

1 INTERNET-DRAFT 5 ISSUES are highlighted like this.
2 <draft-ietf-ipp-notify-poll-00.txt>

Carl-Uno Manros
Tom Hastings
Robert Herriot
Xerox Corp.
Harry Lewis
IBM, Corp.
February 2 March 6, 2000

9 Internet Printing Protocol/1.1: The '~~ipp-notify-poll~~ipp' Notification Polling Method

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

11 Status of this Memo

12 This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of
13 [rfc2026]. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas,
14 and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

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

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

20 **Abstract**

21 The IPP notification specification [ipp-ntfy] is an OPTIONAL extension to IPP/1.0 and IPP/1.1 that
22 requires the definition of one or more delivery methods for dispatching event notification reports to
23 Notification Recipients. This document describes the semantics and syntax of the '~~ipp-notify-poll~~ipp' event
24 notification delivery method. For this delivery method, the client uses an explicit IPP Get-Notifications
25 Printer operation in order to request (pull) event Notifications from the IPP Printer.

26 When a Printer supports the '~~ipp-notify-poll~~ipp' delivery method, it ~~queues~~ holds each Notification events
27 for a window of time for each Subscription object for a certain length of time. The amount of time is called
28 the "lease time" and it is the same for all events in a Printer. If a Notification Recipient does not want to
29 miss events, the time between consecutive pollings of Notification Recipients poll these Subscription
30 objects at the rate specified by the time window must be less than the lease time. The Get-Notifications
31 request indicates whether the client wants to receive all pending events Notifications for (1) any
32 Subscription for which the client is the owner, (2) any Subscription associated with a Job or (23) a
33 particular Subscription object. ~~The Get-Notifications operation retrieves all pending Notifications that~~
34 ~~occurred for an interval of time in the past for the requested Subscription objects.~~ With the Get-
35 Notifications operation, tThe Printer returns the all pending existing Notifications along with two time
36 intervals. One specifies the length of the lease for all future events and the other specifies the recommended
37 interval to wait to the next Get-Notifications operation. The second time interval is less than the first. ~~that~~

38 ~~specify the next time window: one is the minimum interval that the client should wait before performing~~
39 ~~another Get-Notifications on the subscription-id and the other is the maximum interval that the Printer is~~
40 ~~guaranteed to keep any new Notifications associated with the subscription-id.~~

41 The Printer may keep the channel open if the ~~minimum~~recommended interval is sufficiently short, but in
42 any case the client performs a new Get-Notifications operation each time it wants more Notifications.
43 Since the time interval between consecutive client requests is normally less than the lease time,~~will be~~
44 ~~making Get-Notification requests before the time window expires,~~consecutive responses will normally
45 contain some events that are identical.~~the Printer will, on occasion, return the same event Notification in~~
46 ~~two successive responses.~~The later ones in the previous response will become the earliest in the next
47 response. The client is expected to filter out these duplicates which is easy to do because of the sequence
48 number in each Notification.

49 The full set of IPP documents includes:

- 50 Design Goals for an Internet Printing Protocol [RFC2567]
- 51 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
- 52 Internet Printing Protocol/1.1: Model and Semantics [ipp-mod]
- 53 Internet Printing Protocol/1.1: Encoding and Transport [ipp-pro]
- 54 Internet Printing Protocol/1.1: Implementer's Guide [ipp-iig]
- 55 Mapping between LPD and IPP Protocols [RFC2569]
- 56 Internet Printing Protocol/1.0 & 1.1: Event Notification Specification [ipp-ntfy]

57

58 The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed printing
59 functionality, and it enumerates real-life scenarios that help to clarify the features that need to be included
60 in a printing protocol for the Internet. It identifies requirements for three types of users: end users,
61 operators, and administrators. It calls out a subset of end user requirements that are satisfied in IPP/1.0. A
62 few OPTIONAL operator operations have been added to IPP/1.1.

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

67 The "Internet Printing Protocol/1.1: Model and Semantics" document describes a simplified model with
68 abstract objects, their attributes, and their operations that are independent of encoding and transport. It
69 introduces a Printer and a Job object. The Job object optionally supports multiple documents per Job. It
70 also addresses security, internationalization, and directory issues.

