

1 INTERNET-DRAFT **6 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.  
March 86, 2000

3  
4  
5  
6  
7  
8  
9 Internet Printing Protocol (IPP):  
10 **The 'ipp' Notification Polling Method**

11  
12 Copyright (C) The Internet Society (~~1999~~2000). All Rights Reserved.

13 Status of this Memo

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

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

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

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

22 **Abstract**

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

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

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

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

47 The full set of IPP documents includes:

48 Design Goals for an Internet Printing Protocol [RFC2567]

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

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

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

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

53 Mapping between LPD and IPP Protocols [RFC2569]

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

55

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

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

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

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

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

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

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

85

86

**Table of Contents**

87 1 Introduction ..... 5

88 2 Terminology ..... 5

89 3 Model and Operation ..... 6

90 4 Get-Notifications operation ..... 7

91 4.1 GET-NOTIFICATIONS REQUEST ..... 8

92 4.2 GET-NOTIFICATIONS RESPONSE ..... 10

93 5 Extension to Print-Job, Print-URI, Create-Job, Create-Printer-Subscription and Create-Printer-Subscription ..... 11

94 5.1 RESPONSE ..... 11

95

96 6 Encoding ..... 12

97 7 IANA Considerations ..... 12

98 8 Internationalization Considerations ..... 12

99 9 Security Considerations ..... 12

100 10 References ..... 13

101 11 Authors' Addresses ..... 13

102 12 Full Copyright Statement ..... 14

103

104

## 105 1 Introduction

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

114 When a Printer supports the 'ipp' delivery method, it holds each Event Notification for a certain length of  
115 time~~.~~ The amount of time is called the "event-lease time" ~~and it is the same for all events in a Printer.~~ A  
116 Printer may assign the same event-lease time to each event or different times. If a Notification Recipient  
117 does not want to miss Event Notifications, the time between consecutive pollings of Subscription objects  
118 must be less than the event-lease time of the Event Notifications that occur between pollings. The Get-  
119 Notifications request indicates whether the client wants to receive all pending ~~events~~ Event Notifications  
120 for (1) any Subscription for which the client is the owner, (2) any Subscription associated with a particular  
121 Job, (3) any Subscription with a particular notification recipient urlURL, or (34) an particular identified set  
122 of Subscription objects. With the Get-Notifications operation, the Printer returns all existing Event  
123 Notifications along with two time intervals. One specifies the minimum time at which event-leases of  
124 future events of the type returned will begin to expire ~~length of the event-lease for all future events~~ and the  
125 other specifies the recommended interval for the client to wait before sending ~~to~~ the next Get-Notifications  
126 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 ~~later~~ ones in the previous  
131 response will become the oldest ~~earliest~~ in the next response. The client is expected to filter out these  
132 duplicates, which is easy to do because of the sequence number in each Notification. The reason for not  
133 removing the Notifications from the Subscription object with every Get-Notifications request, is so that  
134 multiple Notification Recipients can be polling the same subscription object and so the Get-Notification  
135 operation satisfies the rule of idempotency. The former is useful if someone is logged in to several  
136 desktops at the same time and wants to see the same events at both places. The latter is useful if the  
137 network loses the 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], one or more Per-Job Subscriptions can be supplied in the Job  
149 Creation operation or OPTIONALLY as subsequent Create-Job-Subscription operations; one Per-Printer  
150 Subscription can be supplied in the Create-Printer operation. The client that creates these Subscription  
151 objects becomes the owner of the Subscription object.

