

1 INTERNET-DRAFT — 1 ISSUES are highlighted like this.  
2 <draft-ietf-ipp-notify-poll-0001.txt>

Robert Herriot  
Xerox Corp.  
Tom Hastings  
Xerox Corp.  
Carl-Uno Manros  
Xerox Corp.  
Harry Lewis  
IBM, Corp.  
March 8 May 11, 2000

11 Internet Printing Protocol (IPP):  
12 The '**ippipp-get**' Notification Polling Method

13  
14 Copyright (C) The Internet Society (2000). All Rights Reserved.

15 Status of this Memo

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

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

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

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

24 **Abstract**

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

30 When a Printer supports the '**ippipp-get**' delivery method, it holds each Event Notification for a certain  
31 length of time. The amount of time is called the "~~event-lease~~ lease time". A Printer may assign the same  
32 event-lease time to each Event Notification or different times. If a Notification Recipient does not want to  
33 miss Event Notifications, the time between consecutive pollings of Subscription objects must be less than  
34 the event-~~lease~~ lease time of the events that occur between pollings. The Get-Notifications request  
35 indicates whether the client wants to receive all pending event Notifications for ~~(1) any Subscription for~~  
36 ~~which the client is the owner, (2) any Subscription associated with a Job, (3)~~ any Subscription with a  
37 particular delivery-method URL, ~~or (4) an identified set of Subscription objects~~. With the Get-Notifications  
38 operation, the Printer returns all existing Event Notifications along with two time intervals. One specifies  
39 the minimum time at which event-leases of future events of the type returned will begin to expire and the

40 other specifies the recommended interval for the client to wait before sending the next Get-Notifications  
41 operation. The second time interval is less than the first.

42 The Printer may keep the channel open if the recommended interval is sufficiently short, but in any case the  
43 client performs a new Get-Notifications operation each time it wants more Event Notifications. Since the  
44 time interval between consecutive client requests is normally less than the event-lease time, consecutive  
45 responses will normally contain some Event Notifications that are identical. The youngest ones in the  
46 previous response will become the oldest in the next response. The client is expected to filter out these  
47 duplicates, which is easy to do because of the sequence number in each Event Notification.

48 The full set of IPP documents includes:

49 Design Goals for an Internet Printing Protocol [RFC2567]

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

51 Internet Printing Protocol/1.1: Model and Semantics [ipp-mod]

52 Internet Printing Protocol/1.1: Encoding and Transport [ipp-pro]

53 Internet Printing Protocol/1.1: Implementer's Guide [ipp-iig]

54 Mapping between LPD and IPP Protocols [RFC2569]

55 Internet Printing Protocol/1.0 & 1.1: Event Notification Specification [ipp-ntfy]

56

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

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

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

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

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

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

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

86

87

**Table of Contents**

88 1 Introduction ..... 5

89 2 Terminology ..... 5

90 3 Model and Operation ..... 6

91 4 Get-Notifications operation ..... 7

92 4.1 GET-NOTIFICATIONS REQUEST ..... 8

93 4.2 GET-NOTIFICATIONS RESPONSE ..... 10

94 5 Extensions to Print-Job, Print-URI, Create-Job, Create-Printer-Subscription and Create-Printer-Subscription ..... 13

