

1 INTERNET-DRAFT
2 <draft-ietf-ipp-notify-get-021.txt>
3 [Target category: standards track]

Robert Herriot (editor)
Xerox Corp.
Carl Kugler
IBM, Corp.
Harry Lewis
IBM, Corp.

~~November 16, 2000~~ February 28, 2001

8 Internet Printing Protocol (IPP):
9 **The ‘ippget’ Delivery Method for Event Notifications**

11 Copyright (C) The Internet Society (20010). 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 [rfc2026].
15 Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its
16 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 notification extension document [ipp-ntfy] defines operations that a client can perform in order to create
24 *Subscription Objects* in a Printer and carry out other operations on them. A Subscription Object represents a
25 Subscription abstraction. The Subscription Object specifies that when one of the specified *Events* occurs, the
26 Printer sends an asynchronous *Event Notification* to the specified *Notification Recipient* via the specified
27 *Delivery Method* (i.e., protocol).

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

30 The ‘ippget’ Delivery Method is a ‘pull and push’ Delivery Method. That is, the Printer saves Event
31 Notification for a period of time and expects the Notification Recipient to fetch the Event Notifications (the
32 pull part). The Printer continues to send Event Notifications to the Notification Recipient as Events occur (the
33 push part) if the client has selected the option to wait for additional Event Notifications.

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

36 When a Notification Recipient wants to receive Event Notifications, it performs an IPP operation called ‘Get-
37 Notifications’, which this document defines. This operation causes the Printer to return all Event Notifications
38 held for the Notification Recipient. If the Notification Recipient has selected the option to wait for additional
39 Event Notifications, the Printer continues sending Event Notifications to the Notification Recipient as additional
40 Events occur.

41

42 The basic set of IPP documents includes:

- 43 Design Goals for an Internet Printing Protocol [RFC2567]
- 44 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
- 45 Internet Printing Protocol/1.1: Model and Semantics [RFC2911]
- 46 Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
- 47 Internet Printing Protocol/1.1: Implementer’s Guide [ipp-ig]
- 48 Mapping between LPD and IPP Protocols [RFC2569]
- 49 Internet Printing Protocol/1.0 & 1.1: IPP Event Notification Specification [ipp-ntfy]

50

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

56 The “Rationale for the Structure and Model and Protocol for the Internet Printing Protocol” document
57 describes IPP from a high level view, defines a roadmap for the various documents that form the suite of IPP
58 specification documents, and gives background and rationale for the IETF working group’s major decisions.

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

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

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

73 The “Mapping between LPD and IPP Protocols” document gives some advice to implementers of gateways
74 between IPP and LPD (Line Printer Daemon) implementations.

75 The “Event Notification Specification” document describes an extension to the IPP/1.0, IPP/1.1, and future
76 versions. This extension allows a client to subscribe to printing related Events. Subscriptions are modeled as
77 *Subscription Objects*. The Subscription Object specifies that when one of the specified *Event* occurs, the
78 Printer sends an asynchronous *Event Notification* to the specified *Notification Recipient* via the specified
79 *Delivery Method* (i.e., protocol). A client associates Subscription Objects with a particular Job by
80 performing the Create-Job-Subscriptions operation or by submitting a Job with subscription information. A
81 client associates Subscription Objects with the Printer by performing a Create-Printer-Subscriptions
82 operation. Four other operations are defined for Subscription Objects: Get-Subscriptions-Attributes, Get-
83 Subscriptions, Renew-Subscription, and Cancel-Subscription.

84			
85		Table of Contents	
86	1	Introduction.....	6
87	2	Terminology.....	6
88	3	Model and Operation.....	7
89	4	General Information.....	7
90	5	Get-Notifications operation.....	9
91	5.1	Get-Notifications Request.....	10
92	5.2	Get-Notifications Response.....	11
93	6	Subscription Template Attributes.....	14
94	6.1	Subscription Template Attribute Conformance.....	15
95	6.2	Additional Information about Subscription Template Attributes.....	15
96	6.2.1	notify-recipient-uri (uri).....	15
97	6.3	Subscription Description Attribute Conformance.....	15
98	7	Additional Printer Description Attributes.....	15
99	7.1	Printer Description Attribute Conformance.....	15
100	7.2	New Values for Existing Printer Description Attributes.....	16
101	7.2.1	notify-schemes-supported (1setOf uriScheme).....	16
102	7.2.2	operations-supported (1setOf type2 enum).....	16
103	7.3	begin-to-expire-time-interval (integer(0:MAX)).....	16
104	8	New Status Codes.....	17
105	8.1	redirection-other-site (0x300).....	17
106	9	The IPPGET URL Scheme.....	17
107	9.1	The IPPGET URL Scheme Applicability and Intended Usage.....	17
108	9.2	The IPPGET URL Scheme Associated Port.....	17
109	9.3	The IPPGET URL Scheme Associated MIME Type.....	18
110	9.4	The IPPGET URL Scheme Character Encoding.....	18
111	9.5	The IPPGET URL Scheme Syntax in ABNF.....	18
112	9.5.1	IPPGET URL Examples.....	19
113	9.5.2	IPPGET URL Comparisons.....	19
114	10	Encoding.....	20
115	11	Conformance Requirements.....	20
116	11.1	Conformance for IPP Printers.....	20
117	11.2	Conformance for IPP Clients.....	21
118	12	IANA Considerations.....	21
119	12.1	Operation Registrations.....	21
120	12.2	Additional values of existing attributes.....	22
121	12.2.1	Additional values for the “notify-schemes-supported” Printer attribute.....	22
122	12.2.2	Additional values for the “operations-supported” Printer attribute.....	22
123	12.3	Attribute Registrations.....	22