152 When creating each Subscription object, the client supplies the "notify-recipient" (uri) attribute. The  
153 "notify-recipient" attribute specifies both a single Notification Recipient that is to receive the Notifications  
154 when subsequent events occur and the method for Notification delivery that the IPP Printer is to use. For  
155 the 'ipp' Notification delivery method defined in this document, ~~the scheme of the of the uriURL is 'ipp' and~~  
156 ~~the host SHOULD be the client host's uriURL. In addition, the uriURL MAY contains a path to allow for~~  
157 ~~applications be to have a unique uriURL. there is no notify-recipient because the Printer waits for one or~~  
158 ~~more clients to ask for Notifications from a Subscription object rather than sending them.~~

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

161 ~~For most Notification delivery methods, a Printer sends Event Notifications to the delivery uriURL and the~~  
162 ~~Printer does not perform any authentication or authorization with the receivers of the Event Notifications.~~  
163 ~~For the Notification delivery method defined in this document, the client requests Event Notifications from~~  
164 ~~the Printer via a Get-Notifications operation, and the Printer performs the same authentication and~~  
165 ~~authorization as it does for the Get-Job-Attributes operation. Rather~~That is, a Printer MAY allow a client  
166 ~~to perform a Get-Notifications operation on any Subscription object or it MAY restrict access as follows.~~  
167 ~~A~~any client that is authenticated (1) as an operator or administrator or (2) as the owner of the Subscription  
168 object can initiate a Get-Notifications operation for that Subscription object.

169 ~~Therefore~~Because a Printer has to wait for a client to request Event Notifications for the 'ipp' delivery  
170 ~~method,~~ any Printer that supports the 'ipp' notification delivery method MUST hold each Event Notification  
171 ~~at least~~ for the event-lease time that it advertises to clients. ~~Thus~~With this rule, a single user can login at  
172 different places, say his/her office, the lab, and/or several desktops in the same room, and receive the same  
173 Event Notifications from a single Subscription object. ~~In addition, a client that gets no response, perhaps~~  
174 ~~because of a network failure, can perform the Get-Notifications operations two or more times in quick~~  
175 ~~succession and get the same results except for a few newly arrived Event Notifications and a few old Event~~  
176 ~~Notifications whose event-leases have expired.~~

177 ~~The event-lease time assigned to Event Notifications MAY be different for each implementation.~~  
178 ~~Furthermore, a particular implementation MAY assign different event-lease times to each Event~~  
179 ~~Notification. If a Printer assigns different event-lease times to each Event Notification, the event-lease time~~

180 returned with Get-Notifications MUST be a value that ensures a client will not miss future Event  
181 Notifications.

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

- 185 1) any existing Subscription objects for which the client is the owner,
- 186 2) any existing Subscription objects whose notification-recipient is a specified URL
- 187 3) any existing Subscription objects associated with a job-id or
- 188 4) particular Subscription object(s) (for which it MUST be the owner or have read-access rights).

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

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

## 192 4 Get-Notifications operation

193 This REQUIRED operation allows the client to request that pending Event Notifications be delivered as a  
194 response to this request. The client MUST be the owner or have writeread-access rights of the Subscription  
195 objects that are involved and the delivery method specified when the Subscription objects were created  
196 MUST be 'ipp'. When the Printer creates a Subscription Object, either with a Job Creation operation or with  
197 a Create-Printer-Subscription or Create-Job-Subscription operation and a subscription object contains the  
198 'ipp' value for the "notify-recipient" operation attribute, the Printer returns the event-lease time for Events  
199 and the recommended time interval before the client to performs the next Get-Notifications operation. The  
200 client SHOULD perform a Get-Notifications operation at about the recommended interval and if the Printer  
201 receives the Get-Notifications before the event-lease time has elapsed, it MUST have all of the  
202 Notifications since the previous Get-Notification operation or the Subscription object creation, whichever  
203 was most recent.

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

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

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

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

#### 221 4.1 Get-Notifications Request

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

##### 223 Group 1: Operation Attributes

###### 224 Natural Language and Character Set:

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

227

###### 228 Target:

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

231

###### 232 Requesting User Name:

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

235

###### 236 "notification-recipient" (url):

237 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
238 is a [url](#) that identifies one or more Subscription objects for which Event Notifications are being  
239 requested. If the client supplies this attribute, but no notification-recipients are found, the IPP  
240 Printer MUST return the 'client-error-not-found' status code. If some are found and others are not,  
241 the ones that are not found are return in the Unsupported Attributes. By definition, if a notification-  
242 recipient [url](#) exists, there must be at least one Subscription object.

