.2.1.4.3 Validate the presence of a single occurrence of required Operation attributes..... | 16 |
| 79 | 3.1.2.1.5 Validate the values of the REQUIRED Operation attributes..... | 22 |
| 80 | 3.1.2.1.6 Validate the values of the OPTIONAL Operation attributes..... | 25 |
| 81 | 3.1.2.2 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add Documents..... | 28 |
| 82 | 3.1.2.2.1 Default "ipp-attribute-fidelity" if not supplied..... | 28 |
| 83 | 3.1.2.2.2 Check that the Printer object is accepting jobs..... | 28 |
| 84 | 3.1.2.2.3 Validate the values of the Job Template attributes..... | 29 |
| 85 | 3.1.2.3 Algorithm for job validation..... | 29 |
| 86 | 3.1.2.3.1 Check for conflicting Job Template attributes values..... | 34 |
| 87 | 3.1.2.3.2 Decide whether to REJECT the request..... | 34 |
| 88 | 3.1.2.3.3 For the Validate-Job operation, RETURN one of the success status codes..... | 35 |
| 89 | 3.1.2.3.4 Create the Job object with attributes to support..... | 36 |
| 90 | 3.1.2.3.5 Return one of the success status codes..... | 37 |
| 91 | 3.1.2.3.6 Accept appended Document Content..... | 37 |
| 92 | 3.1.2.3.7 Scheduling and Starting to Process the Job..... | 37 |
| 93 | 3.1.2.3.8 Completing the Job..... | 38 |
| 94 | 3.1.2.3.9 Destroying the Job after completion..... | 38 |
| 95 | 3.1.2.3.10 Interaction with "ipp-attribute-fidelity"..... | 38 |
| 96 | 3.1.2.3.11 Character set code conversion support..... | 38 |
| 97 | 3.1.2.3.12 What charset to return when an unsupported charset is requested (Issue 1.19)?..... | 39 |
| 98 | 3.1.2.3.13 Natural Language Override (NLO)..... | 40 |
| 99 | 3.1.3 <i>Status codes returned by operation</i> | 41 |
| 100 | 3.1.3.1 Printer Operations..... | 41 |
| 101 | 3.1.3.1.1 Print-Job..... | 41 |
| 102 | 3.1.3.1.2 Print-URI..... | 43 |
| 103 | 3.1.3.1.3 Validate-Job..... | 43 |
| 104 | 3.1.3.1.4 Create-Job..... | 43 |
| 105 | 3.1.3.1.5 Get-Printer-Attributes..... | 43 |
| 106 | 3.1.3.1.6 Get-Jobs..... | 44 |
| 107 | 3.1.3.1.7 Pause-Printer..... | 45 |
| 108 | 3.1.3.1.8 Resume-Printer..... | 45 |
| 109 | 3.1.3.1.8.1 What about Printers unable to change state due to an error condition?..... | 45 |
| 110 | 3.1.3.1.8.2 How is 'printer-state' handled on Resume-Printer?..... | 46 |
| 111 | 3.1.3.1.9 Purge-Printer..... | 46 |
| 112 | 3.1.3.2 Job Operations..... | 46 |
| 113 | 3.1.3.2.1 Send-Document..... | 47 |
| 114 | 3.1.3.2.2 Send-URI..... | 47 |
| 115 | 3.1.3.2.3 Cancel-Job..... | 47 |
| 116 | 3.1.3.2.4 Get-Job-Attributes..... | 48 |

| | | | |
|-----|-------------|--|----|
| 117 | 3.1.3.2.5 | Hold-Job..... | 49 |
| 118 | 3.1.3.2.6 | Release-Job..... | 49 |
| 119 | 3.1.3.2.7 | Restart-Job..... | 49 |
| 120 | 3.1.3.2.7.1 | Can documents be added to a restarted job?..... | 50 |
| 121 | 3.1.4 | Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)..... | 50 |
| 122 | 3.1.5 | Sending empty attribute groups | 50 |
| 123 | 3.2 | PRINTER OPERATIONS | 51 |
| 124 | 3.2.1 | Print-Job operation..... | 51 |
| 125 | 3.2.1.1 | Flow controlling the data portion of a Print-Job request (Issue 1.22)..... | 51 |
| 126 | 3.2.1.2 | Returning job-state in Print-Job response (Issue 1.30) | 51 |
| 127 | 3.2.2 | Get-Printer-Attributes operation | 52 |
| 128 | 3.2.3 | Get-Jobs operation | 52 |
| 129 | 3.2.3.1 | Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?..... | 52 |
| 130 | 3.2.3.2 | Why is there a "limit" attribute in the Get-Jobs operation?..... | 52 |
| 131 | 3.2.4 | Create-Job operation..... | 52 |
| 132 | 3.3 | JOB OPERATIONS..... | 53 |
| 133 | 3.3.1 | Validate-Job..... | 53 |
| 134 | 3.3.2 | Restart-Job..... | 53 |
| 135 | 4 | OBJECT ATTRIBUTES | 53 |
| 136 | 4.1 | ATTRIBUTE SYNTAX'S | 54 |
| 137 | 4.1.1 | The 'none' value for empty sets (Issue 1.37) | 54 |
| 138 | 4.1.2 | Multi-valued attributes (Issue 1.31)..... | 54 |
| 139 | 4.1.3 | Case Sensitivity in URIs (issue 1.6) | 54 |
| 140 | 4.1.4 | Maximum length for xxxWithLanguage and xxxWithoutLanguage | 55 |
| 141 | 4.2 | JOB TEMPLATE ATTRIBUTES | 55 |
| 142 | 4.2.1 | multiple-document-handling(type2 keyword)..... | 55 |
| 143 | 4.2.1.1 | Support of multiple document jobs..... | 55 |
| 144 | 4.3 | JOB DESCRIPTION ATTRIBUTES | 55 |
| 145 | 4.4 | PRINTER DESCRIPTION ATTRIBUTES..... | 56 |
| 146 | 4.4.1 | queued-job-count..... | 56 |
| 147 | 4.4.1.1 | Why is "queued-job-count" RECOMMENDED (Issue 1.14)? | 56 |
| 148 | 4.4.1.2 | Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)? | 56 |
| 149 | 4.4.2 | printer-current-time (dateTime)..... | 56 |
| 150 | 4.4.3 | 'Printer-uri'..... | 57 |
| 151 | 4.5 | EMPTY JOBS..... | 57 |
| 152 | 5 | DNS CONSIDERATIONS..... | 58 |
| 153 | 5.1 | IPP PRINTER WITH A DNS NAME | 58 |
| 154 | 6 | SECURITY CONSIDERATIONS | 58 |
| 155 | 6.1 | QUERYING JOBS WITH IPP THAT WERE SUBMITTED USING OTHER JOB SUBMISSION PROTOCOLS (ISSUE 1.32) | 58 |
| 156 | 7 | ENCODING AND TRANSPORT | 58 |
| 157 | 7.1 | GENERAL HEADERS | 60 |
| 158 | 7.2 | REQUEST HEADERS | 60 |
| 159 | 7.3 | RESPONSE HEADERS | 61 |
| 160 | 7.4 | ENTITY HEADERS | 62 |
| 161 | 7.5 | OPTIONAL SUPPORT FOR HTTP/1.0..... | 63 |
| 162 | 7.6 | HTTP/1.1 CHUNKING | 63 |
| 163 | 7.6.1 | Disabling IPP Server Response Chunking..... | 63 |
| 164 | 7.6.2 | Warning About the Support of Chunked Requests..... | 63 |
| 165 | 8 | REFERENCES..... | 64 |
| 166 | 8.1 | AUTHORS' ADDRESS..... | 65 |

167 **9 NOTICES**.....66

168 **10 CHANGE HISTORY (TO BE REMOVED AT TIME OF RFC PUBLISHING)**.....67

169 10.1 CHANGES FROM 990914 TO 99092767

170 10.2 CHANGES FROM 990726 TO 990914:67

171 10.3 CHANGES TO PRODUCE THE FEBRUARY 12, 1999 VERSION FROM THE JANUARY 8, 1999 VERSION:67

172 10.4 CHANGES TO PRODUCE THE JANUARY 8, 1999 VERSION FROM THE DECEMBER 6, 1998 VERSION:.....68

173 10.5 CHANGES TO PRODUCE THE DECEMBER 6, 1998 VERSION FROM THE NOVEMBER 16, 1998 VERSION:68

174

175 **TABLES**

176

177 Table 1 - Summary of Printer operation attributes that sender **MUST** supply..... 8

178 Table 2 - Summary of Printer operation attributes that sender **MAY** supply 9

179 Table 3 - Summary of Job operation attributes that sender **MUST** supply..... 10

180 Table 4 - Summary of Job operation attributes that sender **MAY** supply..... 11

181 Table 5 - Printer operation response attributes 11

182 Table 6 - Examples of validating IPP version..... 14

183 Table 7 - Rules for validating single values X against Z 30

184

185
186
187

188 **1 Introduction**

189 The IPP Implementer's Guide (IIG) (this document) contains information that supplements the IPP Model
190 and Semantics [IPP-MOD] and the IPP Transport and Encoding [IPP-PRO] documents. As such this
191 information is not part of the formal specifications. Instead information is presented to help implementers
192 understand the specification, including some of the motivation for decisions taken by the committee in
193 developing the specification. Some of the implementation considerations are intended to help
194 implementers design their client and/or IPP object implementations. If there are any contradictions between
195 this document and [IPP-MOD] or [IPP-PRO], those documents take precedence over this document.

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

197 In order to help the reader of the IIG and the IPP Model and Semantics document, the sections in this
198 document parallel the corresponding sections in the Model document and are numbered the same for ease
199 of cross reference. The sections that correspond to the IPP Transport and Encoding are correspondingly
200 offset.

201 1.1 Conformance language

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

207 Implementers should read section 13 (APPENDIX A) in [IPP-MOD] in order to understand these
208 capitalized words. The words MUST, MUST NOT, and REQUIRED indicate what implementations are
209 required to support in a client or IPP object in order to be conformant to [IPP-MOD] and [IPP-PRO].
210 MAY, NEED NOT, and OPTIONAL indicate was is merely allowed as an implementer option. The verbs
211 SHOULD and SHOULD NOT indicate suggested behavior, but which is not required or disallowed,
212 respectively, in order to conform to the specification.

213 1.2 Other terminology

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

216 1.3 Issues Raised from Interoperability Bake Offs

217 The IPP WG has conducted two open interoperability "Bake Offs". The first bake off was held in
218 September 1998 and Bake Off2 was held in March 1999. See the summary reports in:

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

220 The issues raised from the first bake off are numbered 1.n in this document and are described in:

221 <ftp://ftp.pwg.org/pub/pwg/ipp/approved-clarifications/ipp-agreed-fixes-981030.pdf>

222 These issue resolutions have been incorporated into the November 16, "IPP/1.0 Model and Semantics" [ipp-
223 mod] and the "IPP/1.0 Encoding and Transport" [~~ipp-pr~~ IPP-PRO] documents. However, some of the
224 discussion is left here in the Implementer's Guide to help understanding.

225 The issues raised from Bake Off2 are numbered 2.n in this document and are described in:

226 <ftp://ftp.pwg.org/pub/pwg/ipp/issues/issues-raised-at-bake-off2.pdf>

227 **2 IPP Objects**

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

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

235 **3 IPP Operations**

236 This section corresponds to Section 3 "IPP Operations" in the IPP/1.1 Model and Semantics document
 237 [IPP-MOD].

238 3.1 Common Semantics

239 This section discusses semantics common to all operations.

240 3.1.1 Summary of Operation Attributes

241 Legend for the following table:

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

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

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

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

| Operation Attributes | Printer Operations | | | | | | Responses |
|--|-----------------------------|---------------|----------------|----------------------------|--------------|---|----------------|
| | Requests | | | | | | |
| | Print-Job, Validate-Job (R) | Print-URI (O) | Create-Job (O) | Get-Printer-Attributes (R) | Get-Jobs (R) | Pause-Printer, Resume-Printer, Purge-Printer (O+) | All Operations |
| 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 | |

248

249

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

| Operation Attributes | Printer Operations | | | | | | Responses |
|---|-----------------------------|---------------|----------------|----------------------------|--------------|---|----------------|
| | Requests | | | | | | |
| | Print-Job, Validate-Job (R) | Print-URI (O) | Create-Job (O) | Get-Printer-Attributes (R) | Get-Jobs (R) | Pause-Printer, Resume-Printer, Purge-Printer (O+) | All Operations |
| Operation attributes--OPTIONAL to be supplied by the sender | | | | | | | |
| status-message | | | | | | | O |
| detailed-status-message | | | | | | | O |
| compression | O | O | | | | | |
| document-format | R | R | | O | | | |
| 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 | | |

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

250

251

252

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

| Operation Attributes | Job Operations | | | | | |
|--|-------------------|--------------|----------------|------------------------|---|----------------|
| | Requests | | | | | Responses |
| | Send-Document (O) | Send-URI (O) | Cancel-Job (R) | Get-Job-Attributes (R) | Hold-Job, Release-Job, Restart-Job (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 | |

253

254

255

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

| Operation Attributes | Job Operations | | | | | | |
|---|-------------------|--------------|----------------|------------------------|----------------------------|------------------|----------------|
| | Requests | | | | | | Responses |
| | Send-Document (O) | Send-URI (O) | Cancel-Job (R) | Get-Job-Attributes (R) | Hold-Job, Restart-Job (O+) | Release-Job (O+) | All Operations |
| Operation attributes--OPTIONAL to be supplied by the sender | | | | | | | |
| status-message | | | | | | | O |
| detailed-status-message | | | | | | | O |
| document-access-error | | | | | | | O** |
| compression | O | 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 | | | | | | | |

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

** "document-access-error" applies to the Send-URI operation only.

256

257

Table 5 - Printer operation response attributes

| Operation Attributes | Printer Operations | | | | | | |
|------------------------|---|------------------|---------------|----------------|----------------------------|--------------|---|
| | Response | | | | | | |
| | Print-Job (R), <u>Send-Document (O), Send-URI (O)</u> | Validate-Job (R) | Print-URI (O) | Create-Job (O) | Get-Printer-Attributes (R) | Get-Jobs (R) | Pause-Printer, Resume-Printer, Purge-Printer (O+) |
| job-uri | R | | R | R | | | |
| job-id | R | | R | R | | | |
| job-state | R | | R | R | | | |
| job-state-reasons | R+ | | R+ | R+ | | | |
| document-access-error+ | | | O | | | | |

258

259

260 3.1.2 Suggested Operation Processing Steps for IPP Objects (~~Issue 1.21~~)

