1   Internet Printing Protocol WG   R. Herriot
2   INTERNET-DRAFT   consultant
3   <draft-ietf-ipp-notify-get-09 8.txt>   T. Hastings
4   Updates: RFC 2911 and [ipp-ntfy]   Xerox Corp.
5   [Target category: standards track]   H. Lewis
6   Expires: April 10 August 17, 2003   IBM Corp.
7   October 10, 2002 February 17, 2003

9   Internet Printing Protocol (IPP):
10   **The 'ippget' Delivery Method for Event Notifications**

Status of this Memo:

Abstract

This document describes an extension to the Internet Printing Protocol/1.1: Model and Semantics (RFC 2911, RFC 2910). This document specifies the 'ippget' Pull Delivery Method for use with the "Internet Printing Protocol (IPP): Event Notifications and Subscriptions" specification (ipp-ntfy). This IPPGET Delivery Method is REQUIRED for all clients and Printers that support ipp-ntfy. The Notification Recipient, acting as a client, fetches (pulls) Event Notifications using the Get-Notifications operation defined in this document.

31
32 **Table of Contents**

82   **Table of Tables**

92   1 Introduction

93   This document describes an extension to the Internet Printing Protocol/1.1: Model and Semantics
94   [RFC 2911], [RFC 2910]. This document specifies the 'ippget' Pull Delivery Method for use with the
95   "Internet Printing Protocol (IPP): Event Notifications and Subscriptions" specification [ipp-ntfy]. This
96   IPPGET Delivery Method is REQUIRED for all clients and Printers that support [ipp-ntfy]. The
97   Notification Recipient, acting as a client, fetches (pulls) Event Notifications using the Get-
98   Notifications operation defined in this document. For a description of the base IPP documents, see
99   section 21 of this document. For a description of the IPP Event Notification Model, see [ipp-ntfy].

100   With this Pull Delivery Method, when an Event occurs, the Printer saves the Event Notification for a
101   period of time called the Event Life. The Notification Recipient fetches (pulls) the Event Notifications
102   using the Get-Notifications operation. This operation causes the Printer to return all Event
103   Notifications held for the specified Subscription object(s). If the Notification Recipient has selected
104   the **Event Wait Mode** option to wait for additional Event Notifications, the Printer MAY continue to
105   return Event Notifications to the Notification Recipient as asynchronous Get-Notification responses as
106   Events occur using the transaction originated by the Notification Recipient.

107   The Notification Recipient can terminate **Event Wait Mode** (without closing the connection) by
108   supplying the "notify-wait" (boolean) attribute with a 'false' value in a subsequent Get-Notifications
109   request. Similarly, the Printer can terminate **Event Wait Mode** (without closing the connection) by
110   returning the "notify-get-interval" (integer) operation attribute in a Get-Notifications response which
111   tells the Notification Recipient how long to wait before trying again.

112   2 Terminology

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

114   2.1 Conformance Terminology

115   Capitalized terms, such as **MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY,**
116   **NEED NOT,** and **OPTIONAL**, have special meaning relating to conformance as defined in RFC 2119
117   [RFC2119] and [RFC2911] section 12.1. If an implementation supports the extension defined in this
118   document, then these terms apply; otherwise, they do not. These terms define conformance to *this*
119   *document only*; they do not affect conformance to other documents, unless explicitly stated otherwise.

120   2.2 Other terminology

121   This document uses the same terminology as [RFC2911], such as "**client**", "**Printer**", "**Job**",
122   "**attribute**", "**attribute value**", "**keyword**", "**operation**", "**request**", "**response**", and "**support**" with
123   the same meanings. This document also uses terminology defined in [ipp-ntfy], such as "**Subscription**
124   **(object)**", "**Notification Recipient**", "**Event**", "**Event Notification**", "**Compound Event**
125   **Notification**," "**Event Life**", and "**Event Notification Attribute Group**" with the same meanings. In
126   addition, this document defines the following terms for use in this document:

127   **Event Wait Mode:** The mode requested by a Notification Recipient client in its Get-Notifications
128   Request and granted by a Printer to keep the connection open while the Printer sends
129   subsequent Event Notifications to the Notification Recipient as they occur as additional Get-
130   Notification operation responses.