243

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

248

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

252

253 Note: this attribute allows a subscribing client to pick [urls](#) that are unique, e.g. the client's  
254 own [url](#) or a friend's [url](#), which in both cases is likely the [url](#) of the person's host. [An](#)



255 application could make a uriURL unique for each application if it wants. If a client uses such a  
256 uriURL as the value of this attribute, the client gets events Notifications for all Subscription objects  
257 whose "notification-recipient" is the specified uriURL. This mechanism is more general than  
258 getting all subscriptions owned by a client. It allows clients who didn't subscribe to get Event  
259 Notifications without knowing job-ids or subscription-ids.  
260

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

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

266 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
267 is an integer value that identifies one or more Subscription objects for which Event Notifications are  
268 being requested. If the client supplies this attribute, but none of the Subscription objects are found,  
269 the IPP Printer MUST return the 'client-error-not-found' status code. If some are found and others  
270 are not, the ones that are not found are return in the Unsupported Attributes.  
271

272 ~~If the client does not supply this attribute the "job-ids" attribute and the "notification" attribute, then~~  
273 ~~the IPP Printer returns event Notifications for all Subscription objects for which the client is the~~  
274 ~~owner and the "notify-recipients" attribute is 'ipp'. It is not an error if there are currently no~~  
275 ~~Subscription objects for this client; the response then contains no Notifications.~~  
276

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

278 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It  
279 is an integer value that identifies one or more job-ids. These job-ids identify the Subscription  
280 objects for which Event Notifications are being requested. If the client supplies this attribute, but no  
281 Jobs are found, the IPP Printer MUST return the 'client-error-not-found' status code. If some are  
282 found and others are not, the ones that are not found are returned in the Unsupported Attributes. It  
283 is not an error if there are no Subscription objects for a Job.  
284

285 If the client ~~supplies none of the last three attributes described for this operation~~ ~~does not supply this~~  
286 ~~attribute, the "subscription-ids" attribute and the "notification-recipients" attribute,~~ then the IPP  
287 Printer returns Event Notifications for all Subscription objects for which the client is the owner and  
288 the "notify-recipients" attribute is 'ipp'. It is not an error if there are currently no Subscription  
289 objects for this client; the response then contains no Notifications.  
290

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

293  
294 If a client supplies more than one of the last three attributes described for this operation, the Printer  
295 returns Event Notifications for all Subscription objects specified by all attributes. If these attributes  
296 describe duplicate Event Notifications, the Printer MAY remove them.

297

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

304

## 305 4.2 Get-Notifications Response

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

### 308 Group 1: Operation Attributes

#### 309 Status Message:

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

313

#### 314 Natural Language and Character Set:

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

317

#### 318 "recommended-time-interval" (integer(0:MAX)):

319 The value of this attribute is the recommended number of seconds that SHOULD ~~elapses~~ elapse  
320 before the client performs this operation again for these ~~subscription-ids~~ Subscription objects. A  
321 client MAY perform this operation at any time, and a Printer MUST respond with all existing  
322 Notifications. A client observes this value in order to be a "good network citizen". The value that a  
323 Printer returns for this attribute MUST NOT exceed 80% of the "event-lease-time-interval" in order  
324 to give a client plenty of time to perform another Get-Notifications operation before the event-lease  
325 of the oldest Event Notifications expire.

326

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

328 The value of this attribute is the minimum number of seconds until the event-lease expiration time  
329 for that the Printer will retain-all future Event Notifications associated with the Subscription objects  
330 generating the requested Event Notifications. Thus this number is the maximum number of seconds  
331 that elapses before this client SHOULD issue this operation again for these ~~subscription-~~  
332 idsSubscription objects. A Printer MUST preserve all Notifications that occur for the number of  
333 seconds specified by this attribute starting at the time it is sent in a response. A client MAY perform  
334 this operation at any time, and a Printer MUST respond with all existing Event Notifications. If a  
335 Printer receives this operation after this time interval, it MAY have discarded some Notifications  
336 since the last response.

337

338

339

340 Group 2: Unsupported Attributes

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

342

343 If the “subscription-ids” attribute contained subscription-ids that do not exist, the Printer returns  
344 them in this group as value of the “subscription-ids” attribute.