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

76 The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to
77 implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some of the
78 considerations that may assist them in the design of their client and/or IPP object implementations. For
79 example, a typical order of processing requests is given, including error checking. Motivation for some of
80 the specification decisions is also included.

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

83 The "Event Notification Specification" document defines OPTIONAL operations that allow a client to
84 subscribe to printing related events. Subscriptions include "Per-Job subscriptions" and "Per-Printer
85 subscriptions". Subscriptions are modeled as Subscription objects. Four other operations are defined for
86 subscription objects: get attributes, get subscriptions, renew a subscription, and cancel a subscription.

87

88

Table of Contents

89	1	Introduction	5
90	2	Terminology	6
91	3	Model and Operation	6
92	4	Get-Notifications operation	7
93	4.1	GET-NOTIFICATIONS REQUEST	7
94	4.2	GET-NOTIFICATIONS RESPONSE	9
95	5	Extension to Print-Job, Print-URI, Create-Job, Create-Printer-Subscription and Create-Printer-Subscription	10
96			
97	5.1	RESPONSE	10
98	6	Encoding	11
99	7	IANA Considerations	12
100	8	Internationalization Considerations	12
101	9	Security Considerations	12
102	10	References	12
103	11	Author's Addresses	13
104	12	Full Copyright Statement	13
105			

106

107

1 Introduction

108 IPP printers that support the OPTIONAL IPP notification extension [ipp-ntfy] either a) accept, store, and
109 use notification subscriptions to generate event Notification reports and implement one or more delivery
110 methods for notifying interested parties, or b) support a subset of these tasks and farm out the remaining
111 tasks to a Notification Delivery Service. The '~~ipp-notify-poll~~ipp' event notification delivery method
112 specified in this document defines a Get-Notifications operation that may be used in a variety of
113 notification scenarios. Its primary intended use is for clients that want to be Notification Recipients.
114 However, the Get-Notifications operation may also be used by Notification Delivery Services for
115 subsequent distribution to the Ultimate Notification Recipients.

116 When a Printer supports the '~~ipp-notify-poll~~ipp' delivery method, it ~~queues~~ holds each Notification events
117 for a certain length of time ~~for a window of time for each Subscription object~~. The amount of time is called
118 the "lease time" and it is the same for all events in a Printer. If a Notification Recipient does not want to
119 miss events, the time between consecutive pollings of Subscription objects must be less than the lease time.
120 ~~Notification Recipients poll these Subscription objects at the rate specified by the time window.~~ The Get-
121 Notifications request indicates whether the client wants to receive all pending events Notifications for (1)
122 any Subscription for which the client is the owner, (2) any Subscription associated with a particular Job or
123 (23) a particular Subscription object. With the Get-Notifications operation, the Printer returns all existing
124 Notifications along with two time intervals. One specifies the length of the lease for all future events and
125 the other specifies the recommended interval to wait to the next Get-Notifications operation. The second
126 time interval is less than the first. ~~The Get-Notifications operation retrieves all pending Notifications that~~
127 ~~occurred for an interval of time in the past for the requested Subscription objects. The Printer returns all~~
128 ~~pending Notifications along with two time intervals that specify the next time window: one is the minimum~~
129 ~~interval that the client should wait before performing a Get-Notifications on the subscription id and the~~
130 ~~other is the maximum interval that the Printer is guaranteed to keep any new Notifications associated with~~
131 ~~the subscription id.~~

132 The Printer may keep the channel open- if the recommended interval is sufficiently short ~~if the minimum~~
133 ~~interval is sufficiently short~~, but in any case the client performs a new Get-Notifications operation each time
134 it wants more Notifications. Since the time interval between consecutive client requests is normally less
135 than the lease time, consecutive responses will normally contain some events that are identical. ~~will be~~
136 ~~making Get Notification requests before the time window expires, the Printer will, on occasion, return the~~
137 ~~same event Notification in two successive responses.~~ The later ones in the previous response will become
138 the earliest in the next response. The client is expected to filter out these duplicates, which is easy to do
139 because of the sequence number in each Notification. The reason for not removing the Notifications from
140 the Subscription object with every Get-Notifications request, is so that multiple Notification Recipients can
141 be polling the same subscription object and so the Get-Notification operation satisfies the rule of
142 idempotency. ~~This~~ The former is useful if ~~you someone is are~~ logged in to several desktops at the same
143 time and ~~wants~~ to see the same events at both places. The latter is useful if the network loses the response.

