

1 INTERNET-DRAFT **There are 10 issues highlighted like this.**
2 <draft-ietf-ipp-notify-get-00.txt>

3
4 Carl-Uno Manros
5 Tom Hastings
6 Xerox Corp.
7 February 3, 2000

8 Internet Printing Protocol/1.1: **The 'ipp-notify-get' Notification Delivery Method**

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

10 **ISSUE 01 - What should the name of this delivery method and protocol be that we use in the title of this document?**

11 **ISSUE 02 - What should the scheme name be? Consider 'ipp-notify-get' a working title, until we see several schemes. The 'ipp-notify-poll', 'ipp-notify-sent', and 'ipp-snmpp' delivery methods are our other examples. The IETF likes words or well-recognized acronyms, not abbreviations in scheme names, so lets include "notify"?**

12 **ISSUE 03 - Should the scheme name be used in the title?**

13 Status of this Memo

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

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

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

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

18 **Abstract**

19 The IPP notification specification [ipp-ntfy] is an OPTIONAL extension to IPP/1.0 and IPP/1.1 that requires the definition of one or more delivery methods for dispatching event notification reports to Notification Recipients. This document describes the semantics and syntax of the 'ipp-notify-get' event notification delivery method. For this delivery method, the client uses an explicit IPP Get-Notifications Printer operation in order to request (pull) event Notifications from the IPP Printer. The Get-Notifications request indicates whether the client wants to receive all future events Notifications for (1) any Subscription for which the client is the owner or (2) a particular Subscription object. In either case, the event Notifications are returned as MIME multi-part-related responses to the Get-Notifications request. The HTTP channel is kept open, so that subsequent event Notifications are returned using additional MIME multi-part-related responses.

36 The full set of IPP documents includes:

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

44

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

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

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

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

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

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

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

74

75

Table of Contents

76 1 Introduction4

77 2 Terminology4

78 3 Model and Operation4

79 4 Get-Notifications operation5

80 4.1 GET-NOTIFICATIONS REQUEST6

81 4.2 GET-NOTIFICATIONS RESPONSE7

82 5 Encoding7

83 6 IANA Considerations8

84 7 Internationalization Considerations8

85 8 Security Considerations8

86 9 References9

87 10 Author's Addresses9

88 11 Change History10

89 11.1 CHANGES MADE TO THE DECEMBER 7, 1999 VERSION TO MAKE THE FEBRUARY 3, 2000 VERSION.....10

90 12 Full Copyright Statement.....10

91

92

93 1 Introduction

94 IPP printers that support the OPTIONAL IPP notification extension [ipp-ntfy] either a) accept, store, and
95 use notification subscriptions to generate notification reports and implement one or more delivery methods
96 for notifying interested parties, or b) support a subset of these tasks and farm out the remaining tasks to a
97 Notification Delivery Service. The 'ipp-notify-get' event notification delivery method specified in this
98 document defines a Get-Notifications operation that may be used in a variety of notification scenarios. Its
99 primary intended use is for clients that want to be Notification Recipients to explicitly request (pull) event
100 Notifications from the IPP Printer upon request. However, the Get-Notifications operation may also be
101 used by Notification Delivery Services to request (pull) event Notifications from an IPP Printer for
102 subsequent distribution to the Ultimate Notification Recipients. The HTTP channel is kept open, so that
103 subsequent event Notifications are returned using additional MIME multi-part-related responses.

104 2 Terminology

105 This document uses terms such as "attributes", "keywords", and "support". These terms have special
106 meaning and are defined in the model terminology [ipp-mod] section 12.2.

107 Capitalized terms, such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED
108 NOT, and OPTIONAL, have special meaning relating to conformance. These terms are defined in [ipp-
109 mod] section 12.1 on conformance terminology, most of which is taken from RFC 2119 [RFC2119].

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

111 **REQUIRED:** if an implementation supports the extensions described in this document, it **MUST**
112 support a **REQUIRED** feature.

113 **OPTIONAL:** if an implementation supports the extensions described in this document, it **MAY** support
114 an **OPTIONAL** feature.