95 5.1 RESPONSE ..... 13

96 5.1 RESPONSE ..... 13

97 6 Encoding ..... 14

98 7 IANA Considerations ..... 14

99 8 Internationalization Considerations ..... 14

100 9 Security Considerations ..... 15

101 10 References ..... 15

102 11 Authors' Addresses ..... 15

103 12 Full Copyright Statement ..... 16

104

105

## 106 1 Introduction

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

115 When a Printer supports the 'ippipp-get' delivery method, it holds each Event Notification for a certain  
116 length of time. The amount of time is called the "event-lease time". A Printer may assign the same event-  
117 lease time to each event or different times. If a Notification Recipient does not want to miss Event  
118 Notifications, the time between consecutive pollings of Subscription objects must be less than the event-  
119 lease time of the Event Notifications that occur between pollings. The Get-Notifications request indicates  
120 whether the client wants to receive all pending Event Notifications for ~~(1) any Subscription for which the~~  
121 ~~client is the owner, (2) any Subscription associated with a particular Job, (3) any Subscription with a~~  
122 ~~particular notification recipient URL, or (4) an identified set of Subscription objects.~~ With the Get-  
123 Notifications operation, the Printer returns all existing Event Notifications along with two time intervals.  
124 One specifies the minimum time at which event-leases of future events of the type returned will begin to  
125 expire and the other specifies the recommended interval for the client to wait before sending the next Get-  
126 Notifications operation. The second time interval is less than the first.

127 The Printer may keep the channel open if the recommended interval is sufficiently short, but in any case the  
128 client performs a new Get-Notifications operation each time it wants more Notifications. Since the time  
129 interval between consecutive client requests is normally less than the event-lease time, consecutive  
130 responses will normally contain some events that are identical. The youngest ones in the previous response  
131 will become the oldest in the next response. The client is expected to filter out these duplicates, which is  
132 easy to do because of the sequence number in each Notification. The reason for not removing the  
133 Notifications from the Subscription object with every Get-Notifications request, is so that multiple  
134 Notification Recipients can be polling the same subscription object and so the Get-Notification operation  
135 satisfies the rule of idempotency. The former is useful if someone is logged in to several desktops at the  
136 same time and wants to see the same events at both places. The latter is useful if the network loses the  
137 response.

## 138 2 Terminology

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

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

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

144 Notification Recipient - See [ipp-ntfy]

145 Subscription object - See [ipp-ntfy]

146 Ultimate Notification Recipient - See [ipp-ntfy]

### 147 3 Model and Operation

148 In the IPP Notification Model [ipp-ntfy], at most one ~~or more~~ Per-Job Subscriptions can be supplied in the  
149 Job Creation operation. ~~or~~ In addition one Per-Job Subscription can be supplied in ~~OPTIONALLY as~~  
150 ~~subsequent~~ Create-Job-Subscription operations, and; one Per-Printer Subscription can be supplied in the  
151 Create-Printer operation. The client that creates these Subscription objects becomes the owner of the  
152 Subscription object.

153 When creating each Subscription object, the client supplies the "~~notify-recipient~~notify-recipient-uri" (uri)  
154 attribute. The "~~notify-recipient~~notify-recipient-uri" attribute specifies both a single Notification Recipient  
155 that is to receive the Notifications when subsequent events occur and the URL's scheme specifies the  
156 method for Notification delivery that the IPP Printer is to use. For the -Notification delivery method  
157 defined in this document, the scheme of the URL is 'ippipp-get' and the host SHOULD be the client host's  
158 URL. In addition, the URL MAY contains a path to allow for applications to have a unique URL. Because  
159 the Get-Notifications operation uses the "notification-recipient-uri" to specify the events that it wants in the  
160 response, the Subscriber can partition events into suitable groups by associating a different URL with each  
161 group – the URLs may have the same host but different paths. If a Subscriber wants a friend to receive  
162 Event Notification via this delivery method, it can use the friend's URL as the "notification-recipient-uri".  
163 When the friend performs the Get-Notifications operation on the URL, it receives all pending the  
164 notifications, even those event caused by subscriptions owned by others.

165 ISSUE 1: The 'ipp' is a convenient reuse of 'ipp', but does it imply the existence of a print service at each  
166 client that is not a reality?

167 For most Notification delivery methods, a Printer sends Event Notifications to the delivery URL and the  
168 Printer does not perform any authentication or authorization with the receivers of the Event Notifications.  
169 For the Notification delivery method defined in this document, the client requests Event Notifications from  
170 the Printer via a Get-Notifications operation, and the Printer performs the same authentication and  
171 authorization as it does for the Get-Job-Attributes operation. That is, a -Printer MAY allow a client to  
172 perform a Get-Notifications operation on any Subscription object or it MAY restrict access as follows. Any  
173 client that is authenticated (1) as an operator or administrator or (2) as the owner of the Subscription object  
174 can initiate a Get-Notifications operation for that Subscription object.