144 2 Terminology

145 This section defines the following additional terms that are used throughout this document:

146 REQUIRED: if an implementation supports the extensions described in this document, it MUST
147 support a REQUIRED feature.

148 OPTIONAL: if an implementation supports the extensions described in this document, it MAY support
149 an OPTIONAL feature.

150 Notification Recipient - See [ipp-ntfy]

151 Subscription object - See [ipp-ntfy]

152 Ultimate Notification Recipient - See [ipp-ntfy]

153 3 Model and Operation

154 In the IPP Notification Model [ipp-ntfy], one or more Per-Job Subscriptions can be supplied in the Job
155 Creation operation or OPTIONALLY as subsequent Create-Job-Subscription operations; one Per-Printer
156 Subscription can be supplied in the Create-Printer operation. The client that creates these Subscription
157 objects becomes the owner of the Subscription object.

158 When creating each Subscription object, the client supplies the "notify-recipient" (uri) attribute. The
159 "notify-recipient" attribute specifies both a single Notification Recipient that is to receive the Notifications
160 when subsequent events occur and the method for Notification delivery that the IPP Printer is to use. For
161 the '~~ipp-notify-poll~~ipp' Notification delivery method defined in this document, there is no notify-recipient
162 because the Printer waits for one or more clients to ask for Notifications from a Subscription object rather
163 than sending them. Rather, any client that is authenticated (1) as an operator or administrator or (2) as the
164 owner of the Subscription object can initiate a Get-Notifications operation for that Subscription object.
165 Therefore, any Printer that supports the '~~ipp-notify-poll~~ipp' notification delivery method MUST ~~queue~~ hold
166 ~~each~~ event Notifications for ~~the lease time that it advertises to clients a sliding window of time for each~~
167 ~~Subscription object~~. Thus a single user can login at different places, say his/her office, the lab, and/or
168 several desktops in the same room, and receive the same event Notifications from a single Subscription
169 object.

170 The client issues a Get-Notifications Printer operation in order to initiate the delivery of the pending
171 Notifications held by the Printer for the Subscription objects requested. The client can indicate in the Get-
172 Notifications request whether it wants to receive all pending Notifications for

173 1) any existing Subscription objects for which it is the owner,

174 2) any existing Subscription objects associated with a job-id or

175 3) particular Subscription object(s) (for which it MUST be the owner or have read-access rights).

176 In either any case, the Notifications are returned in a response to the Get-Notifications request.

177 If the client requests a persistent channel ~~and if the Printer has returned minimum intervals that are~~
178 ~~sufficiently short~~, then the Printer **MAY** keep the channel open. Either the client or the IPP Printer can
179 disconnect the HTTP connection.

180 ~~ISSUE 01: Should it be possible for a client to ask for the Per Job Subscriptions for a particular job using a~~
181 ~~"job-id", instead of the subscription-id, which currently isn't returned by a Job Creation operation?~~

182 4 Get-Notifications operation

183 This REQUIRED operation allows the client to request that pending Notifications be delivered as a
184 response to this request. The client **MUST** be the owner or have write-access rights of the Subscription
185 objects that are involved and the delivery method specified when the Subscription objects were created
186 **MUST** be 'ipp-notify-pollipp'. When the Printer creates a Subscription Object, either with a Job Creation
187 operation or with a Create-Printer-Subscription or Create-Job-Subscription operation and a subscription
188 object contains the 'ipp-notify-pollipp' value for the "notify-recipient" operation attribute, the Printer returns
189 ~~the lease time for Events and the recommended time interval before the client to performs the next Get-~~
190 ~~Notifications operation, a minimum and maximum interval in the response.~~ The client **SHOULD** perform a
191 Get-Notifications operation ~~after at about~~ the ~~minimum-recommended~~ interval and if the Printer receives
192 the Get-Notifications before the ~~maximum-lease time interval~~ has elapsed, it **MUST** have all of the
193 Notifications ~~that has occurred since the~~ since the previous Get-Notification operation or the Subscription
194 object ~~was created~~ creation, whichever was most recent.