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

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

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

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

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

287 3.1.2.1 Suggested Operation Processing Steps for all Operations

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

| 291 | IIG Sect # | Flow | IPP error status codes |
|-----|--------------|-------------------------|--|
| 292 | ----- | ---- | ----- |
| 293 | | | |
| 294 | | v | err |
| 295 | 3.1.2.1.1 | <Validate version> | --> server-error-version-not-supported |
| 296 | | ok | |
| 297 | | v | err |
| 298 | 3.1.2.1.2 | <Validate operation> | --> server-error-operation-not-supported |
| 299 | | ok | |
| 300 | | v | err |
| 301 | 3.1.2.1.4.1- | <Validate presence> | --> client-error-bad-request |
| 302 | 3.1.2.1.4.2 | <of attributes> | |
| 303 | | ok | |
| 304 | | v | err |
| 305 | 3.1.2.1.4.3 | <Validate presence> | --> client-error-bad-request |
| 306 | | <of operation attr> | |
| 307 | | ok | |
| 308 | | v | err |
| 309 | 3.1.2.1.5 | <Valied values of> | --> client-error-bad-request |
| 310 | | <operation attrs> | client-error-request-value-too-long |
| 311 | | <(length, tag, range,> | |
| 312 | | <multi-value)> | |
| 313 | | ok | |
| 314 | | v | err |
| 315 | 3.1.2.1.5 | <Validate values> | --> client-error-bad-request |
| 316 | | <with supported values> | client-error-charset-not-supported |
| 317 | | ok | client-error-attributes-or-values- |
| 318 | | | not-supported |
| 319 | | v | err |
| 320 | 3.1.2.1.6 | <Validate optionally> | --> client-error-bad-request |
| 321 | | <operation attr> | client-error-natural-language-not- |
| 322 | | ok | supported |
| 323 | | | client-error-request-value-too-long |
| 324 | | | client-error-attributes-or-values- |
| 325 | | v | not-supported |
| 326 | | | |

327 3.1.2.1.1 Validate version number

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

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

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

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

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

346

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

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

352 3.1.2.1.2 Validate operation identifier

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

356 3.1.2.1.3 Validate the request identifier

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

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

362 3.1.2.1.4 Validate attribute group and attribute presence and order

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

364 3.1.2.1.4.1 Validate the presence and order of attribute groups

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

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

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

377 3.1.2.1.4.2 Ignore unknown attribute groups in the expected position

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

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

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

390 3.1.2.1.4.3 Validate the presence of a single occurrence of required Operation attributes
391 Client requests and IPP object responses contain Operation attributes that [IPP-MOD] Section 3 requires to
392 be present. Attributes within a group may be in any order, except for the ordering of target, charset, and
393 natural languages attributes. These attributes **MUST** be first, and **MUST** be supplied in the following order:
394 charset, natural language, and then target. An IPP object verifies that the attributes that Section 4 requires to
395 be supplied by the client have been supplied in the request (attributes without an * in the following tables).
396 An asterisk (*) indicates groups and Operation attributes that the client may omit in a request or an IPP
397 object may omit in a response.

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

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

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

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

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

- 418 R indicates a **REQUIRED** attribute or operation that an IPP object **MUST** support
- 419 O indicates an **OPTIONAL** attribute or operation that an IPP object **NEED NOT** support
- 420 * indicates that a client **MAY** omit the attribute in a request and that an IPP object **MAY** omit the
421 attribute in a response. The absence of an * means that a client **MUST** supply the
422 attribute in a request and an IPP object **MUST** supply the attribute in a response.
- 423 + indicates that this is not a IPP/1.0 operation, but is only a part of IPP/1.1 and future versions of IPP.

424

425

Operation Requests

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

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

429

430 Print-Job Request (R):

431 Group 1: Operation Attributes (R)
432 attributes-charset (R)
433 attributes-natural-language (R)
434 printer-uri (R)
435 requesting-user-name (R*)
436 job-name (R*)
437 ipp-attribute-fidelity (R*)
438 document-name (R*)
439 document-format (R*)
440 document-natural-language (O*)
441 compression (O*)
442 job-k-octets (O*)
443 job-impressions (O*)
444 job-media-sheets (O*)
445 Group 2: Job Template Attributes (R*)
446 <Job Template attributes> (O*)
447 (see [IPP-MOD] Section 4.2)
448 Group 3: Document Content (R)
449 <document content>

450

451 Validate-Job Request (R):

452 Group 1: Operation Attributes (R)
453 attributes-charset (R)
454 attributes-natural-language (R)
455 printer-uri (R)
456 requesting-user-name (R*)
457 job-name (R*)
458 ipp-attribute-fidelity (R*)
459 document-name (R*)
460 document-format (R*)
461 document-natural-language (O*)
462 compression (O*)
463 job-k-octets (O*)
464 job-impressions (O*)
465 job-media-sheets (O*)
466 Group 2: Job Template Attributes (R*)
467 <Job Template attributes> (O*)
468 (see [IPP-MOD] Section 4.2)

469

470 Print-URI Request (O):

471 Group 1: Operation Attributes (R)
472 attributes-charset (R)

473 attributes-natural-language (R)
474 printer-uri (R)
475 document-uri (R)
476 requesting-user-name (R*)
477 job-name (R*)
478 ipp-attribute-fidelity (R*)
479 document-name (R*)
480 document-format (R*)
481 document-natural-language (O*)
482 compression (O*)
483 job-k-octets (O*)
484 job-impressions (O*)
485 job-media-sheets (O*)
486 Group 2: Job Template Attributes (R*)
487 <Job Template attributes> (O*) (see
488 (see [IPP-MOD] Section 4.2)
489
490 Create-Job Request (O):
491 Group 1: Operation Attributes (R)
492 attributes-charset (R)
493 attributes-natural-language (R)
494 printer-uri (R)
495 requesting-user-name (R*)
496 job-name (R*)
497 ipp-attribute-fidelity (R*)
498 job-k-octets (O*)
499 job-impressions (O*)
500 job-media-sheets (O*)
501 Group 2: Job Template Attributes (R*)
502 <Job Template attributes> (O*) (see
503 (see [IPP-MOD] Section 4.2)
504
505 Get-Printer-Attributes Request (R):
506 Group 1: Operation Attributes (R)
507 attributes-charset (R)
508 attributes-natural-language (R)
509 printer-uri (R)
510 requesting-user-name (R*)
511 requested-attributes (R*)
512 document-format (R*)
513
514 Get-Jobs Request (R):
515 Group 1: Operation Attributes (R)
516 attributes-charset (R)
517 attributes-natural-language (R)
518 printer-uri (R)
519 requesting-user-name (R*)
520 limit (R*)
521 requested-attributes (R*)
522 which-jobs (R*)

523 my-jobs (R*)
524
525 Send-Document Request (O):
526 Group 1: Operation Attributes (R)
527 attributes-charset (R)
528 attributes-natural-language (R)
529 (printer-uri & job-id) | job-uri (R)
530 last-document (R)
531 requesting-user-name (R*)
532 document-name (R*)
533 document-format (R*)
534 document-natural-language (O*)
535 compression (O*)
536 Group 2: Document Content (R*)
537 <document content>
538
539 Send-URI Request (O):
540 Group 1: Operation Attributes (R)
541 attributes-charset (R)
542 attributes-natural-language (R)
543 (printer-uri & job-id) | job-uri (R)
544 last-document (R)
545 document-uri (R)
546 requesting-user-name (R*)
547 document-name (R*)
548 document-format (R*)
549 document-natural-language (O*)
550 compression (O*)
551
552 Cancel-Job Request (R):
553 Release-Job Request (O+):
554 Group 1: Operation Attributes (R)
555 attributes-charset (R)
556 attributes-natural-language (R)
557 (printer-uri & job-id) | job-uri (R)
558 requesting-user-name (R*)
559 message (O*)
560
561 Get-Job-Attributes Request (R):
562 Group 1: Operation Attributes (R)
563 attributes-charset (R)
564 attributes-natural-language (R)
565 (printer-uri & job-id) | job-uri (R)
566 requesting-user-name (R*)
567 requested-attributes (R*)
568
569 Pause-Printer Request (O+):
570 Resume-Printer Request (O+):
571 Purge-Printer Request (O+):
572 Group 1: Operation Attributes (R)

573 attributes-charset (R)
 574 attributes-natural-language (R)
 575 printer-uri (R)
 576 requesting-user-name (R*)
 577
 578 Hold-Job Request (O+):
 579 Restart-Job Request (O+):
 580 Group 1: Operation Attributes (R)
 581 attributes-charset (R)
 582 attributes-natural-language (R)
 583 (printer-uri & job-id) | job-uri (R)
 584 requesting-user-name (R*)
 585 job-hold-until (R*)
 586 message (O*)
 587

588 Operation Responses

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

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

592
 593 Print-Job Response (R):
 594 Create-Job Response (O):
 595 Send-Document Response (O):
 596 Group 1: Operation Attributes (R)
 597 attributes-charset (R)
 598 attributes-natural-language (R)
 599 status-message (O*)
 600 detailed-status-message (O*)
 601 Group 2: Unsupported Attributes (R*) (see Note 3)
 602 <unsupported attributes> (R*)
 603 Group 3: Job Object Attributes (R*) (see Note 2)
 604 job-uri (R)
 605 job-id (R)
 606 job-state (R)
 607 job-state-reasons (O* | R+)
 608 job-state-message (O*)
 609 number-of-intervening-jobs (O*)
 610
 611 Validate-Job Response (R):
 612 Cancel-Job Response (R):
 613 Hold-Job Response (O+):
 614 Release-Job Response (O+):
 615 Restart-Job Response (O+):
 616 Group 1: Operation Attributes (R)
 617 attributes-charset (R)
 618 attributes-natural-language (R)
 619 status-message (O*)

620 detailed-status-message (O*)
621 Group 2: Unsupported Attributes (R*) (see Note 3)
622 <unsupported attributes> (R*)
623
624 Print-URI Response (O):
625 Send-URI Response (O):
626 Group 1: Operation Attributes (R)
627 attributes-charset (R)
628 attributes-natural-language (R)
629 status-message (O*)
630 detailed-status-message (O*)
631 document-access-error (O*)
632 Group 2: Unsupported Attributes (R*) (see Note 3)
633 <unsupported attributes> (R*)
634 Group 3: Job Object Attributes (R*) (see Note 2)
635 job-uri (R)
636 job-id (R)
637 job-state (R)
638 job-state-reasons (O* | R+)
639 job-state-message (O*)
640 number-of-intervening-jobs (O*)
641
642 Get-Printer-Attributes Response (R):
643 Group 1: Operation Attributes (R)
644 attributes-charset (R)
645 attributes-natural-language (R)
646 status-message (O*)
647 detailed-status-message (O*)
648 Group 2: Unsupported Attributes (R*) (see Note 4)
649 <unsupported attributes> (R*)
650 Group 3: Printer Object Attributes (R*) (see Note 2)
651 <requested attributes> (R*)
652
653 Get-Jobs Response (R):
654 Group 1: Operation Attributes (R)
655 attributes-charset (R)
656 attributes-natural-language (R)
657 status-message (O*)
658 detailed-status-message (O*)
659 Group 2: Unsupported Attributes (R*) (see Note 4)
660 <unsupported attributes> (R*)
661 Group 3: Job Object Attributes (R*) (see Note 2, 5)
662 <requested attributes> (R*)
663
664 Get-Job-Attributes Response (R):
665 Group 1: Operation Attributes (R)
666 attributes-charset (R)
667 attributes-natural-language (R)
668 status-message (O*)
669 detailed-status-message (O*)

670 Group 2: Unsupported Attributes (R*) (see Note 4)
671 <unsupported attributes> (R*)
672 Group 3: Job Object Attributes (R*) (see Note 2)
673 <requested attributes> (R*)
674
675 Pause-Printer Response (O+):
676 Resume-Printer Response (O+):
677 Purge-Printer Response (O+):
678 Group 1: Operation Attributes (R)
679 attributes-charset (R)
680 attributes-natural-language (R)
681 status-message (O*)
682 detailed-status-message (O*)
683 Group 2: Unsupported Attributes (R*) (see Note 4)
684 <unsupported attributes> (R*)
685

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

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

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

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

694 3.1.2.1.5 Validate the values of the REQUIRED Operation attributes

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

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

- 699 a) that the length of each Operation attribute value is correct for the attribute syntax tag supplied by
700 the client according to [IPP-MOD] Section 4.1,
- 701 b) that the attribute syntax tag is correct for that Operation attribute according to [IPP-MOD]
702 Section 3,
- 703 c) that the value is in the range specified for that Operation attribute according to [IPP-MOD]
704 Section 3,
- 705 d) that multiple values are supplied by the client only for operation attributes that are multi-valued,
706 i.e., that are 1setOf X according to [IPP-MOD] Section 3.

707

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

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

720

721 attributes-charset (charset)

722 IF NOT a single non-empty 'charset' value, REJECT/RETURN 'client-error-bad-request'.
723 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-long'.
724 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-charset-
725 not-supported".
726

727 attributes-natural-language(naturalLanguage)

728 IF NOT a single non-empty 'naturalLanguage' value, REJECT/RETURN 'client-error-bad-request'.
729 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-long'.
730 ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-language-
731 supported" attribute. If the supplied value is not a member of the Printer object's "generated-natural-
732 language-supported" attribute, use the Printer object's "natural-language-configured" value.
733

734 requesting-user-name

735 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
736 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
737 IF the IPP object can obtain a better-authenticated name, use it instead.
738

739 job-name(name)

740 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
741 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
742 IF NOT supplied by the client, the Printer object creates a name from the document-name or document-
743 uri.
744

745 document-name (name)
746 IF NOT a single 'name' value, REJECT/RETURN 'client-error-bad-request'.
747 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
748

749 ipp-attribute-fidelity (boolean)
750 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-
751 request'.
752 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'
753 IF NOT supplied by the client, the IPP object assumes the value 'false'.
754

755 document-format (mimeMediaType)
756 IF NOT a single non-empty 'mimeMediaType' value, REJECT/RETURN 'client-error-bad-request'.
757 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
758 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN 'client-error-
759 document-format-not-supported'
760 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's "document-
761 format-default" attribute.
762

763 document-uri (uri)
764 IF NOT a single non-empty 'uri' value, REJECT/RETURN 'client-error-bad-request'.
765 IF the value length is greater than 1023 octets, REJECT/RETURN 'client-error-request-value-too-long'.
766 IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.
767 IF the client-supplied URI scheme is not supported, i.e. the value is not in the Printer object's referenced-
768 uri-scheme-supported" attribute, the Printer object MUST reject the request and return the 'client-
769 error-uri-scheme-not-supported' status code. The Printer object MAY check to see if the document
770 exists and is accessible. If the document is not found or is not accessible, REJECT/RETURN
771 'client-error-not found'.