131 # 3 Model and Operation

132 In a Subscription Creation Operation, when the "notify-pull-method" attribute is present and has the
133 'ippget' keyword value, the client is requesting that the Printer use the 'ippget' Pull Delivery Method
134 for the Event Notifications associated with the new Subscription Object.

135 When an Event occurs, the Printer MUST generate an Event Notification and MUST assign it the
136 Event Life. The Printer MUST hold an Event Notification for its assigned Event Life.

137 When a Notification Recipient wants to receive Event Notifications for a Subscription object, it
138 performs the Get-Notifications operation supplying the Subscription object's subscription-id, which
139 causes the Printer to return all un-expired Event Notifications held for that Subscription object. If the
140 Notification Recipient has selected the **Event Wait Mode** option to wait for additional Event
141 Notifications, the response to the Get-Notifications request continues indefinitely as the Printer
142 continues to send Event Notifications in the response as Events occur for that Subscription object.

143 When the Notification Recipient requests Event Notifications for per-Job Subscription Objects, the
144 Notification Recipient typically performs the Get-Notifications operation within a second of
145 performing the Subscription Creation operation. Because the Printer MUST save Event Notifications
146 for at least 15 seconds (see section 8.1), the Notification Recipient is unlikely to miss any Event
147 Notifications that occur between the Subscription Creation and the Get-Notifications operation.

148 The 'ippget' Delivery Method is designed primarily for (1) a client that wants to get Events (from the
149 job's per-Job Subscription object) for a job that it has submitted and (2) for a privileged client that
150 wants to get all job or printer Events from a per-Printer Subscription object.

151   # 4 General Information

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

153   **Table 1 – Information about the Delivery Method**

Document Method Conformance Requirement	Delivery Method Realization
1. What is the URL scheme name for the Push Delivery Method or the keyword method name for the Pull Delivery Method?	'ippget' keyword method name
2. Is the Delivery Method REQUIRED, RECOMMENDED or OPTIONAL for an IPP Printer to support?	REQUIRED
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 method with aspects of a push method, though the Printer does not initiate the operation.
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 and in the same direction, so no new firewall considerations.
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? | "ipp-event-life" (integer (15: MAX)) |

154

5 Get-Notifications operation

156 This operation is issued by a client acting in the role of a Notification Recipient requesting the Printer
157 to return all Event Notifications held for the identified Subscription object(s).

158 A Printer MUST support this operation, MUST accept the request in any state (see [RFC2911]
159 "printer-state" and "printer-state-reasons" attributes), and MUST remain in the same state with the
160 same "printer-state-reasons" values.

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

162   1. Whose associated Subscription Object's "notify-subscription-id" Subscription Description
163   attribute equals one of the values of the "notify-subscription-ids" (1setOf integer(1:MAX))
164   operation attribute AND

165   2. Whose associated Subscription Object's contains the "notify-pull-method" attribute and it has
166   the 'ippget' keyword value AND

167   3. Whose "notify-sequence-number" is equal to or greater than the corresponding value of the
168   "notify-sequence-numbers (1setOf integer(1:MAX)) operation attribute, if supplied AND

169   4. Whose Event Life has not yet expired AND

170   5. Where the Notification Recipient client has read-access rights to the identified Subscription
171   Object (see *Access Rights* paragraph below).

172 The Notification Recipient client MUST either: (a) request **Event Wait Mode** by supplying the
173 "notify-wait" operation attribute with a 'true' value or (b) suppress Event Wait Mode by omitting the
174 "notify-wait" operation attribute or by supplying it with a 'false' value. In order to terminate Event
175 Wait Mode subsequently, the Notification Recipient client MUST close the connection. In order to
176 terminate **Event Wait Mode**, the Printer MUST either (a) return the "notify-get-interval" operation
177 attribute in a Get-Notifications response (RECOMMENDED behavior) or (b) close the connection.
178 The "notify-get-interval" operation attributes tells the Notification Recipient how long to wait before
179 trying a subsequent Get-Notifications request.