195 ~~ISSUE 02: Is there anything useful that we could define for the rest of the "notification recipient" (uri)~~
196 ~~attribute, since there is no recipient address needed after the 'ipp-notify-poll:/' since the recipient(s) poll?~~

197 The IPP Printer **MUST** accept the request in any state (see [ipp-mod] "printer-state" and "printer-state-
198 reasons" attributes) and **MUST** remain in the same state with the same "printer-state-reasons".

199 *Access Rights:* The authenticated user (see [ipp-mod] section 8.3) performing this operation must either be
200 the Subscription object owner (as determined when the Subscription object was created by the Job Creation
201 operation, Create-Job-Subscription, or Create-Printer-Subscription operations) or an operator or
202 administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5). Otherwise, the IPP object **MUST**
203 reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-
204 authorized' as appropriate.

205 ~~Issue 02.1: Is it possible for this operation to have an option that causes it to delay completing its response.~~
206 ~~It would initially returns all existing event notifications. Then it would return additional notifications as~~
207 ~~they occur for some period of time. The client would receive these notification events as they occur. The~~
208 ~~question is whether http servers or proxies would behave in this manner.~~

209 4.1 Get-Notifications Request

210 The following groups of attributes are part of the Get-Notifications Request:

211 Group 1: Operation Attributes

212 Natural Language and Character Set:

213 The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod]
214 section 3.1.4.1.

215

216 Target:

217 The "printer-uri" (uri) operation attribute which is the target for this operation as described in [ipp-
218 mod] section 3.1.5.

219

220 Requesting User Name:

221 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
222 described in [ipp-mod] section 8.3.

223

224 "notification-recipient" (url):

225 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It
226 is a url that identifies one or more Subscription objects for which event Notifications are being
227 requested. If the client supplies this attribute, but no notification-recipients are found, the IPP
228 Printer MUST return the 'client-error-not-found' status code. If some are found and others are not,
229 the ones that are not found are return in the Unsupported Attributes. By definition, if a notification-
230 recipient url exists, there must be at least one Subscription object.

231

232 If the client does not supply this attribute, the "jobs-ids" attribute and the "subscription-ids"
233 attribute, then the IPP Printer returns event Notifications for all Subscription objects for which the
234 client is the owner and the "notify-recipients" attribute is 'ipp'. It is not an error if there are currently
235 no Subscription objects for this client; the response then contains no Notifications.

236

237 If a client supplies this attribute and the "subscription-ids" attribute, the Printer returns event
238 Notifications for all Subscription objects specified by both attributes. If the "subscription-ids"
239 attributes contains values that are implied by the "job-id", the Printer MAY remove duplicates.

240

241 Note: this attribute allows a subscribing client to pick urls that are unique, e.g. the client's own url
242 or a friends url, which in both cases is likely the url of the person's host. If a client uses such a url
243 as the value of this attribute, the client gets events for all Subscription objects whose "notification-
244 recipient" is the specified url. This mechanism is more general than getting all subscriptions owned
245 by a client. It allows clients who didn't subscribe to get events without knowing job-ids or
246 subscription-ids.

247

248 **ISSUE 02.5: Is the above option useful?**

249 "subscription-ids" (1setOf integer(1:MAX)):

250 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It
251 is an integer value that identifies one or more Subscription objects for which event Notifications are
252 being requested. If the client supplies this attribute, but none of the Subscription objects are found,
253 the IPP Printer MUST return the 'client-error-not-found' status code. If some are found and others
254 are not, the ones that are not found are return in the Unsupported Attributes.

255

256 If the client does not supply this attribute the "job-ids" attribute and the "notification" attribute, then
257 the IPP Printer returns event Notifications for all Subscription objects for which the client is the
258 owner and the "notify-recipients" attribute is 'ipp-notify-pollipp'. It is not an error if there are
259 currently no Subscription objects for this client; the response then contains no Notifications.;

260

261 "job-ids" (1setOf integer(1:MAX)):

262 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It
263 is an integer value that identifies one or more job-ids. These job-ids identify the Subscription
264 objects for which event Notifications are being requested. If the client supplies this attribute, but no
265 Jobs are found, the IPP Printer MUST return the 'client-error-not-found' status code. If some are
266 found and others are not, the ones that are not found are return in the Unsupported Attributes. It is
267 not an error if there are no Subscription objects for a Job.