772 last-document (boolean)
773 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-
774 request'.
775 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'
776

777 job-id (integer(1:MAX))
778 IF NOT an single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN
779 'client-error-bad-request'.
780 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-error-
781 gone' status code, if keep track of recently deleted jobs.
782

783 requested-attributes (1setOf keyword)

784 IF NOT one or more 'keyword' values, REJECT/RETURN 'client-error-bad-request'.
785 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
786 Ignore unsupported values, which are the keyword names of unsupported attributes. Don't bother to
787 copy such requested (unsupported) attributes to the Unsupported Attribute response group since the
788 response will not return them.
789

790 which-jobs (type2 keyword)

791 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
792 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
793 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to the
794 Unsupported Attributes response group and REJECT/RETURN 'client-error-attributes-or-values-
795 not-supported'.
796 Note: a Printer still supports the 'completed' value even if it keeps no completed/canceled/aborted jobs:
797 by returning no jobs when so queried.
798 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.
799

800 my-jobs (boolean)

801 IF NEITHER a single 'true' NOR a single 'false' 'boolean' value, REJECT/RETURN 'client-error-bad-
802 request'.
803 IF the value length is NOT equal to 1 octet, REJECT/RETURN 'client-error-request-value-too-long'
804 IF NOT supplied by the client, the IPP object assumes the 'false' value.
805

806 limit (integer(1:MAX))

807 IF NOT a single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN
808 'client-error-bad-request'.
809 IF NOT supplied by the client, the IPP object returns all jobs, no matter how many.
810

811 -----

812
813 3.1.2.1.6 Validate the values of the OPTIONAL Operation attributes

814 OPTIONAL Operation attributes are those that an IPP object MAY or MAY NOT support. An IPP object
815 validates the values of the OPTIONAL attributes supplied by the client. The IPP object performs the same
816 syntactic validation checks for each OPTIONAL attribute value as in Section 3.1.2.1.5. As in Section
817 3.1.2.1.5, if any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' or
818 the 'client-error-request-value-too-long' status code.

819 In addition, the IPP object checks each Operation attribute value against some Printer attribute or some
820 hard-coded value if there is no "xxx-supported" Printer attribute defined. If its value is not among those
821 supported or is not in the range supported, then the IPP object REJECTS the request and RETURNS the

822 error status code indicated in the table. If the value of the Printer object's "xxx-supported" attribute is 'no-
823 value' (because the system administrator hasn't configured a value), the check always fails.

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

826 -----

827 document-natural-language (naturalLanguage)

828 IF NOT a single non-empty 'naturalLanguage' value, REJECT/RETURN 'client-error-bad-request'.
829 IF the value length is greater than 63 octets, REJECT/RETURN 'client-error-request-value-too-long'.
830 IF NOT a value that the Printer object supports in document formats, (no corresponding "xxx-
831 supported" Printer attribute), REJECT/RETURN 'client-error-natural-language-not-supported'.
832

833 compression (type3 keyword)

834 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
835 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
836 IF NOT in the Printer object's "compression-supported" attribute, copy the attribute and the unsupported
837 value to the Unsupported Attributes response group and REJECT/RETURN 'client-error-attributes-
838 or-values-not-supported'.

839 Note to IPP/1.0 implementers: Support for the compression attribute was optional in IPP/1.0. However, an
840 IPP/1.0 object SHOULD at least check for the "compression" attribute being present and reject the create
841 request, if they don't support "compression". Not checking is a bug, since the data will be unintelligible.
842 job-k-octets (integer(0:MAX))

843 IF NOT a single 'integer' value equal to 4 octets,
844 REJECT/RETURN 'client-error-bad-request'.
845 IF NOT in the range of the Printer object's "job-k-octets-supported" attribute, copy the attribute and the
846 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-
847 error-attributes-or-values-not-supported'.
848

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

850 IF NOT a single 'integer' value equal to 4 octets,
851 REJECT/RETURN 'client-error-bad-request'.
852 IF NOT in the range of the Printer object's "job-impressions-supported" attribute, copy the attribute and
853 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-
854 error-attributes-or-values-not-supported'.
855

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

857 IF NOT a single 'integer' value equal to 4 octets,
858 REJECT/RETURN 'client-error-bad-request'.

859 IF NOT in the range of the Printer object's "job-media-sheets-supported" attribute, copy the attribute
860 and the unsupported value to the Unsupported Attributes response group and REJECT/RETURN
861 'client-error-attributes-or-values-not-supported'.
862

863 message (text(127))

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

865 IF the value length is greater than 127 octets,

866 REJECT/RETURN 'client-error-request-value-too-long'.
867

868 unknown or unsupported attribute

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

871 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
872 attribute value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.
873

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

889 3.1.2.2 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add
890 Documents

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

| 896 897 IIG Sect # | Flow | IPP error status codes |
|-----------------------|----------------------------|---|
| 898 ----- | ---- | ----- |
| 899 | | |
| 900 | v | No |
| 901 3.1.2.2.1 | <ipp-attribute-fidelity> | -----+ |
| 902 | <supplied?> | |
| 903 | Yes | |
| 904 | | ipp-attribute-fidelity = no |
| 905 | <----- | -----+ |
| 906 | v | No |
| 907 3.1.2.2.2 | <Printer is> | --> server-error-not-accepting-jobs |
| 908 | <accepting jobs?> | |
| 909 | Yes | |
| 910 | v | err |
| 911 3.1.2.3 | <Validate values of> | --> client-error-bad-request |
| 912 | <Job template attributes> | client-error-request-value-too-long |
| 913 | <(length, tag, range,> | |
| 914 | <multi-value)> | |
| 915 | ok | |
| 916 | v | err |
| 917 3.1.2.3 | <Validate values with> | --> client-error-bad-request |
| 918 | <supported values> | client-error-attributes-or-values- |
| 919 | ok | not-supported |
| 920 | v | err |
| 921 3.1.2.3.1 | <Any conflicting> | --> client-error-conflicting-attributes |
| 922 | <Job Template attr values> | client-error-attributes-or-values- |
| 923 | ok | not-supported |
| 924 | v | |

925 3.1.2.2.1 Default "ipp-attribute-fidelity" if not supplied

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

928 3.1.2.2.2 Check that the Printer object is accepting jobs

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

931 3.1.2.2.3 Validate the values of the Job Template attributes

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

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

937 b) that the attribute syntax tag is correct for that attribute according to [IPP-MOD] Sections 4.2 to
938 4.4.

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

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

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

- 950 1. reject the operation and return the 'client-error-bad-~~request-syntax~~' error status code
- 951 2. accept the operation and use the first occurrence of the attribute
- 952 3. accept the operation and use the last occurrence of the attribute

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

955 3.1.2.3 Algorithm for job validation

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

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

- 960 1. If U is multi-valued, validate each value X of U by performing the algorithm in Table 7 with each value
961 X. Each validation is separate from the standpoint of returning unsupported values. Example: If U is

962 "finishings" that the client supplies with 'staple', 'bind' values, then X takes on the successive values:
 963 'staple', then 'bind'

964 2. If V is multi-valued, validate X against each Z of V by performing the algorithm in Table 7 with each
 965 value Z. If a value Z validates, the validation for the attribute value X succeeds. If it fails, the algorithm
 966 is applied to the next value Z of V. If there are no more values Z of V, validation fails. Example" If V is
 967 "sides-supported" with values: 'one-sided', 'two-sided-long', and 'two-sided-short', then Z takes on the
 968 successive values: 'one-sided', 'two-sided-long', and 'two-sided-short'. If the client supplies "sides" with
 969 'two-sided-long', the first comparison fails ('one-sided' is not equal to 'two-sided-long'), the second
 970 comparison succeeds ('two-sided-long' is equal to 'two-sided-long'), and the third comparison ('two-
 971 sided-short' with 'two-sided-long') is not even performed.

972 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.

973 **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. |

974

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

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

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

992 -----

- 993 job-priority (integer(1:100))
- 994 IF NOT a single 'integer' value with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
- 995 request'.
- 996 IF NOT supplied by the client, use the value of the Printer object's "job-priority-default" attribute at job
- 997 submission time.
- 998 IF NOT in the range 1 to 100, inclusive, copy the attribute and the unsupported value to the
- 999 Unsupported Attributes response group.
- 1000 Map the value to the nearest supported value in the range 1:100 as specified by the number of discrete
- 1001 values indicated by the value of the Printer's "job-priority-supported" attribute. See the formula in
- 1002 [IPP-MOD] Section 4.2.1.
- 1003
- 1004 job-hold-until (type3 keyword | name)
- 1005 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
- 1006 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
- 1007 IF NOT supplied by the client, use the value of the Printer object's "job-hold-until" attribute at job
- 1008 submission time.
- 1009 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the
- 1010 unsupported value to the Unsupported Attributes response group.
- 1011
- 1012 job-sheets (type3 keyword | name)
- 1013 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.
- 1014 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
- 1015 IF NOT in the Printer object's "job-sheets-supported" attribute, copy the attribute and the unsupported
- 1016 value to the Unsupported Attributes response group.
- 1017
- 1018 multiple-document-handling (type2 keyword)
- 1019 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.
- 1020 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.
- 1021 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute and
- 1022 the unsupported value to the Unsupported Attributes response group.
- 1023
- 1024 copies (integer(1:MAX))
- 1025 IF NOT a single 'integer' value with a length equal to 4 octets,
- 1026 REJECT/RETURN 'client-error-bad-request'.
- 1027 IF NOT in range of the Printer object's "copies-supported" attribute
- 1028 copy the attribute and the unsupported value to the Unsupported Attributes response group.
- 1029
- 1030 finishings (1setOf type2 enum)
- 1031 IF NOT an 'enum' value(s) each with a length equal to 4 octets, REJECT/RETURN 'client-error-bad-
- 1032 request'.

1033 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the unsupported
1034 value(s), but not any supported values, to the Unsupported Attributes response group.
1035

1036 page-ranges (1setOf rangeOfInteger(1:MAX))

1037 IF NOT a 'rangeOfInteger' value(s) each with a length equal to 8 octets, REJECT/RETURN 'client-
1038 error-bad-request'.

1039 IF first value is greater than second value in any range, the ranges are not in ascending order, or ranges
1040 overlap, REJECT/RETURN 'client-error-bad-request'.

1041 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to the
1042 Unsupported Attributes response group and set the value to the "out-of-band" 'unsupported' value.
1043

1044 sides (type2 keyword)

1045 IF NOT a single 'keyword' value, REJECT/RETURN 'client-error-bad-request'.

1046 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.

1047 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported value
1048 to the Unsupported Attributes response group.
1049

1050 number-up (integer(1:MAX))

1051 IF NOT a single 'integer' value with a length equal to 4 octets,
1052 REJECT/RETURN 'client-error-bad-request'.

1053 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"
1054 attribute, copy the attribute and value to the Unsupported Attribute response group.
1055

1056 orientation-requested (type2 enum)

1057 IF NOT a single 'enum' value with a length equal to 4 octets,
1058 REJECT/RETURN 'client-error-bad-request'.

1059 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the
1060 unsupported value to the Unsupported Attributes response group.
1061

1062 media (type3 keyword | name)

1063 IF NOT a single 'keyword' or 'name' value, REJECT/RETURN 'client-error-bad-request'.

1064 IF the value length is greater than 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.

1065 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported value
1066 to the Unsupported Attributes response group.
1067

1068 printer-resolution (resolution)

1069 IF NOT a single 'resolution' value with a length equal to 9 octets,
1070 REJECT/RETURN 'client-error-bad-request'.

1071 IF NOT in the Printer object's "printer-resolution-supported" attribute, copy the attribute and the
1072 unsupported value to the Unsupported Attributes response group.

1073

1074 print-quality (type2 enum)

1075 IF NOT a single 'enum' value with a length equal to 4 octets,

1076 REJECT/RETURN 'client-error-bad-request'.

1077 IF NOT in the Printer object's "print-quality-supported" attribute, copy the attribute and the unsupported
1078 value to the Unsupported Attributes response group.

1079

1080 unknown or unsupported attribute (i.e., there is no corresponding Printer object "xxx-supported" attribute)

1081 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute

1082 syntax,

1083 REJECT/RETURN 'client-error-bad-request' if the length of the attribute syntax is fixed or 'client-error-
1084 request-value-too-long' if the length of the attribute syntax is variable.1085 ELSE copy the attribute and value to the Unsupported Attributes response group and change the
1086 attribute value to the "out-of-band" 'unsupported' value. Any remaining Job Template Attributes are
1087 either unknown or unsupported Job Template attributes and are validated algorithmically according
1088 to their attribute syntax for proper length (see below).

1089

1090 -----
 1090 If the attribute syntax is supported AND the length check fails, the IPP object REJECTS the request and
 1091 RETURNS the 'client-error-bad-request' if the length of the attribute syntax is fixed or the 'client-error-
 1092 request-value-too-long' status code if the length of the attribute syntax is variable. Otherwise, the IPP object
 1093 copies the unsupported Job Template attribute to the Unsupported Attributes response group and changes
 1094 the attribute value to the "out-of-band" 'unsupported' value. The following table shows the length checks
 1095 for all attribute syntaxes. In the following table: "<=" means less than or equal, "=" means equal to:

| 1096 Name | Octet length check for read-write attributes |
|----------------------------|--|
| 1097 ----- | ----- |
| 1098 'textWithLanguage | <= 1023 AND 'naturalLanguage' <= 63 |
| 1099 'textWithoutLanguage' | <= 1023 |
| 1100 'nameWithLanguage' | <= 255 AND 'naturalLanguage' <= 63 |
| 1101 'nameWithoutLanguage' | <= 255 |
| 1102 'keyword' | <= 255 |
| 1103 'enum' | = 4 |
| 1104 'uri' | <= 1023 |
| 1105 'uriScheme' | <= 63 |
| 1106 'charset' | <= 63 |
| 1107 'naturalLanguage' | <= 63 |
| 1108 'mimeType' | <= 255 |
| 1109 'octetString' | <= 1023 |
| 1110 'boolean' | = 1 |
| 1111 'integer' | = 4 |
| 1112 'rangeOfInteger' | = 8 |
| 1113 'dateTime' | = 11 |
| 1114 'resolution' | = 9 |
| 1115 'lsetOf X' | |

1116

1117 Note: It's possible for a Printer to receive a zero length keyword in a request. Since this is a keyword, its
1118 value needs to be compared with the supported values. Assuming that the printer doesn't have any values in
1119 its corresponding "xxx-supported" attribute that are keywords of zero length, the comparison will fail.
1120 Then the request will be accepted or rejected depending on the value of "ipp-attributes-fidelity" being 'false'
1121 or 'true', respectively. No special handling is required for

1122 3.1.2.3.1 Check for conflicting Job Template attributes values

1123 Once all the Operation and Job Template attributes have been checked individually, the Printer object
1124 SHOULD check for any conflicting values among all the supported values supplied by the client. For
1125 example, a Printer object might be able to staple and to print on transparencies, however due to physical
1126 stapling constraints, the Printer object might not be able to staple transparencies. The IPP object copies the
1127 supported attributes and their conflicting attribute values to the Unsupported Attributes response group.
1128 The Printer object only copies over those attributes that the Printer object either ignores or substitutes in
1129 order to resolve the conflict, and it returns the original values which were supplied by the client. For
1130 example suppose the client supplies "finishings" equals 'staple' and "media" equals 'transparency', but the
1131 Printer object does not support stapling transparencies. If the Printer chooses to ignore the stapling request
1132 in order to resolve the conflict, the Printer objects returns "finishings" equal to 'staple' in the Unsupported
1133 Attributes response group. If any attributes are multi-valued, only the conflicting values of the attributes
1134 are copied.