180 *Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation MUST
181 be (1) the owner of each Subscription Object identified by the "notify-subscription-ids" operation
182 attribute (see section 5.1.1), (2) an operator or administrator of the Printer (see [RFC2911] Sections 1
183 and 8.5), or (3) be otherwise authorized by the Printer's administrator-configured security policy to
184 request Event Notifications from the target Subscription Object(s). Otherwise, the IPP Printer MUST
185 reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-
186 error-not-authorized' status code as appropriate. Furthermore, the Printer's security policy MAY limit

187 the attributes returned by the Get-Notifications operation, in a manner similar to the Get-Job-Attributes
188 operation (see [RFC2911] end of section 3.3.4.2).

189 **5.1 Get-Notifications Request**

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

191 Group 1: Operation Attributes

192 Natural Language and Character Set:
193 The "attributes-charset" and "attributes-natural-language" attributes as described in
194 [RFC2911] section 3.1.4.1.
195
196 Target:
197 The "printer-uri" (uri) operation attribute which is the target for this operation as described in
198 [RFC2911] section 3.1.5.
199
200 Requesting User Name:
201 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
202 described in [RFC2911] section 8.3.
203

204 **5.1.1 notify-subscription-ids (1setOf integer(1:MAX))**

205 This attribute identifies one or more Subscription objects for which Events are requested. The
206 client MUST supply this attribute with at least one value. The Printer object MUST support
207 this attribute with multiple values.
208
209 If no Subscription Object exists with the supplied identifier or the identified Subscription
210 Object does not contain the "notify-pull-method" attribute with the 'ippget' keyword value,
211 the Printer MUST return the 'client-error-not-found' status code.
212
213 Note: The name of both the "notify-subscription-ids" and "notify-sequence-
214 numbers" end in 's', since they are multi-valued. However, there are other
215 occurrences of these attribute names without the 's' that are single valued.

216 **5.1.2 notify-sequence-numbers (1setOf integer(1:MAX))**

217 This attribute specifies one or more lowest Event Notification sequence number values for the
218 Subscription objects identified by the corresponding values of the "notify-subscription-ids"
219 operation attribute. The Notification Recipient SHOULD supply this attribute and the
220 number of values SHOULD be the same as the number of values of the "notify-subscriptions-
221 ids" attribute. The Printer MUST support this attribute with multiple values.
222
223 The Printer MUST NOT return Notification Events with lower sequence numbers for the
224 corresponding Subscription object. Therefore, by supplying the proper values for this
225 attribute the Notification Recipient can prevent getting the same Event Notifications from a

226   Subscription object that were returned on a previous Get-Notifications request. The
227   Notification Recipient SHOULD remember the highest "notify-sequence-number" value
228   returned for each Subscription object requested and SHOULD pass that value for each
229   requested Subscription object on the next Get-Notifications request.
230
231   If the Notification Recipient supplies fewer values for this attribute (including omitting this
232   attribute) than for the "notify-subscription-ids" operation attribute, the Printer assumes a '1'
233   value for each missing value. A value of '1' causes the Printer to return any un-expired Event
234   Notification for that Subscription object, since '1' is the lowest possible sequence number. If
235   the Notification Recipient supplies more values for this attribute than the number of values
236   for the "notify-subscription-ids" operation attribute, the Printer ignores the extra values.
237
238   Note: If a Notification Recipient performs two consecutive Get-Notifications operations with
239   the same value for "notify-sequence-number" (or omits the attribute), the time stamp of the
240   first Event Notification in the second Get-Notifications Response may be less than the time
241   stamp of the last Event Notification in the first Get-Notification Response. This happens
242   because the Printer sends all unexpired Event Notification with a sequence number equal or
243   higher according to the ordering specified in [ipp-ntfy] and some Event Notifications from the
244   first Get-Notifications operation may not have expired by the time the second Get-
245   Notifications operation occurs.
246

247   ### 5.1.3 notify-wait (boolean)

248   This value indicates whether or not the Notification Recipient wants **Event Wait Mode**. The
249   client MAY supply this attribute. The Printer object MUST support both values of this
250   attribute.
251
252   If the client supplies the 'false' value or omits this attribute, the client is not requesting **Event**
253   **Wait Mode**. If the value is 'true', the client is requesting **Event Wait Mode**. See the
254   beginning of section 5.2 for the rules for **Event Wait Mode**.

