

1 INTERNET-DRAFT
2 <draft-ietf-ipp-notify-poll-02.txt>

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

10 Internet Printing Protocol (IPP):
11 **The ‘ipp-get’ Notification Polling Method**

12
13 Copyright (C) The Internet Society (2000). All Rights Reserved.

14 Status of this Memo

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

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

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

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

23 **Abstract**

24 The notification extension document [ipp-ntfy] defines operations that a client can perform in order to
25 create *Subscription Objects* in a Printer and carry out other operations on them. A Subscription Object
26 represents a Subscription abstraction. The Subscription Object specifies that when one of the specified
27 *Events* occurs, the Printer sends an asynchronous *Event Notification* to the specified *Notification Recipient*
28 via the specified *Delivery Method* (i.e., protocol).

29 The notification extension document [ipp-ntfy] specifies that each Delivery Method is defined in another
30 document. This document is one such document, and it specifies the ‘ipp-get’ delivery method.

31 The ‘ipp-get’ Delivery Method is a ‘pull’ Delivery Method. That is, the Printer saves Event Notification for
32 a period of time and expects the Notification Recipient to fetch the Event Notifications.

33 When a Printer supports this Delivery Method, it holds each Event Notification for an amount of time,
34 called the *Event Notification Lease Time*.

35 When a Notification Recipient wants to receive Event Notifications, it performs an IPP operation called
36 ‘Get-Notifications’, which this document defines. This operation causes the Printer to return all Event
37 Notifications held for the Notification Recipient along with information that tells the client when to perform

38 this operation again.

39 The full set of IPP documents includes:

- 40 Design Goals for an Internet Printing Protocol [RFC2567]
- 41 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
- 42 Internet Printing Protocol/1.1: Model and Semantics [ipp-mod]
- 43 Internet Printing Protocol/1.1: Encoding and Transport [ipp-pro]
- 44 Internet Printing Protocol/1.1: Implementer's Guide [ipp-iig]
- 45 Mapping between LPD and IPP Protocols [RFC2569]
- 46 Internet Printing Protocol/1.0 & 1.1: IPP Event Notification Specification [ipp-ntfy]

47

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

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

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

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

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

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

73 The "Event Notification Specification" document describes an extension to the IPP/1.0, IPP/1.1, and future
74 versions. This extension allows a client to subscribe to printing related Events. Subscriptions are modeled
75 as *Subscription Objects*. The Subscription Object specifies that when one of the specified *Event* occurs, the
76 Printer sends an asynchronous *Event Notification* to the specified *Notification Recipient* via the specified
77 *Delivery Method* (i.e., protocol). A client associates Subscription Objects with a particular Job by
78 performing the Create-Job-Subscriptions operation or by submitting a Job with subscription information. A
79 client associates Subscription Objects with the Printer by performing a Create-Printer-Subscriptions

80 operation. Four other operations are defined for Subscription Objects: Get-Subscriptions-Attributes, Get-
81 Subscriptions, Renew-Subscription, and Cancel-Subscription.

Table of Contents

82			
83			
84	1	Introduction	6
85	2	Terminology	6
86	3	Model and Operation	7
87	4	Get-Notifications operation	8
88	4.1	GET-NOTIFICATIONS REQUEST	9
89	4.2	GET-NOTIFICATIONS RESPONSE	10
90	5	Extensions to Print-Job, Print-URI, Create-Job, Create-Printer-Subscription and Create-Printer-Subscription	13
91			
92	5.1	RESPONSE	14
93	6	Encoding.....	14
94	7	IANA Considerations	14
95	8	Internationalization Considerations.....	14
96	9	Security Considerations.....	15
97	10	References	15
98	11	Authors' Addresses.....	15
99	12	Full Copyright Statement.....	16
100			

101 1 Introduction

102 The notification extension document [ipp-ntfy] defines operations that a client can perform in order to
103 create *Subscription Objects* in a Printer and carry out other operations on them. A Subscription Object
104 represents a Subscription abstraction. The Subscription Object specifies that when one of the specified
105 *Events* occurs, the Printer sends an asynchronous *Event Notification* to the specified *Notification Recipient*
106 via the specified *Delivery Method* (i.e., protocol).