124 12.4 Status code Registrations 23

125 13 Internationalization Considerations 23

126 14 Security Considerations 23

127 15 References 23

128 16 Authors’ Addresses..... 25

129 17 Full Copyright Statement 25

130

131 **Table of Tables**

132 Table 1 – Information about the Delivery Method.....8

133 Table 2 – Attributes in Event Notification Content 13

134 Table 3 – Additional Attributes in Event Notification Content for Job Events 14

135 Table 4 – Combinations of Events and Subscribed Events for “job-impressions-completed” 14

136 Table 5 – Additional Attributes in Event Notification Content for Printer Events..... 14

137 Table 6 – Operation-id assignments 16

138 Table 7 – The "event-notification-attributes-tag" value..... 20

139

140

140 1 Introduction

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

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

148 The 'ippget' Delivery Method is a 'pull and push' Delivery Method. That is, the Printer saves Event
149 Notification for a period of time and expects the Notification Recipient to fetch the Event Notifications (the
150 pull part). The Printer continues to send Event Notifications to the Notification Recipient as Events occur (the
151 push part) if the client has selected the option to wait for additional Event Notifications.

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

154 When a Notification Recipient wants to receive Event Notifications, it performs an IPP operation called 'Get-
155 Notifications', which this document defines. This operation causes the Printer to return all Event Notifications
156 held for the Notification Recipient. If the Notification Recipient has selected the option to wait for additional
157 Event Notifications, the Printer the Printer continues to send Event Notifications to the Notification Recipient
158 as Events occur.

159 2 Terminology

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

161 Capitalized terms, such as **MUST**, **MUST NOT**, **REQUIRED**, **SHOULD**, **SHOULD NOT**, **MAY**,
162 **NEED NOT**, and **OPTIONAL**, have special meaning relating to conformance to this specification. These
163 terms are defined in [RFC2911 section 13.1 on conformance terminology, most of which is taken from RFC
164 2119 [RFC2119].

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

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

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

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

172 3 Model and Operation

173 In a Subscription Creation Operation, when the value of the “notify-recipient-uri” attribute has the scheme
174 ‘ippget’, the client is requesting that the Printer use the ‘ippget’ Delivery Method for the Event Notifications
175 associated with the new Subscription Object. The client SHOULD choose a value for the address part of the
176 “notify-recipient-uri” attribute that uniquely identifies the Notification Recipient.

177 When an Event occurs, the Printer MUST generate an Event Notification and MUST assign it the Event
178 Notification Lease Time. The Printer MUST hold an Event Notification for its assigned Event Notification
179 Lease Time. The Printer MUST assign the same Event Notification Lease Time to each Event Notification.

180 When a Notification Recipient wants to receive Event Notifications, it performs the Get-Notifications
181 operation, which causes the Printer to return all unexpired Event Notifications held for the Notification
182 Recipient. If the Notification Recipient has selected the option to wait for additional Event Notifications, the
183 response to the Get-Notifications request continues indefinitely as the Printer continues to send Event
184 Notifications in the response as Events occur. For the Get-Notification operation, the Printer sends only those
185 Event Notifications that are generated from Subscription Objects whose “notify-recipient-uri” attribute value
186 equals the value of the “notify-recipient-uri” Operation Attribute in the Get-Notifications operation.

187 If a Notification Recipient performs the Get-Notifications operation twice in quick succession, it will receive
188 nearly the same Event Notification both times because most of the Event Notifications are those that the
189 Printer saves for a few seconds after the Event occurs. There are two possible differences. Some old Event
190 Notifications may not be present in the second response because their Event Notification Leases have expired.
191 Some new Event Notifications may be present in the second response but not the first response.

192 When the Notification Recipient requests Event Notifications for per-Job Subscription Objects, the
193 Notification Recipient typically performs the Get-Notifications operation within a second of performing the
194 Subscription Creation operation. Because the Printer is likely to save Event Notifications for several seconds,
195 the Notification Recipient is unlikely to miss any Event Notifications that occur between the Subscription
196 Creation and the Get-Notifications operation.