175 Because a Printer has to wait for a client to request Event Notifications for the 'ippipp-get' delivery method,  
176 any Printer that supports the 'ippipp-get' notification delivery method MUST hold each Event Notification  
177 at least for the event-lease time that it advertises to clients. With this rule, a single user can login at  
178 different places, say his/her office, the lab, and/or several desktops in the same room, and receive the same  
179 Event Notifications from a single Subscription object. In addition, a client that gets no response, perhaps  
180 because of a network failure, can perform the Get-Notifications operations two or more times in quick

181 succession and get the same results except for a few newly arrived Event Notifications and a few old Event  
182 Notifications whose event-leases have expired.

183 The event-lease time assigned to Event Notifications MAY be different for each implementation.  
184 Furthermore, a particular implementation MAY assign different event-lease times to each Event  
185 Notification. If a Printer assigns different event-lease times to each Event Notification, the event-lease time  
186 returned with Get-Notifications MUST be a value that ensures a client will not miss future Event  
187 Notifications.

188 The client issues a Get-Notifications Printer operation in order to initiate the delivery of the pending  
189 Notifications held by the Printer for the Subscription objects requested. In this operation, the client  
190 specifies the "notification-recipient-uri" attribute and the Printer returns all pending Event Notifications  
191 associated with Subscription objects whose "notification-recipient-uri" attribute matches the "notification-  
192 recipient-uri" attribute specified in the operation.

193 ~~can indicate in the Get-Notifications request whether it wants to receive all pending Notifications for:~~

194 ~~1) any existing Subscription objects for which the client is the owner,~~

195 ~~2) any existing Subscription objects whose notification-recipient is a specified URL~~

196 ~~3) any existing Subscription objects associated with a job-id or~~

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

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

199 If the client requests a persistent channel, then the Printer MAY keep the channel open. Either the client or  
200 the IPP Printer can disconnect the HTTP connection.

#### 201 4 Get-Notifications operation

202 This REQUIRED operation allows the client to request that pending Event Notifications be delivered as a  
203 response to this request. The client MUST be the owner or have read-access rights of the Subscription  
204 objects that are involved and the delivery method specified when the Subscription objects were created  
205 MUST be 'ippipp-get'.

206 This operation returns all pending Event Notifications specified by the "notify-recipient-uri" operation  
207 attribute. To help a client know when to perform this operation again, the Printer returns both the event-  
208 lease time and the suggested-ask-again time in the following operations:

209 a) Job Creation, Create-Printer-Subscription and Create-Job-Subscription operation if the scheme  
210 of the "notify-recipient-uri" operation attribute is 'ipp-get'.

211 b) All Get-Notifications operation

212 ~~When the Printer creates a Subscription Object, either with a Job Creation operation or with a Create-~~  
213 ~~Printer Subscription or Create Job Subscription operation and a subscription object contains the 'ipp' value~~  
214 ~~for the "notify-recipient" operation attribute, the Printer returns the event-lease time for Events and the~~  
215 ~~recommended time interval before the client to performs the next Get Notifications operation.~~ The client  
216 SHOULD perform a Get-Notifications operation at about the ~~suggested-ask-again time~~ recommended  
217 ~~interval~~ and if the Printer receives the Get-Notifications before the event-lease time has elapsed, it MUST  
218 have all of the Notifications since the previous Get-Notification operation or the Subscription object  
219 creation, whichever was most recent.