107 The notification extension document [ipp-ntfy] specifies that each Delivery Method is defined in another
108 document. This document is one such document, and it specifies the 'ipp-get' delivery method.

109 The 'ipp-get' Delivery Method is a 'pull' Delivery Method. That is, the Printer saves Event Notification for
110 a period of time and expects the Notification Recipient to fetch the Event Notifications.

111 When a Printer supports this Delivery Method, it holds each Event Notification for an amount of time,
112 called the *Event Notification Lease Time*.

113 When a Notification Recipient wants to receive Event Notifications, it performs an IPP operation called
114 'Get-Notifications', which this document defines. This operation causes the Printer to return all Event
115 Notifications held for the Notification Recipient along with information that tells the client when to perform
116 this operation again.

117 2 Terminology

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

119 Capitalized terms, such as **MUST**, **MUST NOT**, **REQUIRED**, **SHOULD**, **SHOULD NOT**, **MAY**,
120 **NEED NOT**, and **OPTIONAL**, have special meaning relating to conformance to this specification. These
121 terms are defined in [ipp-mod section 13.1 on conformance terminology, most of which is taken from RFC
122 2119 [RFC2119].

123 **Event Notification Lease:** The lease that is associated with an Event Notification. When the lease expires,
124 the Printer discards the associated Event Notification.

125 **Event Notification Lease Time:** The expiration time assigned to a lease that is associated with an Event
126 Notification.

127 **Event Notification Attributes Group:** The attributes group in a response that contains attributes that are
128 part of an Event Notification.

129 For other capitalized terms that appear in this document, see [ipp-ntfy].

130 3 Model and Operation

131 In a Subscription Creation Operation, when the value of the “notify-recipient-uri” attributes has the scheme
132 “ipp-get”, the client is requesting that the Printer use the ‘ipp-get’ Delivery Method for the Event
133 Notifications associated with the new Subscription Object. The client MUST choose a value for the address
134 part of the “notify-recipient-uri” attribute that uniquely identifies the Notification Recipient.

135 When an Event occurs, the Printer MUST generate an Event Notification and MUST assign it an Event
136 Notification Lease Time. The Printer MUST hold an Event Notification for its assigned Event Notification
137 Lease Time and MUST discard it when its Event Notification Lease Time expires. The Printer MAY assign
138 the same Event Notification Lease Time to each Event Notification or it MAY assign a different time.

139 When a Notification Recipient wants to receive Event Notifications, it performs the Get-Notifications
140 operation, which causes the Printer to return all unexpired Event Notifications held for the Notification
141 Recipient along with two time-intervals.

142 The first returned time-interval is the suggested time a Notification Recipient should wait before
143 performing the Get-Notifications operation again. The second time-interval is the time that Event
144 Notification Leases begin to expire for Event Notifications created after the Get-Notifications operation. A
145 Notification Recipient SHOULD perform this operation at the suggested time and somewhat before the
146 Event Notification Leases begin to expire.

147 The Notification Recipient identifies its own Event Notifications with a “notify-recipient-uri” Operation
148 attribute in the request. It matches any Event Notifications associated with a Subscription Object whose
149 “notify-recipient-uri” attribute has the same value as the “notify-recipient-uri” Operation attribute of the
150 request. To avoid getting Event Notification that belong to another Notification Recipient, a client
151 SHOULD pick values for the “notify-recipient-uri” attribute that are unique, e.g. the client’s host address.