115 **Notification Recipient** - See [ipp-ntfy]

116 **Subscription object** - See [ipp-ntfy]

117 **Ultimate Notification Recipient** - See [ipp-ntfy]

118 3 Model and Operation

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

123 When creating each Subscription object, the client supplies the "notify-recipient" (uri) attribute. The
124 "notify-recipient" attribute specifies both a single Notification Recipient that is to receive the event
125 Notifications when subsequent events occur and the method for notification delivery that the IPP Printer is
126 to use. For the Notification delivery method defined in this document, the notification method is 'ipp-
127 notify-get', and the Notification Recipient is omitted, since any client that is authenticated (1) as an operator

128 or administrator or (2) as the owner of the Subscription object can initiate a Get-Notifications operation for
129 that Subscription object. Thus a single user can login at different places, say his/her office, the lab, and/or
130 several desktops in the same room, and receive the same event Notifications from a single Subscription
131 object.

132 For the 'ipp-notify-get' event notification delivery method defined in the document, the client who created
133 the Subscription objects is also the Notification Recipient. The client issues a Get-Notifications Printer
134 operation in order to initiate the delivery of the next event Notifications that occur. The client can indicate
135 in the Get-Notifications request whether it wants to receive all future event Notifications for (1) any
136 existing or future Subscription objects for which it is the owner or (2) a particular Subscription object (for
137 which it MUST be the owner). In either case, the Notifications are returned as MIME multi-part-related
138 responses to the Get-Notifications request. The HTTP channel is kept open for an indefinite period, so that
139 the IPP Printer continues to return additional parts of the MIME multi-part-related responses for each event
140 Notification as it occurs. Either the client or the IPP Printer can disconnect the HTTP connection.
141 However, if the IPP Printer grants an HTTP connection it SHOULD disconnect only under unusual
142 circumstances.

143 **ISSUE 04: Is there a limit to the number of outstanding Get-Notifications requests that an IPP Printer**
144 **supports? What is this number? How does it relate to the maximum number of Subscriptions? Can the**
145 **client determine the number?**

146 **ISSUE 05: Should an implementation be able to queue event Notifications, so that a client can get event**
147 **Notifications that had occurred prior to the Get-Notifications? If so, how long does the IPP Printer keep the**
148 **event Notifications before discarding them (for this delivery method only)? The lease time of the**
149 **Subscription object? If this is possible, should the subscriber get to say whether to queue or not, or is it just**
150 **baked into the implementation. If the former, does the subscriber indicate via a parameter in the**
151 **notification method URL? If the latter, how does a client discover whether event Notifications are queued**
152 **or not? Should we have two different notification methods, one the queues and one that doesn't?**

153 From the December meeting:

154 It was suggested that any "notification queuing service" should
155
156 be the responsibility of the Notification Recipient, not the
157 Printer. However, the Issue was not completely resolved.

158 **4 Get-Notifications operation**

159 This REQUIRED operation allows the client to request that future event Notifications be delivered as
160 MIME multi-part-related responses to this request. The client MUST be the owner of the Subscription
161 objects that are involved and the delivery method specified when the Subscription objects were created
162 MUST be 'ipp-notify-get'. However, the client can and SHOULD issue the Get-Notifications request before
163 having created any Subscription objects, in order not to miss any event Notifications.

164 The IPP Printer MUST accept the request in any state (see [ipp-mod] "printer-state", "printer-state-
 165 reasons", and "printer-is-accepting-jobs" attributes) and MUST remain in the same state with the same
 166 "printer-state-reasons".

| Current "printer-state" | New "printer-state" | new "printer- state-reasons" | IPP Printer's response status code and action: |
|----------------------------|------------------------|---------------------------------|---|
| 'idle' | 'idle' | no change | 'successful-ok' |
| 'processing' | 'processing' | no change | 'successful-ok' |
| 'stopped' | 'stopped' | no change | 'successful-ok' |

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

173 4.1 Get-Notifications Request

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

175 Group 1: Operation Attributes

176 Natural Language and Character Set:

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

179 Target:

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

182 Requesting User Name:

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

185 "subscription-id" (integer(1:MAX)):

186 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It
 187 is an integer value that identifies the Subscription object for which event Notifications are being
 188 requested. If the client supplies this attribute, but the Subscription object is not found, the IPP
 189 Printer MUST return the 'client-error-not-found' status code. If the client does not supply this
 190 attribute, then the IPP Printer returns event Notifications for all Subscription objects for which the
 191 client is the owner and the "notify-recipients" attribute is 'ipp-notify-get'. It is not an error if there
 192 are currently no Subscription objects for this client; the client can create Subscription objects later
 193 that will start returning event Notifications as responses to this operation.
 194
 195
 196

197 4.2 Get-Notifications Response

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

200 Group 1: Operation Attributes

201 Status Message:

202 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY
203 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation
204 attribute as described in [ipp-mod] sections 13 and 3.1.6.
205

206 Natural Language and Character Set:

207 The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod]
208 section 3.1.4.2.
209

210 Group 2: Unsupported Attributes

211 See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes.
212

213 Group 3: Notification Attributes

214 The Printer object responds with one event Notification (see [ipp-ntfy]). If there are multiple events
215 that occur at the same time, the Printer object returns them in separate MIME multi-part-related
216 responses, each as separate IPP operation responses, as well. The HTTP channel is kept open for an
217 indefinite period, so that the IPP Printer continues to return additional parts of the MIME multi-part-
218 related responses for each event Notification as it occurs.