1135 Note: The decisions made to resolve the conflict (if there is a choice) is implementation dependent.

1136 3.1.2.3.2 Decide whether to REJECT the request

1137 If there were any unsupported Job Template attributes or unsupported/conflicting Job Template attribute
1138 values and the client supplied the "ipp-attribute-fidelity" attribute with the 'true' value, the Printer object
1139 REJECTS the request and return the status code:

1140 (1) 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes supplied
1141 by the client.

1142 (2) 'client-error-attributes-or-values-not-supported' status code, otherwise.

1143

1144 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this
1145 step. If the unsupported Operation attribute was a serious error, the above already rejected the request in a
1146 previous step. If control gets to this step with unsupported Operation attributes being returned, they are not
1147 serious errors.

1148 In general, the final results of Job processing are unknown at Job submission time. The client has to rely on
1149 notifications or polling to find out what happens at Job processing time. However, there are cases in which
1150 some Printers can determine at Job submission time that Job processing is going to fail. As an
1151 optimization, we'd like to have the Printer reject the Job in these cases.

1152 There are three types of "processing" errors that might be detectable at Job submission time:

1153 1. 'client-error-document-format-not-supported': For the Print-Job, Send-Document, Print-URI, and Send-
1154 URI operations, if all these conditions are true:

- 1155 · the Printer supports auto-sensing,
- 1156 · the request "document-format" operation attribute is 'application/octet-stream',
- 1157 · the Printer receives document data before responding,
- 1158 · the Printer auto-senses the document format before responding,
- 1159 · the sensed document format is not supported by the Printer

1160 then the Printer should respond with 'client-error-document-format-not-supported' status.

1161 2. 'client-error-compression-error': For the Print-Job, Send-Document, Print-URI, and Send-URI
1162 operations, if all these conditions are true:

- 1163 · the client supplies a supported value for the "compression" operation attribute in the request
- 1164 · the Printer receives document data before responding,
- 1165 · the Printer attempts to decompress the document data before responding,
- 1166 · the document data cannot be decompressed using the algorithm specified by the

1167 "compression" operation attribute

1168 then the Printer should respond with 'client-error-compression-error' status.

1169 3. 'client-error-document-access-error': For the Print-URI, and Send-URI operations, if the Printer attempts
1170 and fails to pull the referenced document data before responding, it should respond with 'client-error-
1171 document-access-error' status.

1172 Some Printers are not able to detect these errors until Job processing time. In that case, the errors are
1173 recorded in the corresponding job-state and job-state reason attributes. (There is no standard way for a
1174 client to determine whether a Printer can detect these errors at Job submission time.) For example, if auto-
1175 sensing happens AFTER the job is accepted (as opposed to auto-sensing at submit time before returning the
1176 response), the implementation aborts the job, puts the job in the 'aborted' state and sets the 'unsupported-
1177 document-format' value in the job's "job-state-reasons".

1178 A client should always provide a valid "document-format" operation attribute whenever practical. In the
1179 absence of other information, a client itself may sniff the document data to determine document format.

1180 Auto sensing at Job submission time may be more difficult for the Printer when combined with
1181 compression. For auto-sensed Jobs, a client may be better off deferring compression to the transfer
1182 protocol layer, e.g.; by using the HTTP Content-Encoding header.

1183 3.1.2.3.3 For the Validate-Job operation, RETURN one of the success status codes

1184 If the requested operation is the Validate-Job operation, the Printer object returns:

- 1185 (1) the "successful-ok" status code, if there are no unsupported or conflicting Job Template attributes or
1186 values.
1187 (2) the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or
1188 values.
1189 (3) the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template
1190 attributes or values.
1191

1192 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this
1193 step. If the unsupported Operation attribute was a serious error, the above already rejected the request in a
1194 previous step. If control gets to this step with unsupported Operation attributes being returned, they are not
1195 serious errors.

1196 3.1.2.3.4 Create the Job object with attributes to support

1197 If "ipp-attribute-fidelity" is set to 'false' (or it was not supplied by the client), the Printer object:

- 1198 (1) creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and
1199 initializes all of the job's other supported Job Description attributes.
1200 (2) removes all unsupported attributes from the Job object.
1201 (3) for each unsupported value, removes either the unsupported value or substitutes the unsupported
1202 attribute value with some supported value. If an attribute has no values after removing unsupported
1203 values from it, the attribute is removed from the Job object (so that the normal default behavior at
1204 job processing time will take place for that attribute).
1205 (4) for each conflicting value, removes either the conflicting value or substitutes the conflicting attribute
1206 value with some other supported value. If an attribute has no values after removing conflicting
1207 values from it, the attribute is removed from the Job object (so that the normal default behavior at
1208 job processing time will take place for that attribute).
1209

1210 If there were no attributes or values flagged as unsupported, or the value of "ipp-attribute-fidelity" was
1211 'false', the Printer object is able to accept the create request and create a new Job object. If the "ipp-
1212 attribute-fidelity" attribute is set to 'true', the Job Template attributes that populate the new Job object are
1213 necessarily all the Job Template attributes supplied in the create request. If the "ipp-attribute-fidelity"
1214 attribute is set to 'false', the Job Template attributes that populate the new Job object are all the client
1215 supplied Job Template attributes that are supported or that have value substitution. Thus, some of the
1216 requested Job Template attributes may not appear in the Job object because the Printer object did not
1217 support those attributes. The attributes that populate the Job object are persistently stored with the Job
1218 object for that Job. A Get-Job-Attributes operation on that Job object will return only those attributes that
1219 are persistently stored with the Job object.

1220 Note: All Job Template attributes that are persistently stored with the Job object are intended to be
1221 "override values"; that is, they that take precedence over whatever other embedded instructions might be in
1222 the document data itself. However, it is not possible for all Printer objects to realize the semantics of
1223 "override". End users may query the Printer's "pdl-override-supported" attribute to determine if the Printer
1224 either attempts or does not attempt to override document data instructions with IPP attributes.

1225 There are some cases, where a Printer supports a Job Template attribute and has an associated default value
1226 set for that attribute. In the case where a client does not supply the corresponding attribute, the Printer does
1227 not use its default values to populate Job attributes when creating the new Job object; only Job Template
1228 attributes actually in the create request are used to populate the Job object. The Printer's default values are
1229 only used later at Job processing time if no other IPP attribute or instruction embedded in the document
1230 data is present.

1231 Note: If the default values associated with Job Template attributes that the client did not supply were to be
1232 used to populate the Job object, then these values would become "override values" rather than defaults. If
1233 the Printer supports the 'attempted' value of the "pdl-override-supported" attribute, then these override
1234 values could replace values specified within the document data. This is not the intent of the default value
1235 mechanism. A default value for an attribute is used only if the create request did not specify that attribute
1236 (or it was ignored when allowed by "ipp-attribute-fidelity" being 'false') and no value was provided within
1237 the content of the document data.

1238 If the client does not supply a value for some Job Template attribute, and the Printer does not support that
1239 attribute, as far as IPP is concerned, the result of processing that Job (with respect to the missing attribute)
1240 is undefined.

1241 3.1.2.3.5 Return one of the success status codes

1242 Once the Job object has been created, the Printer object accepts the request and returns to the client:

- 1243 (1) the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes or
1244 values.
- 1245 (2) the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template
1246 attribute or values.
- 1247 (3) the 'successful-ok-ignored-or-substituted-attributes' status code, if there are only unsupported Job
1248 Template attributes or values.

1249 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in this
1250 step. If the unsupported Operation attribute was a serious error, the above already rejected the request in a
1251 previous step. If control gets to this step with unsupported Operation attributes being returned, they are not
1252 serious errors.
1253

1254 The Printer object also returns Job status attributes that indicate the initial state of the Job ('pending',
1255 'pending-held', 'processing', etc.), etc. See Print-Job Response, [IPP-MOD] section 3.2.1.2.

1256 3.1.2.3.6 Accept appended Document Content

1257 The Printer object accepts the appended Document Content data and either starts it printing, or spools it for
1258 later processing.

1259 3.1.2.3.7 Scheduling and Starting to Process the Job

1260 The Printer object uses its own configuration and implementation specific algorithms for scheduling the Job
1261 in the correct processing order. Once the Printer object begins processing the Job, the Printer changes the
1262 Job's state to 'processing'. If the Printer object supports PDL override (the "pdl-override-supported" attribute
1263 set to 'attempted'), the implementation does its best to see that IPP attributes take precedence over
1264 embedded instructions in the document data.

1265 3.1.2.3.8 Completing the Job

1266 The Printer object continues to process the Job until it can move the Job into the 'completed' state. If an
1267 Cancel-Job operation is received, the implementation eventually moves the Job into the 'canceled' state. If
1268 the system encounters errors during processing that do not allow it to progress the Job into a completed
1269 state, the implementation halts all processing, cleans up any resources, and moves the Job into the 'aborted'
1270 state.

1271 3.1.2.3.9 Destroying the Job after completion

1272 Once the Job moves to the 'completed', 'aborted', or 'canceled' state, it is an implementation decision as to
1273 when to destroy the Job object and release all associated resources. Once the Job has been destroyed, the
1274 Printer would return either the "client-error-not-found" or "client-error-gone" status codes for operations
1275 directed at that Job.

1276 Note: the Printer object SHOULD NOT re-use a "job-uri" or "job-id" value for a sufficiently long time
1277 after a job has been destroyed, so that stale references kept by clients are less likely to access the wrong
1278 (newer) job.

1279 3.1.2.3.10 Interaction with "ipp-attribute-fidelity"

1280 Some Printer object implementations may support "ipp-attribute-fidelity" set to 'true' and "pdl-override-
1281 supported" set to 'attempted' and yet still not be able to realize exactly what the client specifies in the create
1282 request. This is due to legacy decisions and assumptions that have been made about the role of job
1283 instructions embedded within the document data and external job instructions that accompany the document
1284 data and how to handle conflicts between such instructions. The inability to be 100% precise about how a
1285 given implementation will behave is also compounded by the fact that the two special attributes, "ipp-
1286 attribute-fidelity" and "pdl-"override-supported", apply to the whole job rather than specific values for each
1287 attribute. For example, some implementations may be able to override almost all Job Template attributes
1288 except for "number-up". Character Sets, natural languages, and internationalization

1289 This section discusses character set support, natural language support and internationalization.

1290 3.1.2.3.11 Character set code conversion support (~~Issue 1.5~~)

1291 IPP clients and IPP objects are REQUIRED to support UTF-8. They MAY support additional charsets. It
1292 is RECOMMENDED that an IPP object also support US-ASCII, since many clients support US-ASCII, and

1293 indicate that UTF-8 and US-ASCII are supported by populating the Printer's "charset-supported" with 'utf-8'
1294 and 'us-ascii' values. An IPP object is required to code covert with as little loss as possible between the
1295 charsets that it supports, as indicated in the Printer's "charsets-supported" attribute.

1296 How should the server handle the situation where the "attributes-charset" of the response itself is "us-ascii",
1297 but one or more attributes in that response is in the "utf-8" format?

1298 Example: Consider a case where a client sends a Print-Job request with "utf-8" as the value of "attributes-
1299 charset" and with the "job-name" attribute supplied. Later another client submits a Get-Job-Attribute or
1300 Get-Jobs request. This second request contains the "attributes-charset" with value "us-ascii" and
1301 "requested-attributes" attribute with exactly one value "job-name".

1302 According to the IPP-Mod document (section 3.1.4.2), the value of the "attributes-charset" for the response
1303 of the second request must be "us-ascii" since that is the charset specified in the request. The "job-name"
1304 value, however, is in "utf-8" format. Should the request be rejected even though both "utf-8" and "us-ascii"
1305 charsets are supported by the server? or should the "job-name" value be converted to "us-ascii" and return
1306 "successful-ok-conflicting-attributes" (0x0002) as the status code?

1307 Answer: An IPP object that supports both utf-8 (REQUIRED) and us-ascii, the second paragraph of section
1308 3.1.4.2 applies so that the IPP object MUST accept the request, perform code set conversion between these
1309 two charsets with "the highest fidelity possible" and return 'successful-ok', rather than a warning
1310 'successful-ok-conflicting-attributes, or an error. The printer will do the best it can to convert between each
1311 of the character sets that it supports--even if that means providing a string of question marks because none
1312 of the characters are representable in US ASCII. If it can't perform such conversion, it MUST NOT
1313 advertise us-ascii as a value of its "attributes-charset-supported" and MUST reject any request that requests
1314 'us-ascii'.

1315 One IPP object implementation strategy is to convert all request text and name values to a Unicode internal
1316 representation. This is 16-bit and virtually universal. Then convert to the specified operation attributes-
1317 charset on output.

1318 Also it would be smarter for a client to ask for 'utf-8', rather than 'us-ascii' and throw away characters that it
1319 doesn't understand, rather than depending on the code conversion of the IPP object.

1320 3.1.2.3.12 What charset to return when an unsupported charset is requested (Issue 1.19)?

1321 Section 3.1.4.1 Request Operation attributes was clarified in November 1998 as follows:

1322 All clients and IPP objects MUST support the 'utf-8' charset [RFC2044] and MAY support additional
1323 charsets provided that they are registered with IANA [IANA-CS]. If the Printer object does not support the
1324 client supplied charset value, the Printer object MUST reject the request, set the "attributes-charset" to 'utf-
1325 8' in the response, and return the 'client-error-charset-not-supported' status code and any 'text' or 'name'
1326 attributes using the 'utf-8' charset.

1327 Since the client and IPP object MUST support UTF-8, returning any text or name attributes in UTF-8 when
1328 the client requests a charset that is not supported should allow the client to display the text or name.

1329 Since such an error is a client error, rather than a user error, the client should check the status code first so
1330 that it can avoid displaying any other returned 'text' and 'name' attributes that are not in the charset
1331 requested.