197 4 General Information

198 If a Printer supports this Delivery Method, the following are its characteristics.

Table 1 – Information about the Delivery Method

Document Method Conformance Requirement	Delivery Method Realization
1. What is the URL scheme name for the Delivery Method?	ippget
2. Is the Delivery Method REQUIRED, RECOMMENDED or OPTIONAL for an IPP Printer to support?	RECOMMENDED
3. What transport and delivery protocols does the Printer use to deliver the Event Notification Content, i.e., what is the entire network stack?	IPP with one new operation.
4. Can several Event Notifications be combined into a Compound Event Notification?	Yes.
5. Is the Delivery Method initiated by the Notification Recipient (pull), or by the Printer (push)?	This Delivery Method is a pull and a push.
6. Is the Event Notification content Machine Consumable or Human Consumable?	Machine Consumable
7. What section in this document answers the following question? For a Machine Consumable Event Notification, what is the representation and encoding of values defined in section 9.1 of [ipp-ntfy] and the conformance requirements thereof? For a Human Consumable Event Notification, what is the representation and encoding of pieces of information defined in section 9.2 of [ipp-ntfy] and the conformance requirements thereof?	Section 5
8. What are the latency and reliability of the transport and delivery protocol?	Same as IPP and the underlying HTTP transport
9. What are the security aspects of the transport and delivery protocol, e.g., how it is handled in firewalls?	Same as IPP and the underlying HTTP transport
10. What are the content length restrictions?	None
11. What are the additional values or pieces of information that a Printer sends in an Event Notification content and the conformance requirements thereof?	None
12. What are the additional Subscription Template and/or Subscription Description attributes and the conformance requirements thereof?	None

13. What are the additional Printer Description attributes and the conformance requirements thereof?	None
--	------

200

201 5 Get-Notifications operation

202 This operation causes the Printer to return all Event Notifications held for the Notification Recipient.

203 A Printer MUST support this operation.

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

- 205 1. Whose associated Subscription Object’s “notify-recipient-uri” attribute equals the “notify-recipient-
206 uri” Operation attribute AND
- 207 2. Whose associated Subscription Object’s “notify-recipient-uri” attribute has a scheme value of ‘ippget’
208 AND
- 209 3. Whose Event Notification Lease Time has not yet expired AND
- 210 4. Where the Notification Recipient is the owner of or has read-access rights to the associated
211 Subscription Object.

212 The Printer MUST respond to this operation immediately with whatever Event Notifications it currently holds.
213 If the Notification Recipient has selected the option to wait for additional Event Notifications, the Printer
214 MUST continue to send Event Notifications as they occur until all of the associated Subscription Objects are
215 cancelled. A Subscription Object is cancelled either via the Cancel-Subscription operation or by the Printer
216 (e.g. the Subscription Object is cancelled when the associated Job completes).

217 Note, the Printer terminates the operation in the same way that it normally terminates IPP operations. For
218 example, if the Printer is sending chunked data, it can send a 0 length chunk to denote the end of the operation
219 or it can close the connection. If the Notification Recipient wishes to terminate the Get-Notifications
220 operation, it can close the connection.

221 The Printer MUST accept the request in any state (see [RFC2911] “printer-state” and “printer-state-reasons”
222 attributes) and MUST remain in the same state with the same “printer-state-reasons” values.

223 *Access Rights:* If the policy of the Printer is to allow all users to access all Event Notifications, then the Printer
224 MUST accept this operation from any user. Otherwise, the authenticated user (see [RFC2911] section 8.3)
225 performing this operation MUST either be the owner of each Subscription Object identified by the “notify-
226 recipient-uri” Operation attribute (as determined during a Subscription Creation Operation) or an operator or
227 administrator of the Printer (see [RFC2911] Sections 1 and 8.5). Otherwise, the IPP object MUST reject

228 the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-
229 authorized' [status code](#) as appropriate.

230 5.1 Get-Notifications Request

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

232 Group 1: Operation Attributes

233 Natural Language and Character Set:

234 The "attributes-charset" and "attributes-natural-language" attributes as described in [RFC2911]
235 section 3.1.4.1.

236

237 Target:

238 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
239 [RFC2911] section 3.1.5.

240

241 Requesting User Name:

242 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
243 described in [RFC2911] section 8.3.

244

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

246 The client MUST supply this attribute. The Printer object MUST support this attribute. The Printer
247 matches the value of this attribute (byte for byte with no case conversion) against the value of the
248 "notify-recipient-uri" in each Subscription Object in the Printer. If there are no matches, the IPP
249 Printer MUST return the 'client-error-not-found' status code. For each matched Subscription
250 Object, the IPP Printer MUST return all unexpired Event Notifications associated with it. The
251 Printer MUST send additional Event Notifications as Events occur if and only if the value of the
252 "notify-no-wait" attribute is 'false' or not supplied by the client (see the next attribute below).

253

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

257

258 "notify-no-wait" (boolean):

259 The client MAY supply this attribute. The Printer object MUST support this attribute. If the value
260 of this attribute is 'false', the Printer MUST send all un-expired Event Notifications (as defined in the
261 previous attribute) and it MUST continue to send responses for as long as the Subscription Objects
262 associated with the specified "notify-recipient-uri" continue to exist. If the value of this attribute is
263 'true', the Printer MUST send all un-expired Event Notifications (as defined in the previous attribute)
264 and the Printer MUST conclude the operation without waiting for any additional Events to occur. If
265 the client doesn't supply this attribute, the Printer MUST behave as if the client had supplied this
266 attribute with the value of 'false'.