220 ~~Issue 2: Now that the Get Notification operation does not affect the Event Notifications in the Printer, it~~  
221 ~~should not require write access to access them.~~

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

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

230 ~~Issue 3: Is it possible for this operation to have an option that causes it to delay completing its response? It~~  
231 ~~would initially returns all existing Event Notifications. Then it would return additional notifications as they~~  
232 ~~occur for some period of time. The client would receive these Event Notifications as they occur. The~~  
233 ~~question is whether http servers or proxies would behave in this manner so that the client would get the~~  
234 ~~Event Notifications without delay after they are sent by the http server? It has been suggested that the~~  
235 ~~Printer send each burst of Event Notifications be in a separate message body where each message body is~~  
236 ~~part of a multipart MIME content type.~~

#### 237 4.1 Get-Notifications Request

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

##### 239 Group 1: Operation Attributes

###### 240 Natural Language and Character Set:

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

243

###### 244 Target:

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

247



248 Requesting User Name:  
249 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
250 described in [ipp-mod] section 8.3.  
251

252 "notification-recipient-uri" (url):

253 The client MUST supply this attribute. The Printer object MUST support this attribute. The Printer  
254 matches the value of this attribute byte for byte against the value of the "notification-recipient-uri"  
255 in each Subscription object in the Printer. If there are no matches, the IPP Printer MUST return the  
256 'client-error-not-found' status code. If there are matches, the IPP Printer MUST return all  
257 accumulated Event Notifications associated with Subscription objects that contain the matched  
258 "notification-recipient-uri" attribute.

259 ~~The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It~~  
260 ~~is a URL that identifies one or more Subscription objects for which Event Notifications are being~~  
261 ~~requested. If the client supplies this attribute, but no notification recipients are found, the IPP~~  
262 ~~Printer MUST return the 'client-error-not-found' status code. If some are found and others are not,~~  
263 ~~the ones that are not found are return in the Unsupported Attributes. By definition, if a notification-~~  
264 ~~recipient URL exists, there must be at least one Subscription object.~~

265  
266  
267  
268 Note: this attribute allows a subscribing client to pick URLs that are unique, e.g. the client's own  
269 URL or a friend's URL, which in both cases is likely the URL of the person's host. An application  
270 could make a URL unique for each application if it wants. ~~If a client uses such a URL as the value~~  
271 ~~of this attribute, the client gets event Notifications for all Subscription objects whose "notification-~~  
272 ~~recipient" is the specified URL. This mechanism is more general than getting all subscriptions~~  
273 ~~owned by a client. It allows clients who didn't subscribe to get Event Notifications without knowing~~  
274 ~~job-ids or subscription-ids.~~  
275

276 **ISSUE 4:** ~~The "notification-recipient" option allows a client to group multiple Subscription ids with a~~  
277 ~~single URL. A client might decide to use the same URL for all subscriptions for a user, or it might have a~~  
278 ~~separate URL for each client program. In addition a client might use an URL belonging to some other~~  
279 ~~known user and let that user access Event Notifications using that URL. Is the above option useful?~~

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

281 ~~The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It~~  
282 ~~is an integer value that identifies one or more Subscription objects for which Event Notifications are~~  
283 ~~being requested. If the client supplies this attribute, but none of the Subscription objects are found,~~  
284 ~~the IPP Printer MUST return the 'client-error-not-found' status code. If some are found and others~~  
285 ~~are not, the ones that are not found are return in the Unsupported Attributes.~~  
286  
287

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

289 ~~The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It~~  
290 ~~is an integer value that identifies one or more job-ids. These job-ids identify the Subscription~~

291 ~~objects for which Event Notifications are being requested. If the client supplies this attribute, but no~~  
292 ~~Jobs are found, the IPP Printer MUST return the 'client-error-not-found' status code. If some are~~  
293 ~~found and others are not, the ones that are not found are returned in the Unsupported Attributes. It~~  
294 ~~is not an error if there are no Subscription objects for a Job.~~

295  
296 ~~If the client supplies none of the last three attributes described for this operation, then the IPP~~  
297 ~~Printer returns Event Notifications for all Subscription objects for which the client is the owner and~~  
298 ~~the "notify-recipients" attribute is 'ipp'. It is not an error if there are currently no Subscription~~  
299 ~~objects for this client; the response then contains no Notifications.~~