1332 Furthermore, [ipp-mod] section 14.1.4.14 client-error-charset-not-supported (0x040D) was clarified in
1333 November 1998 as follows:

1334 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-
1335 charset" operation attribute, the Printer MUST reject the operation and return this status and any 'text' or
1336 'name' attributes using the 'utf-8' charset (see Section 3.1.4.1).

1337 3.1.2.3.13 Natural Language Override (NLO) (~~Issue 1.45~~)

1338 The 'text' and 'name' attributes each have two forms. One has an implicit natural language, and the other
1339 has an explicit natural language. The 'textWithoutLanguage' and 'textWithLanguage' are the two 'text'
1340 forms. The 'nameWithoutLanguage' and 'nameWithLanguage' are the two 'name' forms. If a receiver (IPP
1341 object or IPP client) supports an attribute with attribute syntax 'text', it MUST support both forms in a
1342 request and a response. A sender (IPP client or IPP object) MAY send either form for any such attribute.
1343 When a sender sends a WithoutLanguage form, the implicit natural language is specified in the "attributes-
1344 natural-language" operation attribute, which all senders MUST include in every request and response.

1345 When a sender sends a WithLanguage form, it MAY be different from the implicit natural language
1346 supplied by the sender or it MAY be the same. The receiver MUST treat either form equivalently.

1347 There is an implementation decision for senders, whether to always send the WithLanguage forms or use
1348 the WithoutLanguage form when the attribute's natural language is the same as the request or response. The
1349 former approach makes the sender implementation simpler. The latter approach is more efficient on the
1350 wire and allows inter-working with non-conforming receivers that fail to support the WithLanguage forms.
1351 As each approach have advantages, the choice is completely up to the implementer of the sender.

1352 Furthermore, when a client receives a 'text' or 'name' job attribute that it had previously supplied, that client
1353 MUST NOT expect to see the attribute in the same form, i.e., in the same WithoutLanguage or
1354 WithLanguage form as the client supplied when it created the job. The IPP object is free to transform the
1355 attribute from the WithLanguage form to the WithoutLanguage form and vice versa, as long as the natural
1356 language is preserved. However, in order to meet this latter requirement, it is usually simpler for the IPP
1357 object implementation to store the natural language explicitly with the attribute value, i.e., to store using an
1358 internal representation that resembles the WithLanguage form.

1359 The IPP Printer MUST copy the natural language of a job, i.e., the value of the "attributes-natural-language"
1360 operation attribute supplied by the client in the create operation, to the Job object as a Job Description
1361 attribute, so that a client is able to query it. In returning a Get-Job-Attributes response, the IPP object MAY
1362 return one of three natural language values in the response's "attributes-natural-language" operation
1363 attribute: (1) that requested by the requester, (2) the natural language of the job, or (3) the configured
1364 natural language of the IPP Printer, if the requested language is not supported by the IPP Printer.

1365 This "attributes-natural-language" Job Description attribute is useful for an IPP object implementation that
1366 prints start sheets in the language of the user who submitted the job. This same Job Description attribute is
1367 useful to a multi-lingual operator who has to communicate with different job submitters in different natural
1368 languages. This same Job Description attribute is expected to be used in the future to generate notification
1369 messages in the natural language of the job submitter.

1370 Early drafts of [IPP-MOD] contained a job-level natural language override (NLO) for the Get-Jobs
1371 response. A job-level (NLO) is an (unrequested) Job Attribute which then specified the implicit natural
1372 language for any other WithoutLanguage job attributes returned in the response for that job.
1373 Interoperability testing of early implementations showed that no one was implementing the job-level NLO
1374 in Get-Job responses. So the job-level NLO was eliminated from the Get-Jobs response. This
1375 simplification makes all requests and responses consistent in that the implicit natural language for any
1376 WithoutLanguage 'text' or 'name' form is always supplied in the request's or response's "attributes-natural-
1377 language" operation attribute.

1378 3.1.3 Status codes returned by operation (~~Issue 1.50~~)

1379 This section corresponds to [IPP-MOD] section 3.1.6 "Operation Response Status Codes and Status
1380 Messages". This section lists all status codes once in the first operation (Print-Job). Then it lists the status
1381 codes that are different or specialized for subsequent operations under each operation.

1382 3.1.3.1 Printer Operations

1383 3.1.3.1.1 Print-Job

1384 The Printer object MUST return one of the following "status-code" values for the indicated reason.
1385 Whether all of the document data has been accepted or not before returning the success or error response
1386 depends on implementation. See Section 13 in [IPP-MOD] for a more complete description of each status
1387 code.

1388 For the following success status codes, the Job object has been created and the "job-id", and "job-uri"
1389 assigned and returned in the response:

1390 successful-ok: no request attributes were substituted or ignored.

1391 successful-ok-ignored-or-substituted-attributes: some supplied (1) attributes were ignored or (2)
1392 unsupported attribute syntaxes or values were substituted with supported values or were ignored.

1393 Unsupported attributes, attribute syntax's, or values MUST be returned in the Unsupported
1394 Attributes group of the response.

1395 successful-ok-conflicting-attributes: some supplied attribute values conflicted with the values of other
1396 supplied attributes and were either substituted or ignored. Attributes or values which conflict with
1397 other attributes and have been substituted or ignored MUST be returned in the Unsupported
1398 Attributes group of the response as supplied by the client.
1399

1400 [ipp-mod] section 3.1.6 Operation Status Codes and Messages states: (~~Issue 1.19~~):

1401 If the Printer object supports the "status-message" operation attribute, it SHOULD use the REQUIRED 'utf-
1402 8' charset to return a status message for the following error status codes (see section 13 in [IPP-MOD]):
1403 'client-error-bad-request', 'client-error-charset-not-supported', 'server-error-internal-error', 'server-error-
1404 operation-not-supported', and 'server-error-version-not-supported'. In this case, it MUST set the value of
1405 the "attributes-charset" operation attribute to 'utf-8' in the error response.

1406 For the following error status codes, no job is created and no "job-id" or "job-uri" is returned:

1407 client-error-bad-request: The request syntax does not conform to the specification.
1408 client-error-forbidden: The request is being refused for authorization or authentication reasons. The
1409 implementation security policy is to not reveal whether the failure is one of authentication or
1410 authorization.
1411 client-error-not-authenticated: Either the request requires authentication information to be supplied or
1412 the authentication information is not sufficient for authorization.
1413 client-error-not-authorized: The requester is not authorized to perform the request on the target object.
1414 client-error-not-possible: The request cannot be carried out because of the state of the system. See also
1415 'server-error-not-accepting-jobs' status code, which MUST take precedence if the Printer object's
1416 "printer-accepting-jobs" attribute is 'false'.
1417 client-error-timeout: not applicable.
1418 client-error-not-found: the target object does not exist.
1419 client-error-gone: the target object no longer exists and no forwarding address is known.
1420 client-error-request-entity-too-large: the size of the request and/or print data exceeds the capacity of the
1421 IPP Printer to process it.
1422 client-error-request-value-too-long: the size of request variable length attribute values, such as 'text' and
1423 'name' attribute syntax's, exceed the maximum length specified in [IPP-MOD] for the attribute and
1424 MUST be returned in the Unsupported Attributes Group.
1425 client-error-document-format-not-supported: the document format supplied is not supported. The
1426 "document-format" attribute with the unsupported value MUST be returned in the Unsupported
1427 Attributes Group. This error SHOULD take precedence over any other 'xxx-not-supported' error,
1428 except 'client-error-charset-not-supported'.
1429 client-error-attributes-or-values-not-supported: one or more supplied attributes, attribute syntax's, or
1430 values are not supported and the client supplied the "ipp-attributes-fidelity" operation attribute with
1431 a 'true' value. They MUST be returned in the Unsupported Attributes Group as explained below.
1432 client-error-uri-scheme-not-supported: not applicable.
1433 client-error-charset-not-supported: the charset supplied in the "attributes-charset" operation attribute is
1434 not supported. The Printer's "configured-charset" MUST be returned in the response as the value of
1435 the "attributes-charset" operation attribute and used for any 'text' and 'name' attributes returned in the
1436 error response. This error SHOULD take precedence over any other error, unless the request syntax
1437 is so bad that the client's supplied "attributes-charset" cannot be determined.
1438 client-error-conflicting-attributes: one or more supplied attribute values conflicted with each other and
1439 the client supplied the "ipp-attributes-fidelity" operation attribute with a 'true' value. They MUST
1440 be returned in the Unsupported Attributes Group as explained below.
1441 server-error-internal-error: an unexpected condition prevents the request from being fulfilled.
1442 server-error-operation-not-supported: not applicable (since Print-Job is REQUIRED).
1443 server-error-service-unavailable: the service is temporarily overloaded.

1444 server-error-version-not-supported: the version in the request is not supported. The "closest" version
1445 number supported MUST be returned in the response.
1446 server-error-device-error: a device error occurred while receiving or spooling the request or document
1447 data or the IPP Printer object can only accept one job at a time.
1448 server-error-temporary-error: a temporary error such as a buffer full write error, a memory overflow, or
1449 a disk full condition occurred while receiving the request and/or the document data.
1450 server-error-not-accepting-jobs: the Printer object's "printer-is-not-accepting-jobs" attribute is 'false'.
1451 server-error-busy: the Printer is too busy processing jobs to accept another job at this time.
1452 server-error-job-canceled: the job has been canceled by an operator or the system while the client was
1453 transmitting the document data.

1454 3.1.3.1.2 Print-URI

1455 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Print-
1456 URI with the following specializations and differences. See Section 14 for a more complete description of
1457 each status code.

1458 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation
1459 attribute is not supported and is returned in the Unsupported Attributes group.
1460 server-error-operation-not-supported: the Print-URI operation is not supported.
1461

1462 3.1.3.1.3 Validate-Job

1463 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to
1464 Validate-Job. See Section 13 in [IPP-MOD] for a more complete description of each status code.

1465 3.1.3.1.4 Create-Job

1466 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Create-
1467 Job with the following specializations and differences. See Section 13 in [IPP-MOD] for a more complete
1468 description of each status code.

1469 server-error-operation-not-supported: the Create-Job operation is not supported.

1470 3.1.3.1.5 Get-Printer-Attributes

1471 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the
1472 Get-Printer-Attributes operation with the following specialization's and differences. See Section 13 in
1473 [IPP-MOD] for a more complete description of each status code.

1474 For the following success status codes, the requested attributes are returned in Group 3 in the response:

1475 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1476 attributes were unsupported.

1477 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1478 operation attribute MAY, but NEED NOT, be returned with the unsupported values.
1479 successful-ok-conflicting-attributes: same as Print-Job.

1480 For the error status codes, Group 3 is returned containing no attributes or is not returned at all:

1481 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.
1482 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.
1483 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1484 MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.
1485 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.
1486 server-error-operation-not-supported: not applicable (since Get-Printer-Attributes is REQUIRED).
1487 server-error-device-error: same as Print-Job, except that no document data is involved.
1488 server-error-temporary-error: same as Print-Job, except that no document data is involved.
1489 server-error-not-accepting-jobs: not applicable..
1490 server-error-busy: same as Print-Job, except the IPP object is too busy to accept even query requests.
1491 server-error-job-canceled: not applicable..

1492 3.1.3.1.6 Get-Jobs

1493 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the
1494 Get-Jobs operation with the following specialization's and differences. See Section 13 in [IPP-MOD] for a
1495 more complete description of each status code.

1496 For the following success status codes, the requested attributes are returned in Group 3 in the response:

1497 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1498 attributes were unsupported.
1499 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1500 operation attribute MAY, but NEED NOT, be returned with the unsupported values.
1501 successful-ok-conflicting-attributes: same as Print-Job.

1502 For any error status codes, Group 3 is returned containing no attributes or is not returned at all. The
1503 following brief error status code descriptions contain unique information for use with Get-Jobs operation.
1504 See section 14 for the other error status codes that apply uniformly to all operations:

1505 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.
1506 client-error-request-entity-too-large: same as Print-job, except that no print data is involved.
1507 client-error-document-format-not-supported: not applicable.
1508 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1509 MUST be ignored and 'successful-ok-ignored-or-substituted-attributes' returned.
1510 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attribute-fidelity" is not involved.
1511 server-error-operation-not-supported: not applicable (since Get-Jobs is REQUIRED).
1512 server-error-device-error: same as Print-Job, except that no document data is involved.
1513 server-error-temporary-error: same as Print-Job, except that no document data is involved.
1514 server-error-not-accepting-jobs: not applicable.
1515 server-error-job-canceled: not applicable.

1516 3.1.3.1.7 Pause-Printer

1517 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Pause-
1518 Printer with the following specializations and differences. See Section 13 in [IPP-MOD] for a more
1519 complete description of each status code.

1520 For the following success status codes, the Printer object is being stopped from scheduling jobs on all its
1521 devices.

1522 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

1523 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

1524 successful-ok-conflicting-attributes: same as Print-Job.

1525

1526 For any of the error status codes, the Printer object has not been stopped from scheduling jobs on all its
1527 devices.

1528 client-error-not-possible: not applicable.

1529 client-error-not-found: the target Printer object does not exist.

1530 client-error-gone: the target Printer object no longer exists and no forwarding address is known.

1531 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.

1532 client-error-document-format-not-supported: not applicable.

1533 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-
1534 jobs" attribute is not involved.

1535 server-error-operation-not-supported: the Pause-Printer operation is not supported.

1536 server-error-device-error: not applicable.

1537 server-error-temporary-error: same as Print-Job, except no document data is involved.

1538 server-error-not-accepting-jobs: not applicable.

1539 server-error-job-canceled: not applicable.

1540 3.1.3.1.8 Resume-Printer

1541 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
1542 specialization's described for Pause-Printer are applicable to Resume-Printer. See Section 13 in [IPP-
1543 MOD] for a more complete description of each status code.

1544 For the following success status codes, the Printer object resumes scheduling jobs on all its devices.

1545 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

1546 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

1547 successful-ok-conflicting-attributes: same as Print-Job.

1548 For any of the error status codes, the Printer object does not resume scheduling jobs.

1549 server-error-operation-not-supported: the Resume-Printer operation is not supported.

1550

1551 [3.1.3.1.8.1 What about Printers unable to change state due to an error condition?](#)

1552 If, in case, the IPP printer is unable to change its state due to some problem with the actual printer device
1553 (say, it is shut down or there is a media-jam as indicated in [ipp-mod]), what should be the result of the
1554 "Resume-printer" operation? Should it still change the 'printer-state-reasons' and return success or should it
1555 fail ?

1556 The 'resume-printer' operation must clear the 'paused' or 'moving-to-paused' 'printer-state-message'. The
1557 operation must return a 'successful-ok' status code.

1558 esume-Printer”:

1559 3.1.3.1.8.2 How is 'printer-state' handled on Resume-Printer?

1561 If “Resume-Printer” succeeds, what should be the value of 'Printer-state' and who should take care of the
1562 'Printer-state' later on ?