267 5.2 Get-Notifications Response

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

269 Group 1: Operation Attributes

270 Status Message:

271 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY
272 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation
273 attribute as described in [RFC2911] sections 13 and 3.1.6.

274
275 The Printer can return any status codes defined in [RFC2911]. If the status code is not 'successful-
276 the Printer MUST NOT return any Event Notification Attribute groups. The following is a
277 description of the important status codes:

278
279 **successful-ok:** the response contains all Event Notification associated with the specified
280 "notify-recipient-uri". If the specified Subscription Objects have no associated Event
281 Notification, the response MUST contain zero Event Notifications.

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

284 **server-error-busy:** The Printer is too busy to accept this operation. If the "suggested-ask-
285 again-time-interval" operation attribute is present in the Operation Attributes of the
286 response, then the Notification Recipient SHOULD wait for the number of seconds
287 specified by the "suggested-ask-again-time-interval" attribute before performing this
288 operation again. If the "suggested-ask-again-time-interval" Operation Attribute is not
289 present, the Notification Recipient should use the normal network back-off algorithms for
290 determining when to perform this operation again.

291 **redirection-other-site:** The Printer does not handle this operation and requests the
292 Notification Recipient to perform the operation with the uri specified by the "notify-ippget-
293 redirect" Operation Attribute in the response.

294
295 Natural Language and Character Set:

296 The "attributes-charset" and "attributes-natural-language" attributes as described in [RFC2911]
297 section 3.1.4.2.

298
299 The Printer MUST use the values of "notify-charset" and "notify-natural-language", respectively,
300 from one Subscription Object associated with the Event Notifications in this response.

301
302 Normally, there is only one matched Subscription Object, or the value of the "notify-charset" and
303 "notify-natural-language" attributes is the same in all Subscription Objects. If not, the Printer MUST
304 pick one Subscription Object from which to obtain the value of these attributes. The algorithm for
305 picking the Subscription Object is implementation dependent. The choice of natural language is not
306 critical because 'text' and 'name' values can override the "attributes-natural-language" Operation

307 attribute. The Printer’s choice of charset is critical because a bad choice may leave it unable to send
308 some ‘text’ and ‘name’ values accurately.

309

310 “printer-up-time” (integer(0:MAX)):

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

315

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

317 The value of this attribute is the number of seconds that the Notification Recipient SHOULD wait
318 before trying this operation again when

319

a) the Printer returns the ‘server-error-busy’ status code OR

320

b) the Printer returns the ‘successful-ok’ status code and the client supplied the “notify-no-
321 wait” attribute with a value of ‘true’.

322

This value is intended to help the client be a good network citizen.

323

324 “notify-ippget-redirect” (uri):

325 The value of this attribute is uri that the Notification Recipient MUST use for the Get-Notifications
326 operation. This attribute is present in the Operation Attributes if and only if the status code has the
327 value ‘redirection-other-site’.

328

329

Group 2: Unsupported Attributes

330

See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes.

331

332

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

334

335

Group 3 through N: Event Notification Attributes

336

The Printer responds with one Event Notification Attributes Group per matched Event Notification.
337 The initial matched Event Notifications are all un-expired Event Notification associated with the
338 matched Subscription Objects. If the Notification Recipient has selected the option to wait for
339 additional Event Notifications, the Printer the subsequent Event Notifications in the response are
340 Event Notifications associated with the matched Subscription Objects as the corresponding Event
341 occurs.

342

343

From the Notification Recipient’s view, the response appears as an initial burst of data, which
344 includes the Operation Attributes Group and one Event Notification Attributes Groups per Event
345 Notification that the Printer is holding. After the initial burst of data, if the Notification Recipient has
346 selected the option to wait for additional Event Notifications, the Notification Recipient receives
347 occasional Event Notification Attribute Groups. Proxy servers may delay some Event Notifications

348 or cause time-outs to occur. The client MUST be prepared to perform the Get-Notifications
349 operation again when time-outs occur.

350
351 Each Event Notification Group MUST start with an ‘event-notification-attributes-tag’ (see the
352 section “Encodings of Additional Attribute Tags” in [ipp-ntfy]).

353
354 Each attribute is encoded using the IPP rules for encoding attributes [RFC2910] and may be
355 encoded in any order. Note: the Get-Jobs response in [RFC2911] acts as a model for encoding
356 multiple groups of attributes.

357
358 Each Event Notification Group MUST contain all of attributes specified in section 9.1 (“Content of
359 Machine Consumable Event Notifications”) of [ipp-ntfy] with exceptions denoted by asterisks in the
360 tables below.

361
362 The tables below are copies of the tables in section 9.1 (“Content of Machine Consumable Event
363 Notifications”) of [ipp-ntfy] except that each cell in the “Sends” column is a “MUST”.

364
365 For an Event Notification for all Events, the Printer includes the following attributes shown in Table
366 2.

367 **Table 2 – 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

368
369 * The Printer MUST send the “printer-current-time” attribute if and only if it supports the “printer-
370 current-time” attribute on the Printer object.

371
372 ** If the associated Subscription Object does not contain a “notify-user-data” attribute, the Printer
373 MUST send an octet-string of length 0.