152 If a Notification Recipient performs the Get-Notifications operation twice in quick succession, it will
153 receive nearly the same Event Notification both times. There are two possible differences. Some old Event
154 Notifications may not be present in the second response because their Event Notification Leases have
155 expired. Some new Event Notifications may be present in the second response but not the first response.

156 The Printer may keep the channel open if the suggested time-interval is sufficiently short, but in any case
157 the client performs a new Get-Notifications operation each time it wants more Event Notifications. Since
158 the time interval between consecutive client requests is normally less than the Event Notification Lease
159 Time, consecutive responses will normally contain some events that are identical. The youngest ones in
160 the previous response will become the oldest in the next response. The client is expected to filter out these
161 duplicates, which is easy to do because of the sequence number in each Event Notification. The reason for
162 not removing the Event Notifications from the Printer with every Get-Notifications request, is so that
163 multiple Notification Recipients can be polling the same Subscription Object and so the Get-Notification
164 operation satisfies the rule of idempotency. The former is useful if someone is logged in to several
165 desktops at the same time and wants to see the same events at both places. The latter is useful if the
166 network loses the response.

167 4 General Information

168 According to the notification extension document [ipp-ntfy], this document MUST contain the following
169 information:

- 170 1. The URL scheme name for the Delivery Method is: 'ipp-get'
- 171 2. Printer support for this delivery method is OPTIONAL.
- 172 3. For Event Notification content, a Printer MUST use the following transport and delivery protocol, i.e.,
173 entire network stack: IPP with one new operation.
- 174 4. Several Event Notifications can be combined into a compound Event Notification. See section 5.
- 175 5. The Notification Recipient MUST initiate the Delivery Method
- 176 6. The Delivery Method is Machine Consumable.
- 177 7. The representation and encoding for each value is the same as for IPP (see section 5).
- 178 8. In the Event Notification content, a Printer MUST send all attributes specified in section 5.
- 179 9. Frequently occurring Events NEED NOT be moderated because the Delivery Method is a 'pull'
180 Delivery Method. An implementation of the Get-Notifications operation SHOULD consider how often
181 it recommends a Notification Recipient to poll again.
- 182 10. This Delivery Method has the same latency and reliability as the underlying HTTP transport.
- 183 11. This Delivery Method has the same security aspects as the underlying HTTP transport.
- 184 12. This Delivery Method has no content length restrictions.
- 185 13. There are no additional values that a Printer MUST send in a Notification content.
- 186 14. There are no additional Subscription Template and/or Subscription Description attributes.
- 187 15. There are no additional Printer Description attributes.

188 5 Get-Notifications operation

189 This operation causes the Printer to return all Event Notifications held for the Notification Recipient along
190 with information about when to perform this operation again.

191 A Printer MUST support this operation.

192 When a Printer performs this operation, it MUST return all and only those Event Notifications:

- 193 a) Whose associated Subscription Object's "notify-recipient-uri" attribute equals the "notify-
194 recipient-uri" Operation attribute AND

- 195 b) Whose associated Subscription Object's "notify-recipient-uri" attribute has a scheme value of
196 'ipp-get' AND
- 197 c) Whose Event Notification Lease Time has not yet expired AND
- 198 d) Where the Notification Recipient is the owner of or has read-access rights to the associated
199 Subscription Object.

200 When a Printer performs this operation, it MUST also return two time-intervals:

- 201 a) the suggested time for a Notification Recipient to perform the Get-Notifications operation again.
- 202 b) the time at which the Printer will begin to discard Event Notifications that occur after this
203 operation. This may be the Event Notification Lease Time (see section 5.2 for details).

204 Note: the Subscription Creation Operations also return these two time-intervals (see section 6).

205 The Printer MUST respond to this operation immediately with whatever Event Notifications it currently
206 holds. It MUST NOT wait for additional Events to occur before sending a response.