1563 The “Resume-Printer” operation may change the “printer-state-reasons” value.

1564 The “printer-state” will change to one of three states:

1565 1) 'idle' - no additional jobs and no error conditions present

1566 2) 'processing' - job available and no error conditions present

1567 3) current state (i.e. no change) an error condition is present (e.g. media jam)

1568 In the third case the 'printer-state-reason' will be cleared by automata when it detects the error condition no
1569 longer exists. The 'printer-state' will move to 'idle' or 'processing' when conditions permit. (i.e. no more
1570 error conditions)

1571 3.1.3.1.9 Purge-Printer

1572 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
1573 specialization's described for Pause-Printer are applicable to Purge-Printer. See Section 13 in [IPP-MOD]
1574 for a more complete description of each status code.

1575 For the following success status codes, the Printer object purges all it's jobs.

1576 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

1577 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

1578 successful-ok-conflicting-attributes: same as Print-Job.

1579 For any of the error status codes, the Printer object does not purge any jobs.

1580 server-error-operation-not-supported: the Purge-Printer operation is not supported.

1581 3.1.3.2 Job Operations

1582 3.1.3.2.1 Send-Document

1583 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to the
1584 Get-Printer-Attributes operation with the following specialization's and differences. See Section 13 in
1585 [IPP-MOD] for a more complete description of each status code.

1586 For the following success status codes, the document has been added to the specified Job object and the
1587 job's "number-of-documents" attribute has been incremented:

1588 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

1589 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

1590 successful-ok-conflicting-attributes: same as Print-Job.

1591 For the error status codes, no document has been added to the Job object and the job's "number-of-
1592 documents" attribute has not been incremented:

1593 client-error-not-possible: Same as Print-Job, except that the Printer's "printer-is-accepting-jobs"
1594 attribute is not involved, so that the client is able to finish submitting a multi-document job after this
1595 attribute has been set to 'true'. Another condition is that the state of the job precludes Send-
1596 Document, i.e., the job has already been closed out by the client. However, if the IPP Printer closed
1597 out the job due to timeout, the 'client-error-timeout' error status SHOULD be returned instead.

1598 client-error-timeout: This request was sent after the Printer closed the job, because it has not received a
1599 Send-Document or Send-URI operation within the Printer's "multiple-operation-time-out" period .

1600 client-error-request-entity-too-large: same as Print-Job.

1601 client-error-conflicting-attributes: same as Print-Job, except that "ipp-attributes-fidelity" operation
1602 attribute is not involved..

1603 server-error-operation-not-supported: the Send-Document request is not supported.

1604 server-error-not-accepting-jobs: not applicable.

1605 server-error-job-canceled: the job has been canceled by an operator or the system while the client was
1606 transmitting the data.

1607 3.1.3.2.2 Send-URI

1608 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
1609 specialization's described for Send-Document are applicable to Send-URI. See Section 13 in [IPP-MOD]
1610 for a more complete description of each status code.

1611 client-error-uri-scheme-not-supported: the URI scheme supplied in the "document-uri" operation
1612 attribute is not supported and the "document-uri" attribute MUST be returned in the Unsupported
1613 Attributes group.

1614 server-error-operation-not-supported: the Send-URI operation is not supported.

1615

1616 3.1.3.2.3 Cancel-Job

1617 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Cancel-
1618 Job with the following specializations and differences. See Section 13 in [IPP-MOD] for a more complete
1619 description of each status code.

1620 For the following success status codes, the Job object is being canceled or has been canceled:

1621 successful-ok: no request attributes were substituted or ignored (same as Print-Job).

1622 successful-ok-ignored-or-substituted-attributes: same as Print-Job.

1623 successful-ok-conflicting-attributes: same as Print-Job.

1624

1625 For any of the error status codes, the Job object has not been canceled or was previously canceled.

1626 client-error-not-possible: The request cannot be carried out because of the state of the Job object
1627 ('completed', 'canceled', or 'aborted') or the state of the system.

1628 client-error-not-found: the target Printer and/or Job object does not exist.

1629 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding address is
1630 known.

1631 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.

1632 client-error-document-format-not-supported: not applicable.

1633 client-error-attributes-or-values-not-supported: not applicable, since unsupported operation attributes
1634 and values MUST be ignored.

1635 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-
1636 jobs" attribute is not involved.

1637 server-error-operation-not-supported: not applicable (Cancel-Job is REQUIRED).

1638 server-error-device-error: same as Print-Job, except no document data is involved.

1639 server-error-temporary-error: same as Print-Job, except no document data is involved.

1640 server-error-not-accepting-jobs: not applicable..

1641 server-error-job-canceled: not applicable.

1642 3.1.3.2.4 Get-Job-Attributes

1643 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Get-
1644 Job-Attributes with the following specializations and differences. See Section 13 in [IPP-MOD] for a more
1645 complete description of each status code.

1646 For the following success status codes, the requested attributes are returned in Group 3 in the response:

1647 successful-ok: no request attributes were substituted or ignored (same as Print-Job) and no requested
1648 attributes were unsupported.

1649 successful-ok-ignored-or-substituted-attributes: same as Print-Job, except the "requested-attributes"
1650 operation attribute MAY, but NEED NOT, be returned with the unsupported values.

1651 successful-ok-conflicting-attributes: same as Print-Job.

1652 For the error status codes, Group 3 is returned containing no attributes or is not returned at all.

1653 client-error-not-possible: Same as Print-Job, in addition the Printer object is not accepting any requests.

1654 client-error-document-format-not-supported: not applicable.

1655 client-error-attributes-or-values-not-supported: not applicable.

1656 client-error-uri-scheme-not-supported: not applicable.
1657 client-error-conflicting-attributes: not applicable
1658 server-error-operation-not-supported: not applicable (since Get-Job-Attributes is REQUIRED).
1659 server-error-device-error: same as Print-Job, except no document data is involved.
1660 server-error-temporary-error: sane as Print-Job, except no document data is involved..
1661 server-error-not-accepting-jobs: not applicable.
1662 server-error-job-canceled: not applicable.

1663 3.1.3.2.5 Hold-Job

1664 All of the Print-Job status codes described in Section 3.1.3.1.1 Print-Job Response are applicable to Hold-
1665 Job with the following specializations and differences. See Section 13 in [IPP-MOD] for a more complete
1666 description of each status code.

1667 For the following success status codes, the Job object is being held or has been held:

1668 successful-ok: no request attributes were substituted or ignored (same as Print-Job).
1669 successful-ok-ignored-or-substituted-attributes: same as Print-Job.
1670 successful-ok-conflicting-attributes: same as Print-Job.
1671

1672 For any of the error status codes, the Job object has not been held or was previously held.

1673 client-error-not-possible: The request cannot be carried out because of the state of the Job object
1674 ('completed', 'canceled', or 'aborted') or the state of the system.
1675 client-error-not-found: the target Printer and/or Job object does not exist.
1676 client-error-gone: the target Printer and/or Job object no longer exists and no forwarding address is
1677 known.
1678 client-error-request-entity-too-large: same as Print-Job, except no document data is involved.
1679 client-error-document-format-not-supported: not applicable.
1680 client-error-conflicting-attributes: same as Print-Job, except that the Printer's "printer-is-accepting-
1681 jobs" attribute is not involved.
1682 server-error-operation-not-supported: the Hold-Job operation is not supported.
1683 server-error-device-error: not applicable.
1684 server-error-temporary-error: same as Print-Job, except no document data is involved.
1685 server-error-not-accepting-jobs: not applicable.
1686 server-error-job-canceled: not applicable.

1687 3.1.3.2.6 Release-Job

1688 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
1689 specialization's described for Hold-Job are applicable to Release-Job. See Section 13 in [IPP-MOD] for a
1690 more complete description of each status code.

1691 server-error-operation-not-supported: the Release-Job operation is not supported.

1692 3.1.3.2.7 Restart-Job

1693 All of the Print-Job status code descriptions in Section 3.1.3.1.1 Print-Job Response with the
1694 specialization's described for Hold-Job are applicable to Restart-Job. See Section 13 in [IPP-MOD] for a
1695 more complete description of each status code.

1696 server-error-operation-not-supported: the Restart-Job operation is not supported.
1697

1698 3.1.3.2.7.1 Can documents be added to a restarted job?

1699 Assume I give a Create-Job request along with a set of 5 documents . All the documents get printed and the
1700 job state is moved to completed . I issue a Restart-Job request on the job. Now the issue is that, if I try to
1701 add new documents to the restarted job ,will the Ipp Server permit me to do so or return "client-error-not-
1702 possible " and again print those 5 jobs?

1703 A job can not move to the 'completed' state until all the documents have been processed. The 'last-
1704 document' flag indicates when the last document for a job is being sent from the client. This is the semantic
1705 equivalent of closing a job. No documents may be added once a job is closed. Section 3.3.7 of the IPP/1.1
1706 model states "The job is moved to the 'pending' job state and restarts the beginning on the same IPP Printer
1707 object with the same attribute values." 'number-of-documents' is a job attribute.

1710 3.1.4 Returning unsupported attributes in Get-Xxxx responses (Issue 1.18)

1711 In the Get-Printer-Attributes, Get-Jobs, or Get-Job-Attributes responses, the client cannot depend on getting
1712 unsupported attributes returned in the Unsupported Attributes group that the client requested, but are not
1713 supported by the IPP object. However, such unsupported requested attributes will not be returned in the
1714 Job Attributes or Printer Attributes group (since they are unsupported). Furthermore, the IPP object is
1715 REQUIRED to return the 'successful-ok-ignored-or-substituted-attributes' status code, so that the client
1716 knows that not all that was requested has been returned.

1717 3.1.5 Sending empty attribute groups (~~Issue 1.16~~)

1718 The [IPP-MOD] and [IPP-PRO] specifications RECOMMEND that a sender not send an empty attribute
1719 group in a request or a response. However, they REQUIRE a receiver to accept an empty attribute group as
1720 equivalent to the omission of that group. So a client SHOULD omit the Job Template Attributes group
1721 entirely in a create operation that is not supplying any Job Template attributes. Similarly, an IPP object
1722 SHOULD omit an empty Unsupported Attributes group if there are no unsupported attributes to be returned
1723 in a response.

1724 The [IPP-PRO] specification REQUIRES a receiver to be able to receive either an empty attribute group or
1725 an omitted attribute group and treat them equivalently. The term "receiver" means an IPP object for a
1726 request and a client for a response. The term "sender" means a client for a request and an IPP object for a
1727 response.

1728 There is an exception to the rule for Get-Jobs when there are no attributes to be returned. [~~ipp-pro~~IPP-
1729 ~~PRO~~] contains the following paragraph:

1730 The syntax allows an xxx-attributes-tag to be present when the xxx-attribute-sequence that follows is
1731 empty. The syntax is defined this way to allow for the response of Get-Jobs where no attributes are returned
1732 for some job-objects. Although it is RECOMMENDED that the sender not send an xxx-attributes-tag if
1733 there are no attributes (except in the Get-Jobs response just mentioned), the receiver MUST be able to
1734 decode such syntax.

1735 3.2 Printer Operations

1736 3.2.1 Print-Job operation

1737 3.2.1.1 Flow controlling the data portion of a Print-Job request (Issue 1.22)

1738 A paused printer, or one that is stopped due to paper out or jam or spool space full or buffer space full, may
1739 flow control the data of a Print-Job operation (at the TCP/IP layer), so that the client is not able to send all
1740 the document data. Consequently, the Printer will not return a response until the condition is changed.

1741 The Printer should not return a Print-Job response with an error code in any of these conditions, since either
1742 the printer will be resumed and/or the condition will be freed either by human intervention or as jobs print.

1743 In writing test scripts to test IPP Printers, the script must also be written not to expect a response, if the
1744 printer has been paused, until the printer is resumed, in order to work with all possible implementations.

1745 3.2.1.2 Returning job-state in Print-Job response (Issue 1.30)

1746 An IPP client submits a small job via Print-Job. By the time the IPP printer/print server is putting together
1747 a response to the operation, the job has finished printing and been removed as an object from the print
1748 system. What should the job-state be in the response?

1749 The Model suggests that the Printer return a response before it even accepts the document content. The Job
1750 Object Attributes are returned only if the IPP object returns one of the success status codes. Then the job-
1751 state would always be "pending" or "pending-held".

1752 This issue comes up for the implementation of an IPP Printer object as a server that forwards jobs to
1753 devices that do not provide job status back to the server. If the server is reasonably certain that the job
1754 completed successfully, then it should return the job-state as 'completed'. Also the server can keep the job
1755 in its "job history" long after the job is no longer in the device. Then a user could query the server and see
1756 that the job was in the 'completed' state and completed as specified by the jobs "time-at-completed" time,
1757 which would be the same as the server submitted the job to the device.

1758 An alternative is for the server to respond to the client before or while sending the job to the device, instead
1759 of waiting until the server has finished sending the job to the device. In this case, the server can return the
1760 job's state as 'pending' with the 'job-outgoing' value in the job's "job-state-reasons" attribute.

- 1761 If the server doesn't know for sure whether the job completed successfully (or at all), it could return the
1762 (out-of-band) 'unknown' value.
- 1763 On the other hand, if the server is able to query the device and/or setup some sort of event notification that
1764 the device initiates when the job makes state transitions, then the server can return the current job state in
1765 the Print-Job response and in subsequent queries because the server knows what the job state is in the
1766 device (or can query the device).
- 1767 All of these alternatives depend on implementation of the server and the device.
- 1768 3.2.2 Get-Printer-Attributes operation
- 1769 1 If a Printer supports the "printer-make-and-model" attribute and returns the .INF file model name of
1770 the printer in that attribute, the Microsoft client will automatically install the correct driver (if available).
- 1771 2 Clients which poll periodically for printer status or queued-job-count should use the "requested-
1772 attributes" operation attribute to limit the scope of the query in order to save Printer and network resources.
- 1773 3.2.3 Get-Jobs operation
- 1774 3.2.3.1 Get-Jobs, my-jobs='true', and 'requesting-user-name' (Issue 1.39)?
- 1775 In [~~ipp-mod~~IPP-MOD] section 3.2.6.1 'Get-Jobs Request', if the attribute 'my-jobs' is present and set to
1776 TRUE, MUST the 'requesting-user-name' attribute be there too, and if it's not present what should the IPP
1777 printer do?
- 1778 [~~ipp-mod~~IPP-MOD] Section 8.3 describes the various cases of "requesting-user-name" being present or not
1779 for any operation. If the client does not supply a value for "requesting-user-name", the printer MUST
1780 assume that the client is supplying some anonymous name, such as "anonymous".
- 1781 3.2.3.2 Why is there a "limit" attribute in the Get-Jobs operation?
- 1782 When using the Get-Jobs operation a client implementer might choose to limit the number of jobs that the
1783 client shows on the first screenful. For example, if its UI can only display 50 jobs, it can defend itself
1784 against a printer that would otherwise return 500 jobs, perhaps taking a long time on a slow dial-up line.
1785 The client can then go and ask for a larger number of jobs in the background, while showing the user the
1786 first 50 jobs. Since the job history is returned in reverse order, namely the most recently completed jobs are
1787 returned first, the user is most likely interested in the first jobs that are returned. Limiting the number of
1788 jobs may be especially useful for a client that is requesting 'completed' jobs from a printer that keeps a long
1789 job history. Clients that don't mind sometimes getting very large responses, can omit the "limit" attribute in
1790 their Get-Jobs requests.
- 1791 3.2.4 Create-Job operation

1792 A Printer may respond to a Create-Job operation with "job-state" 'pending' or 'pending-held' and "job-state-
1793 reason" 'job-data-insufficient' to indicate that operation has been accepted by the Printer, but the Printer is
1794 expecting additional document data before it can move the job into the 'processing' state. Alternatively, it
1795 may respond with "job-state" 'processing' and "job-state-reason" 'job-incoming' to indicate that the Create-
1796 Job operation has been accepted by the Printer, but the Printer is expecting additional Send-Document
1797 and/or Send-URI operations and/or is accessing/accepting document data. The second alternative is for
1798 non-spooling Printers that don't implement the 'pending' state.