255   ## 5.2 Get-Notifications Response

256   The Printer has the following options for responding to a Get-Notifications Request:

257   1.   The Printer can reject the request and return the 'server-error-busy' status code, if the Printer is
258   too busy to accept this operation at this time. In this case, the Printer MUST return the "get-
259   notify-interval" operation attribute to indicate when the client SHOULD try again.

260   2.   If the Notification Recipient did not request **Event Wait Mode** ("notify-wait-mode" = 'false'
261   or omitted), the Printer MUST return immediately whatever Event Notifications it currently
262   holds in the requested Subscription object(s) and MUST return the "notify-get-interval"
263   operation attribute with number of seconds from now at which the Notification Recipient
264   SHOULD repeat the Get-Notifications Request to get future Event Notifications.

265   3. If the Notification Recipient requested **Event Wait Mode** ("notify-wait-mode" = 'true'), the
266   Printer MUST return immediately whatever Event Notifications it currently holds in the
267   requested Subscription object(s) and MUST continue to return Event Notifications as they
268   occur until all of the requested Subscription Objects are canceled. A Subscription Object is
269   canceled either via the Cancel-Subscription operation or by the Printer (e.g., the Subscription
270   Object is canceled when the associated Job completes and is no longer in the Job Retention or
271   Job History phase - see the "ippget-event-life (integer(15:MAX))" attribute discussion in
272   section 8.1).

273   However, the Printer MAY decide to terminate **Event Wait Mode** at any time, including in the
274   first response. In this case the Printer MUST return the "notify-get-interval" operation
275   attribute. This attribute indicates that the Printer wishes to leave **Event Wait Mode** and the
276   number of seconds in the future that the Notification Recipient SHOULD try the Get-
277   Notifications operation again. The Notification Recipient MUST accept this response and
278   MUST disconnect. If the Notification Recipient does not disconnect, the Printer SHOULD do
279   so.

280 From the Notification Recipient's view, the response appears as an initial burst of data, which includes
281 the Operation Attributes Group and one Event Notification Attributes Group per Event Notification
282 that the Printer is holding. After the initial burst of data, if the Notification Recipient has selected the
283 **Event Wait Mode** option to wait for additional Event Notifications, the Notification Recipient
284 receives occasional Event Notification Attribute Groups. Proxy servers may delay some Event
285 Notifications or cause time-outs to occur. The client MUST be prepared to perform the Get-
286 Notifications operation again when time-outs occur.

287 Each attribute is encoded using the IPP rules for encoding attributes [RFC2910] and MAY be encoded
288 in any order. Note: the Get-Jobs response in [RFC2911] acts as a model for encoding multiple groups
289 of attributes. See section 11 for the encoding and transport rules.

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

291 Group 1: Operation Attributes

292   Status Message:
293   In addition to the REQUIRED status code returned in every response, the response
294   OPTIONALLY includes a "status-message" (text(255)) and/or a "detailed-status-message"
295   (text(MAX)) operation attribute as described in [RFC2911] sections 13 and 3.1.6.
296
297   The Printer can return any status codes defined in [RFC2911]. If the status code is not
298   'successful-xxx', the Printer MUST NOT return any Event Notification Attribute groups.
299   The following is a description of the important status codes:
300
301   **successful-ok:** the response contains all Event Notification associated with the specified
302   subscription-ids that had been supplied in the "notify-subscription-ids" operation
303   attribute in the request. If the requested Subscription Objects have no associated
304   Event Notification, the response MUST contain zero Event Notifications.