374
 375 *** If the “notify-attributes” attribute is present on the Subscription Object, the Printer MUST send
 376 all attributes specified by the “notify-attributes” attribute. Note: if the Printer doesn’t support the
 377 “notify-attributes” attribute, it is not present on the associated Subscription Object.

378
 379 For Event Notifications for Job Events, the Printer includes the following additional attributes shown
 380 in Table 3.

381 **Table 3 – 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

382
 383 * The Printer MUST send the “job-impressions-completed” attribute in an Event Notification only
 384 for the combinations of Events and Subscribed Events shown in Table 4.
 385

386 **Table 4 – 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’

387
 388
 389 For Event Notification for Printer Events, the Printer includes the following additional attributes
 390 shown in Table 5.

391 **Table 5 – 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

392 6 Subscription Template Attributes

393 This section defines the Subscription object conformance requirements for Printers.

394 6.1 Subscription Template Attribute Conformance

395 The 'ippget' Delivery Method has the same conformance requirements for Subscription Template attributes as
396 defined in [ipp-ntfy]. The 'ippget' Delivery Method does not define any addition Subscription Template
397 attributes.

398 6.2 Additional Information about Subscription Template Attributes

399 This section defines additional information about Subscription Template attributes defined in [ipp-ntfy].

400 6.2.1 notify-recipient-uri (uri)

401 This section describes the syntax of the value of this attribute for the 'ippget' Delivery Method. The syntax for
402 values of this attribute for other Delivery Method is defined in other Delivery Method Documents.

403 In order to support the 'ippget' Delivery Method and Protocol, the Printer MUST support the following
404 syntax:

405 The 'ippget://' URI scheme. The remainder of the URI indicates something unique about the Notification
406 Recipient, such as its host name and-or host address (and optional path) that the Printer uses to match the
407 "notify-recipient-uri" Operation attribute supplied in the Get-Notifications request.

408 6.3 Subscription Description Attribute Conformance

409 The 'ippget' Delivery Method has the same conformance requirements for Subscription Description attributes
410 as defined in [ipp-ntfy]. The 'ippget' Delivery Method does not define any addition Subscription Description
411 attributes.

412 7 Additional Printer Description Attributes

413 This section defines the Printer Description Attributes conformance requirements for Printers.

414 7.1 Printer Description Attribute Conformance

415 The 'ippget' Delivery Method has the same conformance requirements for Printer Description attributes as
416 defined in [ipp-ntfy]. The 'ippget' Delivery Method does not define any addition Printer Description
417 attributes.

418 7.2 New Values for Existing Printer Description Attributes

419 This section defines additional values for existing Printer Description attributes.

420 7.2.1 notify-schemes-supported (1setOf uriScheme)

421 The following [value for the "notify-schemes-supported" value-attribute](#) is added in order to support the new
422 Delivery Method defined in this document:

423 'ippget' - The IPP Notification Delivery Method defined in this document.

424 7.2.2 operations-supported (1setOf type2 enum)

425 Table 6 lists the "operation-id" value [added-defined](#) in order to support the new [Get-Notifications](#) operation
426 defined in this document.

427 **Table 6 – Operation-id assignments**

Value	Operation Name
0x001C	Get-Notifications

428

429 7.3 begin-to-expire-time-interval (integer(0:MAX))

430 This [Printer Description](#) attribute specifies the number of seconds that a Printer keeps an Event Notification
431 that is associated with [this the 'ippget' Delivery Method](#).

432 The Printer MUST support this attribute if it supports [this the 'ippget' Delivery Method](#).

433 The value of this attribute is the minimum number of seconds that MUST elapse between the time the Printer
434 creates an Event Notification object for [this the 'ippget' Delivery Method](#) and the time the Printer discards the
435 same Event Notification.

436 For example, assume the following:

- 437 1. a client performs a Job Creation operation that creates a Subscription Object associated with this
438 Delivery Method, AND
- 439 2. an Event associated with the new Job occurs immediately after the Subscription Object is created,
440 AND
- 441 3. the same client or some other client performs a Get-Notifications operation N seconds after the Job
442 Creation operation.

443 Then, if N is less than the value of this attribute, the client performing the Get-Notifications operations can
444 expect not miss any Event-Notifications, barring some unforeseen lack of memory space in the Printer.

445 **8 New Status Codes**

446 The following status codes are defined as extensions for this Delivery Method and are returned as the status
447 code of the Get-Notifications operation.

448 **8.1 redirection-other-site (0x300)**

449 This status code means that the Printer doesn't perform that Get-Notifications operation and that the "notify-
450 ippget-redirect" Operation Attribute in the response contains the uri that the Notification Recipient MUST use
451 for performing the Get-Notifications operation.

452 **9 The IPPGET URL Scheme**

453 This section defines the 'ippget' URL and the conformance requirements for using it.

454 **9.1 The IPPGET URL Scheme Applicability and Intended Usage**

455 This section is intended for use in registering the 'ippget' URL scheme with IANA and fully conforms to the
456 requirements in [RFC2717]. This document defines the 'ippget' URL (Uniform Resource Locator) scheme
457 for specifying a unique identifier for an IPP Client which implements the IPP Get-Notifications operation
458 specified in this document (see section 5).