300

301 ~~ISSUE 5: Does the mechanism described in the above paragraph describe a useful option that "notification-~~  
302 ~~recipient" alone cannot do? Should this case be an error instead?~~

303

304 ~~If a client supplies more than one of the last three attributes described for this operation, the Printer~~  
305 ~~returns Event Notifications for all Subscription objects specified by all attributes. If these attributes~~  
306 ~~describe duplicate Event Notifications, the Printer MAY remove them.~~

307

308 ~~ISSUE 6: Is it better if "notification-recipient" is the only way to request Event Notification? If so, this~~  
309 ~~behaves more like other notification delivery methods where a recipient receives those and only those~~  
310 ~~events with its delivery URL. Furthermore, if there is a single mechanism of "notification-recipient" for a~~  
311 ~~client to specify Event Notifications, a Printer can better optimize event leases because it knows which~~  
312 ~~events will be accessed together. If client can specify subscription-ids, each request can contain a different~~  
313 ~~mix of subscription-ids.~~

314

## 315 4.2 Get-Notifications Response

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

### 318 Group 1: Operation Attributes

#### 319 Status Message:

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

323

#### 324 Natural Language and Character Set:

325 The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod]  
326 section 3.1.4.2. The Printer uses the values of "notify-charset" and "notify-natural-language",  
327 respectively, from one of the Subscription objects associated with the Event Notifications in this  
328 response.

329

330 Normally, the values of these attributes is the same in all Subscriptions. If they are not, the Printer  
331 picks the values from one Subscription object to put in these attributes. The algorithm for picking  
332 the Subscription object is implementation dependent. The choice of natural language is not critical  
333 because 'text' and 'name' values can override the operation level natural-language. The Printer's  
334 choice of charset is critical because a bad choice may leave it unable to send some 'text' and 'name'  
335 values accurately.

336  
337 "~~suggested-ask-again~~~~recommended~~-time-interval" (integer(0:MAX)):

338 The value of this attribute is the ~~recommended~~~~suggested~~ number of seconds that SHOULD elapse  
339 before the client performs this operation again for these Subscription objects. A client MAY  
340 perform this operation at any time, and a Printer MUST respond with all existing Notifications. A  
341 client observes this value in order to be a "good network citizen". The value that a Printer returns for  
342 this attribute MUST NOT exceed 80% of the "event-lease-time-interval" in order to give a client  
343 plenty of time to perform another Get-Notifications operation before the event-lease of the oldest  
344 Event Notifications expire.

345  
346 "event-lease-time-interval" (integer(0:MAX)):

347 The value of this attribute is the minimum number of seconds until the event-lease expiration time  
348 for all future Event Notifications associated with the Subscription objects generating the requested  
349 Event Notifications. Thus this number is the maximum number of seconds that elapses before this  
350 client SHOULD issue this operation again for these Subscription objects. A Printer MUST preserve  
351 all Notifications that occur for the number of seconds specified by this attribute starting at the time  
352 it is sent in a response. A client MAY perform this operation at any time, and a Printer MUST  
353 respond with all existing Event Notifications. If a Printer receives this operation after this time  
354 interval, it MAY have discarded some Notifications since the last response.

355  
356 "printer-up-time" (integer(0:MAX)):

357 The value of this attribute is the Printer's "printer-up-time" attribute. Because each Event  
358 Notification also contains the value of this attribute when the event occurred, the value of this  
359 attribute lets a client know when each Event Notification occurred relative to the time of this  
360 response.

## 363 Group 2: Unsupported Attributes

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

365  
366 If the "subscription-ids" attribute contained subscription-ids that do not exist, the Printer returns  
367 them in this group as value of the "subscription-ids" attribute.

## 368 369 Group 3 through N: Event Notification Attributes

370 The Printer object responds with one Event Notification per Group for each supplied Event  
371 Notification. Each Event Notification MUST ~~that-meets~~ the criteria specified by the request. (see  
372 [ipp-ntfy]). Each Event Notification Group MUST start with an 'event-notification-attributes-tag'.