219 ISSUE 06 - Is this correct for MIME multi-part-related responses? This need prototyping.

220 ISSUE 07 - What happens if 100 continue isn't supported?

221 ISSUE 08 - What happens if HTTP keep alive isn't supported?

222 **5 Encoding**

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

224 0x00??

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

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

228
229 notification-attributes-tag = %x07 ; tag of 07

230 ISSUE 9 - The problem with assigning new tags for every new kind of attributes and objects,
 231 is that an implementation that does private or experimental operations that have new kinds
 232 of attributes and/or objects, will be forced to either overload some existing tag or use one of
 233 the tags reserved for future standardization. See email from Ned Freed about the need to
 234 clarify [ipp-pro] about:

0x06-0x0e reserved for future delimiters
 0x0F reserved for future chunking-end-of-attributes-tag

235
 0x11 reserved for future 'default'
 0x14-0x1F reserved for future "out-of-band" values.

236 Whereas if we had a generic tag, that same tag could be used for the private and
 237 experimental operations. The Printer and the client then uses the operation-id itself to
 238 determine what kind of attributes or object is being passed in the request or returned in the
 239 response, respectively.

240 Another possible approach would be to assign one tag for private use and then keep
 241 assigning new tags for standard uses, such as Subscription (0x6) and Notification (0x7).

242 6 IANA Considerations

243 IANA will be asked to register this 'ipp-notify-get' notification delivery scheme.

244 ISSUE 10: Any notification delivery scheme has to be registered with IANA, since it is a URL scheme,
 245 correct?

246 7 Internationalization Considerations

247 With the 'ipp-notify-get' method defined in this document, the client cannot request the Human Consumable
 248 form by supplying the "notify-text-format" operation attribute (see [ipp-ntfy]). Therefore, the IPP Printer
 249 does not have to perform any localization with this notification delivery method. However, the client when
 250 it receives the Get-Notifications response is expected to localize the attributes that have the 'keyword'
 251 attribute syntax according to the charset and natural language requested in the Get-Notifications request.

252 8 Security Considerations

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

258 Unlike other event Notification delivery methods in which the IPP Printer initiates the event Notification,
 259 with the method defined in this document, the Notification Recipient is the client who issues the Get-

260 Notifications operation. Therefore, there is no chance of "spam" notifications with this method.
261 Furthermore, such a client can close down the HTTP channel at any time, and so can avoid future unwanted
262 event Notifications at any time.

263 **9 References**

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

267 [ipp-ntfy]
268 Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing
269 Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-02.txt>, February 2,
270 2000.

271 [ipp-pro]
272 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and
273 Transport", draft-ietf-ipp-protocol-v11-03.txt, June, 1999.

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

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

279 **10 Author's Addresses**

280 Carl-Uno Manros
281 Xerox Corporation
282 737 Hawaii St. ESAE 231
283 El Segundo, CA 90245
284
285 Phone: 310-333-8273
286 Fax: 310-333-5514
287 e-mail: cmanros@cp10.es.xerox.com
288

289 Tom Hastings
290 Xerox Corporation
291 737 Hawaii St. ESAE 231
292 El Segundo, CA 90245
293
294 Phone: 310-333-6413
295 Fax: 310-333-5514
296 e-mail: hastings@cp10.es.xerox.com

297

298 **11 Change History**

299 This section lists the changes made to the document. It does not list additions or deletions of issues.

300 11.1 Changes made to the December 7, 1999 version to make the February 3, 2000 version

301 The following changes were made to the December 7, 1999 version to make the February 3, 2000 version:

- 302 1. Changed the scheme name and title from 'ipp-get' to 'ipp-notify-get'.
- 303 2. Changed the tag delimiter from generic-attributes-tag to notification-attributes-tag as agreed at the
304 December IPP WG meeting.

305 **12 Full Copyright Statement**

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

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

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

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