1799 Should the server wait for the "last-document" operation attribute set to 'true' before starting to "process"
1800 the job?

1801 It depends on implementation. Some servers spool the entire job, including all document data, before
1802 starting to process, so such an implementation would wait for the "last-document" before starting to process
1803 the job. If the time-out occurs without the "last-document", then the server takes one of the indicated
1804 actions in section 3.3.1 in the [IPP-MOD] document. Other servers will start to process document data as
1805 soon as they have some. These are the so-called "non-spooling" printers. Currently, there isn't a way for a
1806 client to determine whether the Printer will spool all the data or will start to process (and print) as soon as it
1807 has some data.

1808 3.3 Job Operations

1809 3.3.1 Validate-Job

1810 The Validate-Job operation has been designed so that its implementation may be a part of the Print-Job
1811 operation. Therefore, requiring Validate-Job is not a burden on implementers. Also it is useful for client's
1812 to be able to count on its presence in all conformance implementations, so that the client can determine
1813 before sending a long document, whether the job will be accepted by the IPP Printer or not.

1814 3.3.2 Restart-Job

1815 The Restart-Job operation allows the reprocessing of a completed job. Some jobs store the document data
1816 on the printer. Jobs created using the Print-Job operation are an example. It is required that the printer
1817 retains the job data after the job has moved to a 'completed state' in order for the Restart-Job operation to
1818 succeed.

1819 Some jobs contain only a reference to the job data. A job created using the Print-URI is an example of such
1820 a job. When the Restart-Job operation is issued the job is reprocessed. The job data MUST be retrieved
1821 again to print the job.

1822 It is possible that a job fails while attempting to access the print data. When such a job is the target of a
1823 Restart-Job the Printer SHALL attempt to retrieve the job data again.

1824 4 Object Attributes

1825 4.1 Attribute Syntax's

1826 4.1.1 The 'none' value for empty sets (Issue 1.37)

1827 [~~ipp-mod~~IPP-MOD] states that the 'none' value should be used as the value of a 1setOf when the set is
1828 empty. In most cases, sets that are potentially empty contain keywords so the keyword 'none' is used, but for
1829 the 3 finishings attributes, the values are enums and thus the empty set is represented by the enum 3.
1830 Currently there are no other attributes with 1setOf values, which can be empty and can contain values that
1831 are not keywords. This exception requires special code and is a potential place for bugs. It would have
1832 been better if we had chosen an out-of-band value, either "no-value" or some new value, such as 'none'.
1833 Since we didn't, implementations have to deal with the different representations of 'none', depending on the
1834 attribute syntax.

1835 4.1.2 Multi-valued attributes (Issue 1.31)

1836 What is the attribute syntax for a multi-valued attribute? Since some attributes support values in more than
1837 one data type, such as "media", "job-hold-until", and "job-sheets", IPP semantics associate the attribute
1838 syntax with each value, not with the attribute as a whole. The protocol associates the attribute syntax tag
1839 with each value. Don't be fooled, just because the attribute syntax tag comes before the attribute keyword.
1840 All attribute values after the first have a zero length attribute keyword as the indication of a subsequent
1841 value of the same attribute.

1842 4.1.3 Case Sensitivity in URIs (issue 1.6)

1843 IPP client and server implementations must be aware of the diverse uppercase/lowercase nature of URIs.
1844 RFC 2396 defines URL schemes and Host names as case insensitive but reminds us that the rest of the URL
1845 may well demonstrate case sensitivity. When creating URL's for fields where the choice is completely
1846 arbitrary, it is probably best to select lower case. However, this cannot be guaranteed and implementations
1847 MUST NOT rely on any fields being case-sensitive or case-insensitive in the URL beyond the URL scheme
1848 and host name fields.

1849 The reason that the IPP specification does not make any restrictions on URIs, is so that implementations of
1850 IPP may use off-the-shelf components that conform to the standards that define URIs, such as RFC 2396
1851 and the HTTP/1.1 specifications [RFC2616]. See these specifications for rules of matching, comparison,
1852 and case-sensitivity.

1853 It is also recommended that System Administrators and implementations avoid creating URLs for different
1854 printers that differ only in their case. For example, don't have Printer1 and printer1 as two different IPP
1855 Printers.

1856 Example of equivalent URI's

1857 `http://abc.com:80/~smith/home.html`

1858 `http://ABC.com/%7Esmith/home.html`

1859 http://ABC.com:/%7esmith/home.html

1860 Example of equivalent URI's using the IPP scheme

1861 ipp://abc.com:631/~smith/home.html

1862 ipp://ABC.com/%7Esmith/home.html

1863 http://ABC.com:631/%7esmith/home.html

1864 The HTTP/1.1 specification [RFC2616] contains more details on comparing URLs.

1865 4.1.4 Maximum length for xxxWithLanguage and xxxWithoutLanguage

1866 The 'textWithLanguage' and 'nameWithLanguage' are compound syntaxes that have two components. The
1867 first component is the 'language' component that can contain up to 63 octets. The second component is the
1868 'text' or 'name' component. The maximum length of these are 1023 octets and 255 octets respectively.
1869 The definition of attributes with either syntax may further restrict the length. (e.g. printer-name
1870 (name(127)))

1871 The length of the 'language' component has no effect on the allowable length of 'text' in
1872 'textWithLanguage' or the length of 'name' in 'nameWithLanguage'

1873 4.2 Job Template Attributes

1874 4.2.1 multiple-document-handling(type2 keyword)

1875 4.2.1.1 Support of multiple document jobs

1876 ISSUE: IPP/1.0 is silent on which of the four effects an implementation would perform if it supports
1877 Create-Job, but does not support "multiple-document-handling".

1878 A fix to IPP/1.0 would be to require implementing all four values of "multiple-document-handling" if
1879 Create-Job is supported at all. Or at least 'single-document-new-sheet' and 'separate-documents-uncollated-
1880 copies'. In any case, an implementation that supports Create-Job SHOULD also support "multiple-
1881 document-handling". Support for all four values is RECOMMENDED, but at least the 'single-document-
1882 new-sheet' and 'separate-documents-uncollated-copies' values, along with the "multiple-document-handling-
1883 default" indicating the default behavior and "multiple-document-handling-supported" values. If an
1884 implementation spools the data, it should also support the 'separate-documents-collated-copies' value as
1885 well.

1886 4.3 Job Description Attributes

1887 The time-at-creation, time-at-processing, and time-at-completed attributes may be returned in integer time
1888 ticks or absolute dateTime syntax. There are various ways for a Printer to get the time of day. Some
1889 suggestions:

- 1890 1. A Printer can get time from an NTP timeserver if there's one reachable on the network. See
1891 RFC 1305. Also DHCP option 32 in RFC 2132 returns the IP address of the NTP server.
- 1892 2. Get the date and time at startup from a human operator
- 1893 3. Have an operator set the date and time using a web administrative interface
- 1894 4. Get the date and time from incoming HTTP requests, though the problems of spoofing need
1895 to be considered. Perhaps comparing several HTTP requests could reduce the chances of spoofing.
- 1896 5. Internal date time clock battery driven.
- 1897 6. Query "<http://tycho.usno.navy.mil/cgi-bin/timer.pl>"

1898 4.4 Printer Description Attributes

1899 4.4.1 queued-job-count

1900 4.4.1.1 Why is "queued-job-count" RECOMMENDED (Issue 1.14)?

1901 The reason that "queued-job-count" is RECOMMENDED, is that some clients look at that attribute alone
1902 when summarizing the status of a list of printers, instead of doing a Get-Jobs to determine the number of
1903 jobs in the queue. Implementations that fail to support the "queued-job-count" will cause that client to
1904 display 0 jobs when there are actually queued jobs.

1905 We would have made it a REQUIRED Printer attribute, but some implementations had already been
1906 completed before the issue was raised, so making it a SHOULD was a compromise.

1907 4.4.1.2 Is "queued-job-count" a good measure of how busy a printer is (Issue 1.15)?

1908 The "queued-job-count" is not a good measure of how busy the printer is when there are held jobs. A future
1909 registration could be to add a "held-job-count" (or an "active-job-count") Printer Description attribute if
1910 experience shows that such an attribute (combination) is needed to quickly indicate how busy a printer
1911 really is.

1912 4.4.2 printer-current-time (dateTime)

1913 A Printer implementation MAY support this attribute by obtaining the date and time by any number of
1914 implementation-dependent means at startup or subsequently. Examples include:

- 1915 (1) an internal date time clock,
1916 (2) from the operator at startup using the console,
1917 (3) from an operator using an administrative web page,
1918 (4) from HTTP headers supplied in client requests,
1919 (5) use HTTP to query "<http://tycho.usno.navy.mil/cgi-bin/timer.pl>"
1920 (6) from the network, using NTP [RFC1305] or DHCP option 32 [RFC2132] that returns the IP address
1921 of the NTP server.

1922 If an implementation supports this attribute by obtaining the current time from the network (at startup or
1923 later), but the time is not available, then the implementation **MUST** return the value of this attribute using
1924 the out-of-band 'no-value' meaning not configured. See the beginning of section 4.1.

1925 Since the new "date-and-time-at-xxx" Job Description attributes refer to the "printer-current-time", they
1926 will be covered also.

1927 4.4.3 Printer-uri

1928 Must the operational attribute for printer-uri match one of the values in printer-uri-supported?

1929 A forgiving printer implementation would not reject the operation. But the implementation has its rights to
1930 reject a printer or job operation if the operational attribute printer-uri is not a value of the printer-uri-
1931 supported. The printer may not be improperly configured. The request obviously reached the printer. The
1932 printer could treat the printer-uri as the logical equivalent of a value in the printer-uri-supported. It would
1933 be implementation dependent for which value, and associated security policy, would apply. This does also
1934 apply to a job object specified with a printer-uri and job-id, or with a job-uri. See section 4.1.3 for how to
1935 compare URI's.

1936 4.5 Empty Jobs

1937 The IPP object model does not prohibit a job that contains no documents. Such a job may be created in a
1938 number of ways including a 'create-job' followed by an 'add-document' that contains no data and has the
1939 'last-document' flag set.

1940 An empty job is processed just as any other job. The operation that "closes" an empty job is not rejected
1941 because the job is empty. If no other conditions exist, other than the job is empty, the response to the
1942 operation will indicate success. After the job is scheduled and processed, the job state SHALL be
1943 'completed'

1944 There will be some variation in the value(s) of the 'job-state-reasons' attribute. It is required that if no
1945 conditions, other than the job being empty, exist the 'job-state-reasons' SHALL include the 'completd-

1946 successfully'. If other conditions existed, the 'completed-with-warnings' or 'completed-with-errors' values
1947 may be used."

1948 **5 DNS Considerations**

1949 5.1 IPP Printer with a DNS name

1950 If the IPP printer has a DNS name should there be at least two values for the printer-uri-supported attribute.
1951 One URL with the fully qualified DNS name the other with the IP address in the URL?

1952 The printer may contain one or the other or both. It's up to the administrator to configure this attribute.

1953 **6 Security Considerations**

1954 This section corresponds to the IPP-MOD Section 8 "Security Considerations.

1955 6.1 Querying jobs with IPP that were submitted using other job submission protocols (Issue 1.32)

1956 The following clarification was added to [~~ipp-mod~~IPP-MOD] section 8.5:

1957 8.5 Queries on jobs submitted using non-IPP protocols

1958 If the device that an IPP Printer is representing is able to accept jobs using other job submission protocols
1959 in addition to IPP, it is RECOMMEND that such an implementation at least allow such "foreign" jobs to be
1960 queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an implementation NEED NOT
1961 support all of the same IPP job attributes as for IPP jobs. The IPP object returns the 'unknown' out-of-band
1962 value for any requested attribute of a foreign job that is supported for IPP jobs, but not for foreign jobs.

1963 It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such "foreign
1964 jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-Attributes and
1965 Cancel-Job. Such an implementation also needs to deal with the problem of authentication of such foreign
1966 jobs. One approach would be to treat all such foreign jobs as belonging to users other than the user of the
1967 IPP client. Another approach would be for the foreign job to belong to 'anonymous'. Only if the IPP client
1968 has been authenticated as an operator or administrator of the IPP Printer object, could the foreign jobs be
1969 queried by an IPP request. Alternatively, if the security policy were to allow users to query other users'
1970 jobs, then the foreign jobs would also be visible to an end-user IPP client using Get-Jobs and Get-Job-
1971 Attributes.

1972 Thus IPP MAY be implemented as a "universal" protocol that provides access to jobs submitted with any
1973 job submission protocol. As IPP becomes widely implemented, providing a more universal access makes
1974 sense.

1975 **7 Encoding and Transport**

1976 This section discusses various aspects of IPP/1.1 Encoding and Transport [IPP-PRO].

1977 A server is not required to send a response until after it has received the client's entire request. Hence, a
1978 client must not expect a response until after it has sent the entire request. However, we recommend that the
1979 server return a response as soon as possible if an error is detected while the client is still sending the data,
1980 rather than waiting until all of the data is received. Therefore, we also recommend that a client listen for an
1981 error response that an IPP server MAY send before it receives all the data. In this case a client, if chunking
1982 the data, can send a premature zero-length chunk to end the request before sending all the data (and so the
1983 client can keep the connection open for other requests, rather than closing it). If the request is blocked for
1984 some reason, a client MAY determine the reason by opening another connection to query the server using
1985 Get-Printer-Attributes.