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

209 *Access Rights:* If the policy of the Printer is to allow all users to access all Event Notifications, then the
210 Printer MUST accept this operation from any user. Otherwise, the authenticated user (see [ipp-mod] section
211 8.3) performing this operation MUST either be the owner of each Subscription Object identified by the
212 "notify-recipient-uri" Operation attribute (as determined during a Subscription Creation Operation) or an
213 operator or administrator of the Printer (see [ipp-mod] Sections 1 and 8.5). Otherwise, the IPP object
214 MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-
215 error-not-authorized' as appropriate.

216 5.1 Get-Notifications Request

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

218 Group 1: Operation Attributes

219 Natural Language and Character Set:

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

222

223 Target:

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

226

227 Requesting User Name:

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

230

231 "notify-recipient-uri" (url):

232 The client MUST supply this attribute. The Printer object MUST support this attribute. The Printer
233 matches the value of this attribute (byte for byte with no case conversion) against the value of the
234 "notify-recipient-uri" in each Subscription Object in the Printer. If there are no matches, the IPP
235 Printer MUST return the 'client-error-not-found' status code. For each matched Subscription
236 Object, the IPP Printer MUST return all unexpired Event Notifications associated with it.

237

238 Note: this attribute allows a subscribing client to pick URLs that are unique, e.g. the client's own
239 URL or a friend's URL, which in both cases is likely the URL of the person's host. An application
240 could make a URL unique for each application.

241 5.2 Get-Notifications Response

242 The following groups of attributes are part of the Get-Notifications Response:

243 Group 1: Operation Attributes

244 Status Message:

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

248

249 The Printer can return any status codes defined in [ipp-mod]. The following is a description of the
250 important status codes:

251

252 **successful-ok:** the response contains all Event Notification associated with the specified
253 "notify-recipient-uri". If the specified Subscription Objects have no associated Event
254 Notification, the response MUST contain zero Event Notifications.

255 **client-error-not-found:** The Printer has no Subscription Object's whose "notify-recipient-uri"
256 attribute equals the "notify-recipient-uri" Operation attribute.

257

258 Natural Language and Character Set:

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

261

262 The Printer MUST use the values of "notify-charset" and "notify-natural-language", respectively,
263 from one Subscription Object associated with the Event Notifications in this response.

264

265 Normally, there is only one matched Subscription Object, or the value of the "notify-charset" and
266 "notify-natural-language" attributes is the same in all Subscription Objects. If not, the Printer MUST
267 pick one Subscription Object from which to obtain the value of these attributes. The algorithm for
268 picking the Subscription Object is implementation dependent. The choice of natural language is not
269 critical because 'text' and 'name' values can override the "attributes-natural-language" Operation
270 attribute. The Printer's choice of charset is critical because a bad choice may leave it unable to send
271 some 'text' and 'name' values accurately.

272

273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316

“suggested-ask-again-time-interval” (integer(0:MAX)):

The value of this attribute is the suggested number of seconds that SHOULD elapse before the client performs the Get-Notifications operation again for these Subscription Objects. A client MAY perform the Get-Notifications operation at any time, and a Printer MUST respond with all unexpired Event Notifications. A Notification Recipient waits until this time interval has elapsed in order to be a “good network citizen”. It is RECOMMENDED that the value of this attribute be 80% of the “begin-to-expire-time-interval” (see the next attribute) in order to give a Notification Recipient plenty of time to perform the Get-Notifications operation again before new Event Notifications expire.

“begin-to-expire-time-interval” (integer(0:MAX)):

The value of this attribute is the minimum number of seconds that MUST elapse before Event Notification Leases begin to expire on Event Notifications produced by matching Subscriptions Objects after the Printer sends the Get-Notifications response. The Printer MUST discard an Event Notification when its Event Notification Lease has expired. That is, if the Printer performs the Get-Notifications operation before the time specified by the “begin-to-expire-time-interval” attribute returned in the previous operation, the Printer MUST still have all of the Event Notifications that have occurred since the previous operation. If the Printer assigns the same Event Notification Lease Time to all Event Notifications, the value of this attribute MUST equal the Event Notification Lease Time. If a Notification Recipient waits until after this time or even slightly less than this time, the Notification Recipient MUST expect to lose some Event Notifications.