305 **successful-ok-events-complete:** indicate when this return is the last return for all
306 Subscription objects that match the request, whether or not there are Event
307 Notifications being returned. This condition occurs for **Event Wait Mode** with
308 Notification Recipients waiting for responses when the Subscription Object is: (1)
309 canceled with a Cancel-Subscription operation, (2) deleted when the Per-Printer
310 Subscription lease time expires, or (3) when the 'job-completed' event occurs for a
311 Per-Job Subscription. This condition also occurs for a Get-Notifications request that
312 a Notification Recipient makes after the job completes, but before the Event Life
313 expires. See section 10.1.
314 **client-error-not-found:** The Printer has no Subscription Object's whose "notify-
315 subscription-id" attribute equals any of the values of the "notify-subscription-ids"
316 operation attribute supplied or the identified Subscription Object does not contain the
317 "notify-pull-method" attribute with the 'ippget' keyword value.
318 **server-error-busy:** The Printer is too busy to accept this operation. The Printer
319 SHOULD return the "notify-get-interval" operation attribute in the Operation
320 Attributes of the response, then the Notification Recipient SHOULD wait for the
321 number of seconds specified by the "notify-get-interval" operation attribute before
322 performing this operation again. If the "notify-get-interval" Operation Attribute is
323 not present, the Notification Recipient SHOULD use the normal network back-off
324 algorithms for determining when to perform this operation again.

326 Natural Language and Character Set:
327 The "attributes-charset" and "attributes-natural-language" attributes as described in
328 [RFC2911] section 3.1.4.2.

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

334 Normally, there is only one matched Subscription Object, or the value of the "notify-charset"
335 and "notify-natural-language" attributes is the same in all Subscription Objects. If not, the
336 Printer MUST pick one Subscription Object from which to obtain the value of these
337 attributes. The algorithm for picking the Subscription Object is implementation dependent.
338 The choice of natural language is not critical because 'text' and 'name' values can override
339 the "attributes-natural-language" operation attribute. The Printer's choice of charset is
340 critical because a bad choice may leave it unable to send some 'text' and 'name' values
341 accurately.

343 **5.2.1 notify-get-interval (integer(0:MAX))**

344 The value of this operation attribute is the number of seconds that the Notification Recipient
345 SHOULD wait before trying the Get-Notifications operation again. The Printer MUST return
346 this operation attribute if: (1) it is too busy to return events, (2) the Notification Recipient
347 client did *not* request **Event Wait Mode**, or (3) the Printer is terminating Event Wait Mode.
348 The client MUST accept this attribute and SHOULD re-issue the Get-Notifications operation

349   (with or without "notify-wait" = 'true') the indicated number of seconds in the future in order
350   to get more Event Notifications This value is intended to help the client be a good network
351   citizen.
352
353   The value of this attribute MUST be at least as large as the value of the Printer's "ippget-
354   event-life" Printer Description attribute (see section 8.1). The Printer MAY return a value
355   that is larger than the value of the "ippget-event-life" Printer Description attribute provided
356   that the Printer increases the Event Life for this Subscription object, so that Notification
357   Recipients taking account of the larger value and polling with a longer interval will *not* miss
358   events. Note; implementing such an algorithm requires some hidden attributes in the
359   Subscription object that are IMPLEMENTATION DEPENDENT.
360
361   If the Printer wants to remain in **Event Wait Mode**, then the Printer MUST NOT return this
362   attribute in the response.
363
364   Here is a complete table of combinations of "notify-wait", "status-code", "notify-get-
365   interval", and Event Notification Attributes Groups for Get-Notification initial (Wait and No
366   Wait) Responses and subsequent **Event Wait Mode** Responses (which may be staying in
367   **Event Wait Mode** or may be requesting the Notification Recipient to leave **Event Wait**
368   **Mode**):
369

370   **Table 2 - Combinations of "notify-wait", "status-code", and "notify-get-interval"**

client sends: "notify-wait"	Printer returns: "status-code"	Printer returns: "notify-get-interval"	Event Notification Attribute Groups
1. 'false'*	'successful-ok'	MUST return N	maybe
2. 'false'*	'not-found'	MUST NOT	MUST NOT
3. 'false'*	'busy'	MUST return N	MUST NOT
4. 'false'*	'events-complete'	MUST NOT	'job-completed'
5. 'true'	'successful-ok'	MUST NOT	MUST
6. 'true'	'successful-ok'	MUST return N	maybe
7. 'true'	'not-found'	MUST NOT	MUST NOT
8. 'true'	'busy'	MUST return N	MUST NOT
9. 'true'	'events-complete'	MUST NOT	'job-completed' or maybe other

371   * 'false' or client omits the "notify-wait" attribute.
372
373   Explanation:
374
375   1-4: client does *not* request **Event Wait Mode**
376   5-9: client requests **Event Wait Mode**
377   2,7: Subscription object not found, or was canceled earlier; client should NOT try again.
378   3,8: server busy, tells client to try later; client should try again in N seconds.