459 The intended usage of the 'ippget' URL scheme is COMMON.

460 **9.2 The IPPGET URL Scheme Associated Port**

461 None.

462 An 'ippget' URL behaves as a unique identifier for IPP Clients and is NOT used to initiate any over-the-wire
463 protocol associations.

464 See: IANA Port Numbers Registry [IANA-PORTREG].

465 9.3 The IPPGET URL Scheme Associated MIME Type

466 All IPP Get-Notifications operations (requests and responses) MUST be conveyed in an 'application/ipp'
467 MIME media type as registered in [IANA-MIMEREG]. An 'ippget' URL MUST uniquely identify an IPP
468 Client that support this 'application/ipp' MIME media type.

469 See: IANA MIME Media Types Registry [IANA-MIMEREG].

470 9.4 The IPPGET URL Scheme Character Encoding

471 The 'ippget' URL scheme defined in this document is based on the ABNF for the URI Generic Syntax
472 [RFC2396] and further updated by [RFC2732] and [RFC2373] (for IPv6 addresses in URLs). The 'ippget'
473 URL scheme is case-insensitive in the host name or host address part; however, the path part is case-
474 sensitive, as in [RFC2396]. Code points outside [US-ASCII] MUST be hex escaped by the mechanism
475 specified in [RFC2396].

476 9.5 The IPPGET URL Scheme Syntax in ABNF

477 This document is intended for use in registering the 'ippget' URL scheme with IANA and fully conforms to the
478 requirements in [RFC2717]. This document defines the 'ippget' URL (Uniform Resource Locator) scheme
479 for specifying a unique identifier for an IPP Client which implements IPP 'Get-Notifications' operation
480 specified in this document.

481 The intended usage of the 'ippget' URL scheme is COMMON.

482 The IPP protocol places a limit of 1023 octets (NOT characters) on the length of a URI (see section 4.1.5
483 'uri' in [RFC2911]). An IPP Printer MUST return the 'client-error-request-value-too-long' status code (see
484 section 13.1.4.10 in [RFC2911]) when a URI received in a request is too long.

485 *Note: IPP Clients and IPP Printers ought to be cautious about depending on URI lengths above*
486 *255 bytes, because some older client or proxy implementations might not properly support these*
487 *lengths.*

488 An 'ippget' URL MUST be represented in absolute form. Absolute URLs always begin with a scheme name
489 followed by a colon. For definitive information on URL syntax and semantics, see "Uniform Resource
490 Identifiers (URI): Generic Syntax and Semantics" [RFC2396]. This specification adopts the definitions of
491 "authority", "abs_path", "query", "reg_name", "server", "userinfo", and "hostport" from [RFC2396], as
492 updated by [RFC2732] and [RFC2373] (for IPv6 addresses in URLs).

493 The 'ippget' URL scheme syntax in ABNF is as follows:

494 ippget_URL = "ippget:" "//" authority [abs_path ["?" query]]
495 authority = server | reg_name

```

496   reg_name   = 1*( unreserved | escaped | "$" | ", " |
497                   ";" | ":" | "@" | "&" | "=" | "+" )
498   server     = [ [ userinfo "@" ] hostport ]
499   userinfo   = *( unreserved | escaped |
500                   ";" | ":" | "&" | "=" | "+" | "$" | ", " )
501   hostport   = host [ ":" port ]
502   abs_path   = "/" path_segments
503

```

If the port is empty or not given, then no port is assumed. The semantics are that the 'ippget' URL is a unique identifier for an IPP Client that will retrieve IPP event notifications via the IPP Get-Notifications operation.

Note: The use of IP addresses in URLs SHOULD be avoided whenever possible (see [RFC1900]).

9.5.1 IPPGET URL Examples

The following are examples of valid 'ippget' URLs for IPP Clients (using DNS host names):

```

509   ippget://abc.com
510   ippget://abc.com/listener
511   ippget://bob@abc.com/listener/1232
512

```

Note: The use of IP addresses in URLs SHOULD be avoided whenever possible (see [RFC1900]).

The choice of 'userinfo@hostport' versus the simpler 'hostport' production in an 'ippget' URL may be influenced by the intended usage.

If a given IPP Client creates an IPP Subscription object for event notifications intended for retrieval by the same IPP Client, then the simple 'hostport' production may be most appropriate.

On the other hand, if a given IPP Client creates an IPP Subscription object for event notifications intended for retrieval by a *different* IPP Client, then the 'userinfo@hostport' production (using, for example, the right-hand side of a 'mailto:' URL, see [RFC2368]) may be most appropriate.

9.5.2 IPPGET URL Comparisons

When comparing two 'ippget' URLs to decide if they match or not, an IPP Client or IPP Printer SHOULD use a case-sensitive octet-by-octet comparison of the entire URLs, with these exceptions:

- Comparisons of host names MUST be case-insensitive;
- Comparisons of scheme names MUST be case-insensitive;
- An empty 'abs_path' is equivalent to an 'abs_path' of "?".