“printer-up-time” (integer(0:MAX)):

The value of this attribute is the Printer’s “printer-up-time” attribute at the time the Printer sends this response. Because each Event Notification also contains the value of this attribute when the event occurred, the value of this attribute lets a Notification Recipient know when each Event Notification occurred relative to the time of this response.

Group 2: Unsupported Attributes

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

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

Group 3 through N: Event Notification Attributes

The Printer responds with one Event Notification Attributes Group per matched Event Notification. The matched Event Notifications are all un-expired Event Notification associated with the matched Subscription Objects. Each Event Notification Group MUST start with an ‘event-notification-attributes-tag’ (see the section “Encodings of Additional Attribute Tags” in [ipp-ntfy]).

Each attribute is encoded using the IPP rules for encoding attributes [ipp-pro] and may be encoded in any order. Note: the Get-Jobs response in [ipp-mod] acts as a model for encoding multiple groups of attributes.

317 Each Event Notification Group MUST contain all of attributes specified in section 9.1 (“Content of
 318 Machine Consumable Event Notifications”) of [ipp-ntfy] with exceptions denoted by asterisks in the
 319 tables below.

320

321 The tables below are copies of the tables in section 9.1 (“Content of Machine Consumable Event
 322 Notifications”) of [ipp-ntfy] except that each cell in the “Sends” column is a “MUST”.

323

324 For an Event Notification for all Events, the Printer includes the following attributes.

325

Table 1 – Attributes in Event Notification Content

Source Value	Sends	Source Object
notify-subscription-id (integer(1:MAX))	MUST	Subscription
notify-printer-uri (uri)	MUST	Subscription
notify-subscribed-event (type2 keyword)	MUST	Event Notification
printer-up-time (integer(MIN:MAX))	MUST	Printer
printer-current-time (dateTime)*	MUST	Printer
notify-sequence-number (integer (0:MAX))	MUST	Subscription
notify-charset (charset)	MUST	Subscription
notify-natural-language (naturalLanguage)	MUST	Subscription
notify-user-data (octetString(63)) **	MUST	Subscription
notify-text (text)	MUST	Event Notification
attributes from the “notify-attributes” attribute ***	MUST	Printer
attributes from the “notify-attributes” attribute ***	MUST	Job
attributes from the “notify-attributes” attribute ***	MUST	Subscription

326

327 * The Printer MUST send “printer-current-time” if and only if it supports the “printer-current-time”
 328 attribute on the Printer object.

329

330 ** If the associated Subscription Object does not contain a “notify-user-data” attribute, the Printer
 331 MUST send an octet-string of length 0.

332

333 *** If the “notify-attributes” attribute is present on the Subscription Object, the Printer MUST send
 334 all attributes specified by the “notify-attributes” attribute. Note: if the Printer doesn't support the
 335 “notify-attributes” attribute, it is not present on the associated Subscription Object.
 336

337 For Event Notifications for Job Events, the Printer includes the following additional attributes.

338 **Table 2 – Additional Attributes in Event Notification Content for Job Events**

Source Value	Sends	Source Object
job-id (integer(1:MAX))	MUST	Job
job-state (type1 enum)	MUST	Job
job-state-reasons (1setOf type2 keyword)	MUST	Job
job-impressions-completed (integer(0:MAX)) *	MUST	Job

339 * The Printer MUST send the “job-impressions-completed” attribute in an Event Notification only
 340 for the combinations of Events and Subscribed Events shown in Table 3.
 341
 342

343 **Table 3 – Combinations of Events and Subscribed Events for “job-impressions-completed”**

Job Event	Subscribed Job Event
'job-progress'	'job-progress'
'job-completed'	'job-completed'
'job-completed'	'job-state-changed'