373 which is the tag that begins an Event Notification Group (see the section "Encodings of Additional  
 374 Attribute Tags" in [ipp-ntfy]).

375  
 376 This group includes the following attributes from the section on " Content of Machine Consumable  
 377 Event Notifications" in [ipp-ntfy]. They are encoded using the IPP rules for encoding attributes [ipp-  
 378 pro] and they may be encoded in any order. Note: the Get-Jobs response acts as a model for  
 379 encoding multiple groups of attributes.

380  
 381 Table 1 and Table 3 contains the following information

- 382 a) Attribute: the name of the attribute to include from the section on " Content of Machine  
 383 Consumable Event Notifications" in [ipp-ntfy].  
 384 b) Condition: the condition for the attribute to be present. The value  
 385 i) " means that the attribute MUST be present in all Event Notifications.  
 386 ii) 'conditional' means the attribute MUST be present if the Printer supports the attribute  
 387 iii) 'progress' means the attribute MUST be present for 'job-progress' or 'job-completed'  
 388 events only.

389  
 390 For a Event Notification for job and printer events, the Printer includes the following attributes.

391 **Table 1 – REQUIRED Attributes in all IPP Event Notification Content**

<u>Attribute</u>	<u>Condition</u>
<u>subscription-request-id (integer (0:MAX))</u>	
<u>notify-text (text)</u>	
<u>notify-text-format (mimeMediaType)</u>	
<u>printer-uri (uri)</u>	
<u>trigger-event (type2 keyword)</u>	
<u>printer-up-time (integer(MIN:MAX))</u>	
<u>printer-current-time (dateTime)</u>	<u>conditional</u>

392  
 393 For Event Notification for job events, the Printer includes the following additional attributes.

394 **Table 2 – REQUIRED Attributes in all IPP Event Notification Content**

<u>Attribute</u>	<u>Condition</u>
<u>job-id (integer(1:MAX))</u>	

<u>Attribute</u>	<u>Condition</u>
<u>job-state (type1 enum)</u>	
<u>job-state-reasons (1setOf type2 keyword)</u>	
<u>job-impressions-completed (integer(0:MAX))</u>	<u>progress</u>

395

396

For Event Notification for printer events, the Printer includes the following additional attributes.

397

**-Table 3 – REQUIRED Attributes in all IPP Event Notification Content**

<u>Attribute</u>	<u>Condition</u>
<u>printer-state (type1 enum)</u>	
<u>printer-state-reasons (1setOf type2 keyword)</u>	
<u>printer-is-accepting-jobs (boolean)</u>	

398

## 399 5 Extensions to Print-Job, Print-URI, Create-Job, Create-Printer- 400 Subscription and Create-Printer-Subscription

### 401 5.1 Response

402 When Print-Job, Print-URI or Create-Job contains a "notify-recipient-uri" attribute and the scheme in its  
403 value is 'ipp-get', the response contains two additional Operation Attributes that pertain to subscriptions.

404 When Create-Job-Subscription or Create-Printer-Subscription operation contains a "notify-recipient-uri"  
405 value whose scheme is 'ipp-get', the response contains two additional Operation Attributes that pertain to  
406 subscriptions.

407 ~~When Print Job, Print URI or Create Job contains a "job\_notify" attribute and the "notify recipient" is 'ipp',~~  
408 ~~the response contains two additional Operation Attributes that pertain to subscriptions.~~

409 ~~When Create Job Subscription or Create Printer Subscription operation contains a "notify recipient" that is~~  
410 ~~'ipp', the response contains two additional Operation Attributes that pertain to subscriptions.~~

#### 411 Group 1: Operation Attributes

412 " ~~suggested-ask-again~~recommended-time-interval" (integer(0:MAX)):

413 The value of this attribute is the ~~suggested~~ recommended-number of seconds that SHOULD elapse  
414 before the client SHOULD perform the Get-Notification operation for the first time with any

415 Subscription objects returned with this job. A client MAY perform the Get-Notification operation at

416 any time, and a Printer MUST respond with all existing Notifications. A client observes this value  
417 in order to be a "good network citizen". The value that a Printer returns for this attribute MUST  
418 NOT exceed 80% of the "event-lease-time-interval" in order to give a client plenty of time to  
419 perform another Get-Notifications operation before the event-lease of the oldest events expire.