527 Characters other than those in the “reserved” and “unsafe” sets (see [RFC2396] and [RFC2732]) are
528 equivalent to their “%” HEX HEX” encoding.

529 For example, the following three URIs are equivalent:

530 ippget://abc.com/~smith/listener

531 ippget://ABC.com/%7Esmith/listener

532 ippget://ABC.com:/%7esmith/listener

533 10 Encoding

534 This notification delivery method uses the IPP transport and encoding [RFC2910] for the Get-Notifications
535 operation with one extension allocated in [ipp-ntfy]:

536 Table 7 – The "event-notification-attributes-tag" value

<u>Tag Value (Hex)</u>	<u>Meaning</u>
<u>0x07</u>	<u>“event-notification-attributes-tag”</u>

537 ~~notification-attributes-tag = %x07 ; tag of 7~~

539 11 Conformance Requirements

540 11.1 Conformance for IPP Printers

541 IPP Printers that conform to this specification:

542 ~~If the Printer supports the ‘ippget’ Delivery Method, the Printer MUST:~~

- 543 1. MUST meet the conformance requirements defined in [ipp-ntfy]~~;~~
- 544 2. MUST support the Get-Notifications operation defined in section 5~~;~~
- 545 3. MUST support the Subscription object attributes as defined in section 6~~;~~
- 546 4. MUST support the additional values for IPP/1.1 Printer Description attributes defined in section 7.2~~;~~
- 547 5. MUST support the “begin-to-expire-time-interval” Printer Description attribute defined in section 7.3~~;~~
- 548 6. MUST support the “redirection-other-site” status code defined 8.1~~;~~

- 549 7. SHOULD reject received 'ippget' URLs in 'application/ipp' request bodies (e.g., in the "notify-
550 recipient-uri" attribute in a Get-Notifications request) that do not conform to the ABNF for 'ippget'
551 URLs specified in section 9.5 of this document;
- 552 8. MUST listen for the IPP Get-Notifications operation requests on IANA-assigned well-known port
553 631, unless explicitly configured by system administrators or site policies;
- 554 9. SHOULD NOT listen for IPP Get-Notifications operation requests on any other port, unless explicitly
555 configured by system administrators or site policies.

556 11.2 Conformance for IPP Clients

557 IPP Clients that conform to this specification:

- 558 1. MUST create unambiguously unique 'ippget' URLs in all cases;
- 559 2. MUST send 'ippget' URLs (e.g., in the "notify-recipient-uri" attribute in a Get-Notifications request)
560 that conform to the ABNF specified in section 9.5 of this document;
- 561 3. MUST send IPP Get-Notifications operation requests via the port specified in the associated 'ipp'
562 URL (if present) or otherwise via IANA assigned well-known port 631;
- 563 4. MUST convert the associated 'ipp' URLs to their corresponding 'http' URL forms according to the
564 rules in section 5 "IPP URL Scheme" in [RFC2910].

565 Note: The use of ambiguous 'ippget' URLs is NOT an optional feature for IPP Clients; it is a non-conformant
566 implementation error.

567 **12 IANA Considerations**

568 IANA is requested to register the 'ippget' URL scheme as defined in section 9. ~~The 'ippget' URL scheme for~~
569 ~~the 'ippget' Delivery Method will be registered with IANA~~ according to the procedures of [RFC2717].

570 The rest of this section contains the exact information for additional IPP entities for IANA to add to the IPP
571 Registries according to the procedures defined in RFC 2911 [RFC2911] section 6.

572 *Note to RFC Editors: Replace RFC NNNN below with the RFC number for this document, so that*
573 *it accurately reflects the content of the information for the IANA Registry.*

574 12.1 Operation Registrations

575 The operations defined in this document will be published by IANA according to the procedures in RFC 2911
576 [RFC2911] section 6.4 with the following path:

577 <ftp.isi.edu/iana/assignments/ipp/operations/>

578 The registry entry will contain the following information:

579	<u>Operations:</u>	<u>Ref.</u>	<u>Section:</u>
580	Get-Notifications operation	RFC NNNN	5
581			

582 **12.2 Additional values of existing attributes**

583 **12.2.1 Additional values for the “notify-schemes-supported” Printer attribute**

584 The “notify-schemes-supported” ‘uriScheme’ attribute value defined in this document will be published by
 585 IANA according to the procedures in RFC 2911 [RFC2911] section 6.1 with the following path:

586 <ftp.isi.edu/iana/assignments/ipp/attribute-values/notify-schemes-supported/>

587 The registry entry will contain the following information:

588		<u>Ref.</u>	<u>Section:</u>
589	ippget	RFC NNNN	7.2.1

590 **12.2.2 Additional values for the “operations-supported” Printer attribute**

591 The “operations-supported” type2 enum attribute value defined in this document will be published by IANA
 592 according to the procedures in RFC 2911 [RFC2911] section 6.1 with the following path:

593 <ftp.isi.edu/iana/assignments/ipp/attribute-values/operations-supported/>

594 The registry entry will contain the following information:

595		<u>Value</u>	<u>Ref.</u>	<u>Section:</u>
596	Get-Notifications	0x001C	RFC NNNN	7.2.2