268

269 If the client does not supply this attribute, the "subscription-ids" attribute and the "notification-
270 recipients" attribute, then the IPP Printer returns event Notifications for all Subscription objects for
271 which the client is the owner and the "notify-recipients" attribute is 'ipp'. It is not an error if there
272 are currently no Subscription objects for this client; the response then contains no Notifications.

273

274 **ISSUE 02.6: Does the above paragraph describe a useful option that notification-recipient cannot do?**
275 **Should this case be an error instead?**

276

277 If a client supplies more than one of the last three attributes described for this operation, the Printer
278 returns event Notifications for all Subscription objects specified by all attributes. If these attribute
279 describe duplicate notification events, the Printer MAY remove them.

280

281

282

283 4.2 Get-Notifications Response

284 The Printer object returns either an immediate error response or a successful response with status code:
285 'successful-ok' when the first event occurs, i.e., when the Printer delivers the first event Notification.

286 Group 1: Operation Attributes

287 Status Message:

288 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY
289 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation
290 attribute as described in [ipp-mod] sections 13 and 3.1.6.

291

292 Natural Language and Character Set:

293 The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod]
294 section 3.1.4.2.

295

296 "~~minimumrecommended~~-time-interval" (integer(0:MAX)):

297 The value of this attribute is the ~~minimumrecommended~~ number of seconds that SHOULD elapse
298 before the client performs this operation again for these subscription-ids. A client MAY perform
299 this operation at any time, and a Printer MUST respond with all ~~pending-existing~~ Notifications. A
300 client observes this value in order to be a "good network citizen". The value that a Printer returns
301 for this attribute MUST NOT exceed 80% of the "lease-time-interval" in order to give a client plenty
302 of time to perform another Get-Notifications operation before the lease of the oldest events expire.

303

304 "~~maximumlease~~-time-interval" (integer(0:MAX)):

305 The value of this attribute is the minimum number of seconds that the Printer will retain all future
306 events. Thus this number is the maximum number of seconds that ~~SHOULD~~-elapses before this
307 client SHOULD issue this operation again for these subscription-ids. A Printer MUST preserve all
308 Notifications that occur for the number of seconds specified by this attribute starting at the time it is
309 sent in a response. A client MAY perform this operation at any time, and a Printer MUST respond
310 with all ~~pending-existing~~ Notifications. If a Printer receives this operation after this time interval, it
311 ~~SHOULD-MAY~~ have discarded some Notifications since the last response.

312

313 ~~ISSUE 04 – Or MUST the Printer discard events that occurred earlier than the sliding time window
314 specified by the difference between these two values? Otherwise, the clients may get back a lot of
315 duplicate events on subsequent requests.~~

316

317

318 Group 2: Unsupported Attributes

319 See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes.

320

321 If the "subscription-ids" attribute contained subscription-ids that do not exist, the Printer returns
322 them in this group as value of the "subscription-ids" attribute.

323

324 Group 3 through N: Notification Attributes

325 The Printer object responds with one event Notification per Group for each pending Notification
326 that meets the criteria specified by the subscription-ids attribute and requesting user name.(see [ipp-
327 ntfy]).

328 5 Extension to Print-Job, Print-URI, Create-Job, Create-Printer-Subscription 329 and Create-Printer-Subscription

330 5.1 Response

331 When Print-Job, Print-URI or Create-Job contains a "job-notify" attribute and the "notify-recipient" is '~~ipp-
332 notify-pollipp~~', the response contains two additional Operation Attributes that pertain to subscriptions.

333 When Create-Job-Subscription or Create-Printer-Subscription operation contains a “notify-recipient” that is
334 'ipp-notify-pollipp', the response contains two additional Operation Attributes that pertain to subscriptions.