420  
421

422 "event-lease-time-interval" (integer(0:MAX)):

423 The value of this attribute is the minimum number of seconds until the event-lease expiration time  
424 for all future Event Notifications associated with the Subscription objects generating the requested  
425 Event Notifications. Thus this number is the maximum number of seconds that elapses before a  
426 client SHOULD perform the Get-Notification operation for the first time with any Subscription  
427 objects returned with this job. A Printer MUST preserve all Notifications that occur for the number  
428 of seconds specified by this attribute starting at the time it is sent in a response. A client MAY  
429 perform the Get-Notification operation at any time, and a Printer MUST respond with all existing  
430 Event Notifications. If a Printer receives a Get-Notification operation after this time interval, it may  
431 have discarded some Notifications since the last response.

432

## 433 6 Encoding

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

435 0x00??

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

437 **ISSUE: what is the value?**

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

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

## 441 7 IANA Considerations

442 There is nothing to register.

## 443 8 Internationalization Considerations

444 With the 'ippipp-get' method defined in this document, the client cannot request the Human Consumable  
445 form by supplying the "notify-format" operation attribute (see [ipp-ntfy]). The only supported value for this  
446 delivery method is "application/ipp". Therefore, the IPP Printer does not have to perform any localization  
447 with this notification delivery method. However, the client when it receives the Get-Notifications response

448 is expected to localize the attributes that have the 'keyword' attribute syntax according to the charset and  
449 natural language requested in the Get-Notifications request.

## 450 9 Security Considerations

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

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

## 461 10 References

462 [ipp-mod]

463 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.1: Model and  
464 Semantics", <draft-ietf-ipp-model-v11-06.txt>, March 1, 2000.

465 [ipp-ntfy]

466 Isaacson, S., Martin, J., deBry, R., Hastings, R. Herriot, T., Shepherd, M., Bergman, R., "Internet  
467 Printing Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-03.txt>, May  
468 11, 2000.

469 [ipp-pro]

470 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and  
471 Transport", draft-ietf-ipp-protocol-v11-05.txt, March 1, 2000.

472 [rfc2026]

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

474 [RFC2616]

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

## 477 11 Authors' Addresses

478

479 Robert Herriot  
480 Xerox Corp.  
481 3400 Hill View Ave, Building 1

482 Palo Alto, CA 94304  
483  
484 Phone: 650-813-7696  
485 Fax: 650-813-6860  
486 e-mail: [robert.herriot@pahv.xerox.com](mailto:robert.herriot@pahv.xerox.com)

487  
488 Tom Hastings  
489 Xerox Corporation  
490 737 Hawaii St. ESAE 231  
491 El Segundo, CA 90245  
492  
493 Phone: 310-333-6413  
494 Fax: 310-333-5514  
495 e-mail: [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)

496  
497 Carl-Uno Manros  
498 Xerox Corporation  
499 701 Aviation Blvd.  
500 El Segundo, CA 90245  
501  
502 Phone: 310-333-  
503 Fax: 310-333-5514  
504 e-mail: [manros@cp10.es.xerox.com](mailto:manros@cp10.es.xerox.com)

505  
506 Harry Lewis  
507 IBM  
508 P.O. Box 1900  
509 Boulder, CO 80301-9191  
510  
511 Phone: (303) 924-5337  
512 FAX:  
513 e-mail: [harryl@us.ibm.com](mailto:harryl@us.ibm.com)  
514

## 515 **12 Full Copyright Statement**

516 Copyright (C) The Internet Society (2000). All Rights Reserved.

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



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

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

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