597 **12.3 Attribute Registrations**

598 The attributes defined in this document will be published by IANA according to the procedures in RFC 2911
 599 [RFC2911] section 6.2 with the following path:

600 <ftp.isi.edu/iana/assignments/ipp/attributes/>

601 The registry entry will contain the following information:

602	<u>Printer Description attributes:</u>	<u>Ref.</u>	<u>Section:</u>
603	<u>begin-to-expire-time-interval (integer(0:MAX))</u>	RFC NNNN	7.3

604 12.4 Status code Registrations

605 The status codes defined in this document will be published by IANA according to the procedures in RFC
 606 2911 [RFC2911] section 6.6 with the following path:

607 <ftp.isi.edu/iana/assignments/ipp/status-codes/>

608 The registry entry will contain the following information:

609 <u>Status codes:</u>	<u>Ref.</u>	<u>Section:</u>
610 <u>redirection-other-site (0x300)</u>	<u>RFC NNNN</u>	<u>8.1</u>
611		

612 **13 Internationalization Considerations**

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

614 In addition, when the client receives the Get-Notifications response, it is expected to localize the attributes that
 615 have the 'keyword' attribute syntax according to the charset and natural language requested in the Get-
 616 Notifications request.

617 **14 Security Considerations**

618 The IPP Model and Semantics document [RFC2911] discusses high-level security requirements (Client
 619 Authentication, Server Authentication and Operation Privacy). Client Authentication is the mechanism by
 620 which the client proves its identity to the server in a secure manner. Server Authentication is the mechanism by
 621 which the server proves its identity to the client in a secure manner. Operation Privacy is defined as a
 622 mechanism for protecting operations from eavesdropping.

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

628 **15 References**

629 [ipp-iig]
 630 Hastings, T., Manros, C., Kugler, K, Holst H., Zehler, P., "Internet Printing Protocol/1.1: draft-ietf-ipp-
 631 implementers-guide-v11-02.txt, work in progress, January 25, 2001

- 632 [ipp-ntfy]
633 R. Herriot, Hastings, T., Isaacson, S., Martin, J., deBry, R., Shepherd, M., Bergman, R., "Internet Printing
634 Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-064.txt>, ~~June 30,~~
635 ~~2000~~[February 24, 2001](#).
- 636 [\[RFC1900\]](#)
637 [B. Carpenter, Y. Rekhter. Renumbering Needs Work, RFC 1900, February 1996.](#)
- 638 [~~fe~~[RFC2026](#)]
639 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 640 [\[RFC2119\]](#)
641 [S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997](#)
- 642 [\[RFC2368\]](#)
643 [P. Hoffman, L. Masinter, J. Zawinski. The "mailto" URL Scheme, RFC 2368, July 1998.](#)
- 644 [\[RFC2373\]](#)
645 [R. Hinden, S. Deering. IP Version 6 Addressing Architecture, RFC 2373, July 1998.](#)
- 646 [\[RFC2396\]](#)
647 [Berners-Lee, T. et al. Uniform Resource Identifiers \(URI\): Generic Syntax, RFC 2396, August 1998](#)
- 648 [\[RFC2567\]](#)
649 [Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.](#)
- 650 [\[RFC2568\]](#)
651 [Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol", RFC
652 2568, April 1999.](#)
- 653 [\[RFC2569\]](#)
654 [Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC 2569,
655 \[April 1999.\]\(#\)](#)
- 656 [RFC2616]
657 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer
658 Protocol - HTTP/1.1", RFC 2616, June 1999.
- 659 [\[RFC2717\]](#)
660 [R. Petke and I. King, "Registration Procedures for URL Scheme Names", RFC 2717, November 1999.](#)
- 661 [\[RFC2732\]](#)
662 [R. Hinden, B. Carpenter, L. Masinter. Format for Literal IPv6 Addresses in URL's, RFC 2732,
663 \[December 1999.\]\(#\)](#)

- 664 [RFC2910]
665 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and Transport",
666 RFC 2910, September 2000.
- 667 [RFC2911]
668 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.1: Model and
669 Semantics", RFC 2911, September 2000.

670 **16 Authors' Addresses**

671
672 Robert Herriot
673 Xerox Corp.
674 3400 Hill View Ave, Building 1
675 Palo Alto, CA 94304
676
677 Phone: 650-813-7696
678 Fax: 650-813-6860
679 e-mail: robert.herriot@pahv.xerox.com
680

681 Carl Kugler
682 IBM
683 P.O. Box 1900
684 Boulder, CO 80301-9191
685
686 Phone:
687 Fax:
688 e-mail: kugler@us.ibm.com
689

690 Harry Lewis
691 IBM
692 P.O. Box 1900
693 Boulder, CO 80301-9191
694
695 Phone: 303-924-5337
696 FAX:
697 e-mail: harryl@us.ibm.com
698

699 **17 Full Copyright Statement**

700 Copyright (C) The Internet Society (2001~~0~~). All Rights Reserved.

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

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

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

716 **Acknowledgement**

717
718 Funding for the RFC Editor function is currently provided by the Internet Society.