1986 IPP, by design, uses TCP's built-in flow control mechanisms [RFC 793] to throttle clients when Printers are
1987 busy. Therefore, it is perfectly normal for an IPP client transmitting a Job to be blocked for a really long
1988 time. Accordingly, socket timeouts must be avoided. Some socket implementations have a timeout option,
1989 which specifies how long a write operation on a socket can be blocked before it times out and the blocking
1990 ends. A client should set this option for infinite timeout when transmitting Job submissions.

1991 Some IPP client applications might be able to perform other useful work while a Job transmission is
1992 blocked. For example, the client may have other jobs that it could transmit to other Printers simultaneously.
1993 A client may have a GUI, which must remain responsive to the user while the Job transmission is blocked.
1994 These clients should be designed to spawn a thread to handle the Job transmission at its own pace, leaving
1995 the main application free to do other work. Alternatively, single-threaded applications could use non-
1996 blocking I/O.

1997 Some Printer conditions, such as jam or lack of paper, could cause a client to be blocked indefinitely.
1998 Clients may open additional connections to the Printer to Get-Printer-Attributes, determine the state of the
1999 device, alert a user if the printer is stopped, and let a user decide whether to abort the job transmission or
2000 not.

2001 In the following sections, there are tables of all HTTP headers, which describe their use in an IPP client or
2002 server. The following is an explanation of each column in these tables.

- 2003 • the "header" column contains the name of a header
- 2004 • the "request/client" column indicates whether a client sends the header.
- 2005 • the "request/ server" column indicates whether a server supports the header when received.
- 2006 • the "response/ server" column indicates whether a server sends the header.
- 2007 • the "response /client" column indicates whether a client supports the header when received.
- 2008 • the "values and conditions" column specifies the allowed header values and the conditions for the
2009 header to be present in a request/response.

2010 The table for "request headers" does not have columns for responses, and the table for "response headers"
2011 does not have columns for requests.

2012 The following is an explanation of the values in the "request/client" and "response/ server" columns.

- 2013 • **must:** the client or server MUST send the header,
- 2014 • **must-if:** the client or server MUST send the header when the condition described in the "values and
2015 conditions" column is met,

- 2016 • **may:** the client or server MAY send the header
- 2017 • **not:** the client or server SHOULD NOT send the header. It is not relevant to an IPP implementation.

2018 The following is an explanation of the values in the "response/client" and "request/ server" columns.

- 2019 • **must:** the client or server MUST support the header,
- 2020 • **may:** the client or server MAY support the header
- 2021 • **not:** the client or server SHOULD NOT support the header. It is not relevant to an IPP
- 2022 implementation.

2023 7.1 General Headers

2024 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 | |

2025 7.2 Request Headers

2026 The following is a table for the request headers.

| Request-Header | Client | Server | Request Values and Conditions |
|----------------|--------|--------|--|
| Accept | may | must | "application/ipp" only. This value is the default if the client omits it |
| Accept-Charset | not | not | Charset information is within the application/ipp |

| Request-Header | Client | Server | Request Values and Conditions |
|-----------------------|---------------|---------------|---|
| Accept-Encoding | may | must | entity 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 | |

2027 7.3 Response Headers

2028 The following is a table for the request headers.

| Response-Header | Server | Client | Response Values and Conditions |
|------------------------|---------------|---------------|---------------------------------------|
| Accept-Ranges | not | not | |
| Age | not | not | |

| Response-Header | Server | Client | Response Values and Conditions |
|------------------------|---------------|---------------|---|
| 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. |

2029 7.4 Entity Headers

2030 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 | |

| Entity-Header | Request | | Response | | Values and Conditions |
|---------------|---------|--------|----------|--------|-----------------------|
| | Client | Server | Server | Client | |
| Expires | not | not | not | not | |
| Last-Modified | not | not | not | not | |

2031 7.5 Optional support for HTTP/1.0

2032 IPP implementations consist of an HTTP layer and an IPP layer. In the following discussion, the term
 2033 "client" refers to the HTTP client layer and the term "server" refers to the HTTP server layer. The Encoding
 2034 and Transport document [IPP-PRO] requires that HTTP 1.1 **MUST** be supported by all clients and all
 2035 servers. However, a client and/or a server implementation may choose to also support HTTP 1.0.

2036 • This option means that a server may choose to communicate with a (non-conforming) client that only
 2037 supports HTTP 1.0. In such cases the server should not use any HTTP 1.1 specific parameters or features
 2038 and should respond using HTTP version number 1.0.

2039 • This option also means that a client may choose to communicate with a (non-conforming) server that
 2040 only supports HTTP 1.0. In such cases, if the server responds with an HTTP 'unsupported version number'
 2041 to an HTTP 1.1 request, the client should retry using HTTP version number 1.0.

2042 7.6 HTTP/1.1 Chunking

2043 7.6.1 Disabling IPP Server Response Chunking

2044 Clients **MUST** anticipate that the HTTP/1.1 server may chunk responses and **MUST** accept them in
 2045 responses. However, a (non-conforming) HTTP client that is unable to accept chunked responses may
 2046 attempt to request an HTTP 1.1 server not to use chunking in its response to an operation by using the
 2047 following HTTP header:

2048 TE: identity

2049 This mechanism should not be used by a server to disable a client from chunking a request, since chunking
 2050 of document data is an important feature for clients to send long documents.

2051 7.6.2 Warning About the Support of Chunked Requests

2052 This section describes some problems with the use of chunked requests and HTTP/1.1 servers.

2053 The HTTP/1.1 standard [RFC2616] requires that conforming servers support chunked requests for any
 2054 method. However, in spite of this requirement, some HTTP/1.1 implementations support chunked
 2055 responses in the GET method, but do not support chunked POST method requests. Some HTTP/1.1
 2056 implementations that support CGI scripts [CGI] and/or servlets [Servlet] require that the client supply a
 2057 Content-Length. These implementations might reject a chunked POST method and return a 411 status code

2058 (Length Required), might attempt to buffer the request and run out of room returning a 413 status code
2059 (Request Entity Too Large), or might successfully accept the chunked request.

2060 Because of this lack of conformance of HTTP servers to the HTTP/1.1 standard, the IPP standard [IPP-
2061 PRO] REQUIRES that a conforming IPP Printer object implementation support chunked requests and that
2062 conforming clients accept chunked responses. Therefore, IPP object implementers are warned to seek
2063 HTTP server implementations that support chunked POST requests in order to conform to the IPP standard
2064 and/or use implementation techniques that support chunked POST requests.

2065 **8 References**

2066 [CGI]

2067 CGI/1.1 (<http://www.ietf.org/internet-drafts/draft-coar-cgi-v11-00.txt>).

2068 [IPP-MOD]

2069 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
2070 Semantics", draft-ietf-ipp-model-v11-06.txt, March 1, ~~2000~~, ~~2000~~.

2071 [IPP-PRO]

2072 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and
2073 Transport", draft-ietf-ipp-protocol-v11-05.txt, March 1, 2000.

2074 [RFC 793]

2075 J. Postel, "Transmission Control Protocol", RFC 793.

2076 [RFC1123]

2077 Braden, S., "Requirements for Internet Hosts - Application and Support", RFC 1123, October, 1989.

2078 [RFC2026]

2079 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

2080 [~~RFC~~~~fe~~2119]

2081 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March
2082 1997.

2083 [RFC2396]

2084 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic
2085 Syntax", RFC 2396, August 1998.

2086 [RFC2565]

2087 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
2088 Semantics", RFC 2566, April 1999.

2089 [RFC2566]

2090 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and
2091 Transport", RFC 2565, April 1999.

- 2092 [RFC2567]
2093 Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-03.txt, November,
2094 1998.
- 2095 [RFC2568]
2096 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
2097 RFC 2568, April 1999.
- 2098 [RFC2569]
2099 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC
2100 2569, April 1999.
- 2101 [RFC2616]
2102 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext
2103 Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
- 2104 [Servlet]
2105 Servlet Specification Version 2.1 (<http://java.sun.com/products/servlet/2.1/index.html>).
- 2106 [SSL]
2107 Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.
- 2108 8.1 Authors' Address
- 2109 Thomas N. Hastings
2110 Xerox Corporation
2111 701 Aviation Blvd.
2112 El Segundo, CA 90245
2113 hastings@cp10.es.xerox.com
2114
- 2115 Carl-Uno Manros
2116 Xerox Corporation
2117 701 Aviation Blvd.
2118 El Segundo, CA 90245
2119 manros@cp10.es.xerox.com
2120
- 2121 Carl Kugler
2122 Mail Stop 003G
2123 IBM Printing Systems Co
2124 6300 Diagonal Hwy
2125 Boulder CO80301
2126 Kugler@us.ibm.com
2127
- 2128 Henrik Holst
2129 i-data Printing Systems

2130 Vadstrupvej 35-43
2131 2880 Bagsvaerd, Denmark
2132 hh@l-data.com

2133
2134 [Peter Zehler](#)
2135 [Xerox Corporation](#)
2136 [800 Philipd Road](#)
2137 [Webster, NY 14580](#)

2138
2139 [Phone: 716 265-8755](tel:7162658755)
2140 peter.zehler@usa.xerox.com

2141 **9 Notices**

2142 The IETF takes no position regarding the validity or scope of any intellectual property or other rights that
2143 might be claimed to pertain to the implementation or use of the technology described in this document or
2144 the extent to which any license under such rights might or might not be available; neither does it represent
2145 that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to
2146 rights in standards-track and standards-related documentation can be found in BCP-11[BCP-11]. Copies of
2147 claims of rights made available for publication and any assurances of licenses to be made available, or the
2148 result of an attempt made to obtain a general license or permission for the use of such proprietary rights by
2149 implementers or users of this specification can be obtained from the IETF Secretariat.

2150 The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications,
2151 or other proprietary rights, which may cover technology that may be required to practice this standard.
2152 Please address the information to the IETF Executive Director.

2153 Copyright (C) The Internet Society (1999). All Rights Reserved

2154 This document and translations of it may be copied and furnished to others, and derivative works that
2155 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
2156 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
2157 this paragraph are included on all such copies and derivative works. However, this document itself may not
2158 be modified in any way, such as by removing the copyright notice or references to the Internet Society or
2159 other Internet organizations, except as needed for the purpose of developing Internet standards in which
2160 case the procedures for copyrights defined in the Internet Standards process must be followed, or as
2161 required to translate it into languages other than English.

2162 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its
2163 successors or assigns.

2164 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET
2165 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,
2166 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE
2167 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
2168 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

2169 **10 Change History (to be removed at time of RFC publishing)**

2170 The change history is in *reverse* chronological order:

2171 10.1 Changes from 990914 to 990927

- 2172 1. Add comments about this document is also IPP/1.0 relevant.
- 2173 2. Section 4.1.3: Add more examples of URI's with the port 631 and the ipp scheme.
- 2174 3. Section 4.4.3: Move the DNS stuff to the 'how to compare URI's.
- 2175 4. Section 4.4.3.2: Swap lines, first tell about the forgiven printer and then what the printer is allowed to
2176 do.
- 2177 5. Fixed some errors in the Summary Attribute tables 1-5 and broke them into five portrait tables, so that it
2178 can be made into plain text for INTERNET-DRAFTS.

2179 10.2 Changes from 990726 to 990914:

- 2180 1. Added IPP/1.1 operations and attributes to table 1.
- 2181 2. Validate version: Added text and table from issue 32
- 2182 3. Printer-uri-supported: Added section 4.4.3
- 2183 4. Added IPP/1.1 operations to section 3.1.2.1.4.3
- 2184 5. Added answer to question "Should the server wait for the "last-document" operation attribute set to
2185 'true' before starting to "process" the job?" in section 3.2.4
- 2186 6. Changed 'server-error-uri-scheme-not-supported' to 'client-error-uri-scheme-not-supported' in section
2187 3.1.2.1.5 when talking about the 'document-uri' attribute.
- 2188 7. Added 'Suggested Operation Processing Steps' and 'Suggested Additional Processing Steps for
2189 Operations that Create/Validate Jobs and Add Document' flow-chart overview.

2190 10.3 Changes to produce the February 12, 1999 version from the January 8, 1999 version:

- 2191 1. Section 2.2.1.5: added check for document not found or accessible in Print-URI and Send-URI
- 2192 2. Section 3.6.2: Clarified that the IPP standard requires that servers **MUST** accept chunked requests
2193 and that clients **MUST** accept chunked responses, in spite of the lack of conformance of HTTP
2194 servers to the HTTP/1.1 requirement to support chunking.

- 2195 10.4 Changes to produce the January 8, 1999 version from the December 6, 1998 version:
- 2196 1. Added section 3.6.2: Warning About the Use of Chunked Requests with CGI Script
2197 Implementations
 - 2198 2. Section 2.2.1.2: changed "printer-operations-supported" to "operations-supported".
 - 2199 3. Section 2.2.1.6: changed "job-media-supported" to "job-media-sheets-supported"
 - 2200 4. Section 2.2.3: separated the validation checks for variable length attributes into two separate tests:
2201 one for correct attribute syntax and one for correct length.
 - 2202 5. Section 2.2.3: changed "multiple-document-handling-supported" to "printer-resolution-supported"
 - 2203 6. Section 2.6.1: recommended that an IPP object also support US-ASCII charset.
 - 2204 7. Section 3: Clarified that a server is not required to send a response until after it has received the
2205 client's entire request, but recommend that the server return a response as soon as possible if an error
2206 is detected while the client is still sending the data, rather than waiting until all of the data is
2207 received. Also recommended that a client listen for an error response that an IPP server MAY send
2208 before it receives all the data.
- 2209 10.5 Changes to produce the December 6, 1998 version from the November 16, 1998 version:
- 2210 Included all of the remaining agreed issues raised before the November 16, 1998 production of the Internet-
2211 Drafts for IPP/1.0 that included adding explanations to the Implementers Guide.
- 2212 Changes from 990422 to 990726:
- 2213 1. Encoding and Transport: Address issues 4, 5, 20 from Issues-raised-at-Bake-Off2.doc
 - 2214 2. Decide whether to accept or reject the request: discuss issues 6, 9, 10
 - 2215 3. Get-Printer-Attributes: add notes about printer-make-and-model and .INF files; issue 7
 - 2216 4. Create-Job: clarify job-incoming vs. data-insufficient; issue 13
 - 2217 5. Get-Printer Attributes: polling -- issue 16
 - 2218 6. Job Description Attributes: ways to get time; issue 17
 - 2219 7. Validate the values of the Job Template Attributes: clarify zero-length keywords; issue 22
 - 2220 8. Validate Optional Operation Attributes: Note about checking for compression in IPP/1.0; issue 28
 - 2221 9. Validate version number: advantages to backward compatibility; issue 33

2222 10. Note: examples for issue 2 seem to be covered sufficiently in the new MOD doc.