335 Group 1: Operation Attributes

336 "minimumrecommended-time-interval" (integer(0:MAX)):

337 The value of this attribute is the minimumrecommended number of seconds that SHOULD-elapses
338 before the client SHOULD performs the Get-Notification operation for the first time with any
339 subscription-ids returned with this job. A client MAY perform the Get-Notification operation at any
340 time, and a Printer MUST respond with all pending-existing Notifications. A client observes this
341 value in order to be a “good network citizen”. The value that a Printer returns for this attribute
342 MUST NOT exceed 80% of the "lease-time-interval" in order to give a client plenty of time to
343 perform another Get-Notifications operation before the lease of the oldest events expire.

344
345 ISSUE 05: if we don't want to have Job Creation operations return subscription id's, then allow a
346 "job-ids" operation attribute in the Get-Notifications request in addition to the "subscription-ids"
347 operation attribute.

349 "maximumlease-time-interval" (integer(0:MAX)):

350 The value of this attribute is the minimum number of seconds that the Printer will retain all future
351 events. Thus this number is the maximum number of seconds that SHOULD-elapses before a Printer
352 client SHOULD receives-perform the Get-Notification operation for the first time with any
353 subscription-ids returned with this job. A Printer MUST preserve all Notifications that occur for the
354 number of seconds specified by this attribute starting at the time it is sent in a response. A client
355 MAY perform the Get-Notification operation at any time, and a Printer MUST respond with all
356 pending Notifications. If a Printer receives a Get-Notification operation after this time interval, it
357 may have discarded some Notifications since the last response.

358

359 6 Encoding

360 The operation-id assigned for the Get-Notification operation is:

361 0x00??

362 and should be added to the next version of [ipp-mod] section 4.4.15 "operations-supported".

363 This notification delivery method uses the IPP transport and encoding [ipp-pro] for the Get-Notifications
364 operation with one extension:

365 Instead of defining a new object attribute tag, a Generic Object attributes tag is defined that is used
366 for all new objects, such as Subscription objects, etc. Then this one new tag can also be used for the
367 Get-Notifications response Group 3 tag and subsequent groups in section 4.2:

368 notification-attributes-tag = %x07 ; tag of 7

369 7 IANA Considerations

370 ~~IANA will be asked to register this 'ipp-notify-poll' notification delivery scheme.~~There is nothing to
371 register.

372 8 Internationalization Considerations

373 With the '~~ipp-notify-poll~~ipp' method defined in this document, the client cannot request the Human
374 Consumable form by supplying the "notify-~~text~~-format" operation attribute (see [ipp-ntfy]). The only
375 supported value for this delivery method is "application/ipp". Therefore, the IPP Printer does not have to
376 perform any localization with this notification delivery method. However, the client when it receives the
377 Get-Notifications response is expected to localize the attributes that have the 'keyword' attribute syntax
378 according to the charset and natural language requested in the Get-Notifications request.

379 9 Security Considerations

380 The IPP Model and Semantics document [ipp-mod] discusses high level security requirements (Client
381 Authentication, Server Authentication and Operation Privacy). Client Authentication is the mechanism by
382 which the client proves its identity to the server in a secure manner. Server Authentication is the
383 mechanism by which the server proves its identity to the client in a secure manner. Operation Privacy is
384 defined as a mechanism for protecting operations from eavesdropping.

385 Unlike other event Notification delivery methods in which the IPP Printer initiates the event Notification,
386 with the method defined in this document, the Notification Recipient is the client who issues the Get-
387 Notifications operation. Therefore, there is no chance of "spam" notifications with this method.
388 Furthermore, such a client can close down the HTTP channel at any time, and so can avoid future unwanted
389 event Notifications at any time.

390 10 References

391 [ipp-mod]

392 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
393 Semantics", <draft-ietf-ipp-model-v11-04.txt>, June, 1999.

394 [ipp-ntfy]

395 Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing
396 Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-01.txt>, October 14,
397 1999.

398 [ipp-pro]

399 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and
400 Transport", draft-ietf-ipp-protocol-v11-03.txt, June, 1999.

401 [rfc2026]

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

403 [RFC2616]

404 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext
405 Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.

406 **11 Author's Addresses**

407 Robert Herriot
408 3400 Hillview Ave., Bldg #1
409 Palo Alto, CA 94304
410
411 Phone: 650-813-7696
412 Fax: 650-813-6860
413 Email: rherriot@pahv.xerox.com
414

415 **12 Full Copyright Statement**

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

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

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

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