344 For Event Notification for Printer Events, the Printer includes the following additional attributes.
 345

346 **Table 4 – Additional Attributes in Event Notification Content for Printer Events**

Source Value	Sends	Source Object
printer-state (type1 enum)	MUST	Printer
printer-state-reasons (1setOf type2 keyword)	MUST	Printer
printer-is-accepting-jobs (boolean)	MUST	Printer

347 6 Extensions to Subscription Creation Operations

348 6.1 Response

349 When a Subscription Creation Operation contains a “notify-recipient-uri” attribute and the scheme in its
350 value is ‘ipp-get’, the response MUST contain two additional Operation Attributes that pertain to this
351 Delivery Method. Note: Subscription Creation Operations include: Print-Job, Print-URI, Create-Job,
352 Create-Job-Subscriptions and Create-Printer-Subscriptions.

353 Group 1: Operation Attributes

354 “suggested-ask-again-time-interval” (integer(0:MAX)):

355 This attribute has the same meaning as the “suggested-ask-again-time-interval” attribute in the Get-
356 Notifications operation except that it suggests when to perform the Get-Notifications operation for
357 the first time on all Subscription Objects in the response whose “notify-recipient-uri” scheme is
358 ‘ipp-get’.

359

360 “begin-to-expire-time-interval” (integer(0:MAX)):

361 This attribute has the same meaning as the “begin-to-expire-time-interval” attribute in the Get-
362 Notifications operation except that it indicates when the Event Notification Lease begins to expire
363 for all Subscription Objects in the response whose “notify-recipient-uri” scheme is ‘ipp-get’.

364 7 Encoding

365 The operation-id assigned for the Get-Notifications operation is:

366 0x001C

367 and should be added to the next version of [ipp-mod] section 4.4.15 “operations-supported”.

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

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

371 8 IANA Considerations

372 There is nothing to register.

373 9 Internationalization Considerations

374 The IPP Printer MUST localize the “notify-text” attribute as specified in section 14 of [ipp-ntfy].

375 In addition, when the client receives the Get-Notifications response, it is expected to localize the attributes
376 that have the ‘keyword’ attribute syntax according to the charset and natural language requested in the Get-
377 Notifications request.

378 10 Security Considerations

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

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

389 11 References

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

393 [ipp-ntfy]
394 R. Herriot, Hastings, T., Isaacson, S., Martin, J., deBry, R., Shepherd, M., Bergman, R., "Internet
395 Printing Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-04.txt>, June
396 30, 2000.

397 [ipp-pro]
398 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and
399 Transport", draft-ietf-ipp-protocol-v11-05.txt, March 1, 2000.

400 [rfc2026]
401 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

402 [RFC2616]
403 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext
404 Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.

405 12 Authors' Addresses

406
407 Robert Herriot
408 Xerox Corp.
409 3400 Hill View Ave, Building 1
410 Palo Alto, CA 94304
411
412 Phone: 650-813-7696

413 Fax: 650-813-6860
414 e-mail: robert.herriot@pahv.xerox.com

415
416 Tom Hastings
417 Xerox Corporation
418 737 Hawaii St. ESAE 231
419 El Segundo, CA 90245

420
421 Phone: 310-333-6413
422 Fax: 310-333-5514
423 e-mail: hastings@cp10.es.xerox.com

424
425 Carl-Uno Manros
426 Xerox Corporation
427 701 Aviation Blvd.
428 El Segundo, CA 90245

429
430 Phone: 310-333-
431 Fax: 310-333-5514
432 e-mail: manros@cp10.es.xerox.com

433
434 Harry Lewis
435 IBM
436 P.O. Box 1900
437 Boulder, CO 80301-9191
438
439 Phone: (303) 924-5337
440 FAX:
441 e-mail: harryl@us.ibm.com
442

443 **13 Full Copyright Statement**

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

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

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

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