345

346 Group 3 through N: Notification Attributes

347 The Printer object responds with one Event Notification per Group for each ~~pending~~ Notification  
348 that meets the criteria specified by the ~~subscription-ids attribute and requesting user~~  
349 ~~namerequest~~. (see [ipp-ntfy]).350 **5 Extension to Print-Job, Print-URI, Create-Job, Create-Printer-Subscription**  
351 **and Create-Printer-Subscription**

## 352 5.1 Response

353 When Print-Job, Print-URI or Create-Job contains a “job-notify” attribute and the “notify-recipient” is 'ipp',  
354 the response contains two additional Operation Attributes that pertain to subscriptions.355 When Create-Job-Subscription or Create-Printer-Subscription operation contains a “notify-recipient” that is  
356 'ipp', the response contains two additional Operation Attributes that pertain to subscriptions.

357 Group 1: Operation Attributes

358 "recommended-time-interval" (integer(0:MAX)):

359 The value of this attribute is the recommended number of seconds that SHOULD elapses before the  
360 client SHOULD perform the Get-Notification operation for the first time with any Subscription  
361 objects ~~subscription-ids~~ returned with this job. A client MAY perform the Get-Notification  
362 operation at any time, and a Printer MUST respond with all existing Notifications. A client observes  
363 this value in order to be a “good network citizen”. The value that a Printer returns for this attribute  
364 MUST NOT exceed 80% of the "event-lease-time-interval" in order to give a client plenty of time to  
365 perform another Get-Notifications operation before the event-lease of the oldest events expire.

366

367

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

369 The value of this attribute is the minimum number of seconds until the event-lease expiration time  
370 for that the Printer will retain all future Event Notifications associated with the Subscription objects  
371 generating the requested Event Notifications. Thus this number is the maximum number of seconds  
372 that elapses before a client SHOULD perform the Get-Notification operation for the first time with  
373 any Subscription objects ~~subscription-ids~~ returned with this job. A Printer MUST preserve all  
374 Notifications that occur for the number of seconds specified by this attribute starting at the time it is

375 sent in a response. A client MAY perform the Get-Notification operation at any time, and a Printer  
376 MUST respond with all ~~existing~~ ~~pending~~ Event Notifications. If a Printer receives a Get-  
377 Notification operation after this time interval, it may have discarded some Notifications since the  
378 last response.

379

## 380 6 Encoding

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

382 0x00??

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

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

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

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

## 390 7 IANA Considerations

391 There is nothing to register.

## 392 8 Internationalization Considerations

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

## 399 9 Security Considerations

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

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

## 410 10 References

411 [ipp-mod]

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

414 [ipp-ntfy]

415 Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing  
416 Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-02.txt>, March 8, 2000.

417 [ipp-pro]

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

420 [rfc2026]

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

422 [RFC2616]

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

## 425 11 Authors' Addresses

426

427 Carl-Uno Manros  
428 Xerox Corporation  
429 701 Aviation Blvd.  
430 El Segundo, CA 90245

431

432 Phone: 310-333-  
433 Fax: 310-333-5514  
434 e-mail: manros@cp10.es.xerox.com

435

436 Tom Hastings  
437 Xerox Corporation  
438 737 Hawaii St. ESAE 231  
439 El Segundo, CA 90245

440

441 Phone: 310-333-6413

442 Fax: 310-333-5514  
443 e-mail: [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)  
444

445 Robert Herriot  
446 Xerox Corp.  
447 3400 Hill View Ave, Building 1  
448 Palo Alto, CA 94304  
449

450 Phone: 650-813-7696  
451 Fax: 650-813-6860  
452 e-mail: [robert.herriot@pahv.xerox.com](mailto:robert.herriot@pahv.xerox.com)  
453

454 Harry Lewis  
455 IBM  
456 P.O. Box 1900  
457 Boulder, CO 80301-9191  
458

459 Phone: (303) 924-5337  
460 FAX:  
461 e-mail: [harryl@us.ibm.com](mailto:harryl@us.ibm.com)  
462

## 463 **12 Full Copyright Statement**

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

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

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

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