379   4: client polled after job completed, but before Event Life expired, and got the 'job-
380   completed' event, so the client shouldn't bother trying again; client should NOT try again
381   later.
382   5: Printer returns one or more Event Notifications and is OK to stay in **Event Wait Mode**;
383   the client waits for more Event Notifications to be returned.
384   6: Printer wants to leave **Event Wait mode**. Can happen on the first response (with or
385   without Event Notifications) or happen on a subsequent response with or without Event
386   Notifications; the client SHOULD try again in N seconds.
387   9: Printer either (1) returns 'job-completed' event or (2) the Subscription Object was
388   canceled by either a Cancel-Job or a Per-Printer Subscription expired without being renewed.
389   For case (1), at least one Event Notification MUST be returned, while for case (2), it is
390   unlikely that any Event Notifications are returned; the client should NOT try again.

391   **5.2.2 printer-up-time (integer(1:MAX))**

392   The value of this attribute is the Printer's "printer-up-time" attribute at the time the Printer
393   sends this response. The Printer MUST return this attribute. Because each Event Notification
394   also contains the value of this attribute when the event occurred, the value of this attribute lets
395   a Notification Recipient know when each Event Notification occurred relative to the time of
396   this response.
397
398   Group 2: Unsupported Attributes

399   See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes.
400
401
402   Group 3 through N: Event Notification Attributes

403   The Printer responds with one Event Notification Attributes Group per matched Event
404   Notification. The entire response is considered a single Compound Event Notification (see
405   [ipp-ntfy]). The matched Event Notifications are all un-expired Event Notification associated
406   with the matched Subscription Objects and MUST follow the "Event Notification Ordering"
407   requirements for Event Notifications within a Compound Event Notification specified in [ipp-
408   ntfy] section 9. In other words, the Printer MUST order these Event Notification groups in
409   ascending time stamp (and sequence number) order for a Subscription object. If Event
410   Notifications for multiple Subscription objects are being returned, the Notification Events for
411   the next Subscription object follow in ascending time stamp order, etc.
412
413   Each Event Notification Group MUST contain all of attributes specified in section 9.1
414   ("Content of Machine Consumable Event Notifications") of [ipp-ntfy] with exceptions
415   denoted by asterisks in the tables below.
416
417   The tables below are copies of the tables in section 9.1 ("Content of Machine Consumable
418   Event Notifications") of [ipp-ntfy] except that each cell in the "Sends" column is a "MUST".
419

420   If more than one Event Notification is being returned and the status of each is not the same,
421   then the Printer MUST return a "notify-status-code" attribute in each Event Notification
422   Attributes group to indicate the differing status values.
423

424   For an Event Notification for all Events, the Printer includes the attributes shown in Table 3.

425   **Table 3 – 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(1: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

426

427   * As specified in [ipp-ntfy] section 9, the value of the "printer-up-time" attribute sent in each
428   Event Notification MUST be the time at which the Event occurred, not the time at which the
429   Event Notification was sent.
430

431   ** The Printer MUST send the "printer-current-time" attribute if and only if it supports the
432   "printer-current-time" attribute on the Printer object.
433

434   *** If the associated Subscription Object does not contain a "notify-user-data" attribute, the
435   Printer MUST send an octet-string of length 0.
436

437   **** If the "notify-attributes" attribute is present on the Subscription Object, the Printer
438   MUST send all attributes specified by the "notify-attributes" attribute. Note: if the Printer
439   doesn't support the "notify-attributes" attribute, it is not present on the associated
440   Subscription Object.
441

442   For Event Notifications for Job Events, the Printer includes the additional attributes shown in
443   Table 4.

444   **Table 4 – 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

445
446   * The Printer MUST send the "job-impressions-completed" attribute in an Event Notification
447   only for the combinations of Events and Subscribed Events shown in Table 5.
448

449   **Table 5 – 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'

450
451
452   For Event Notification for Printer Events, the Printer includes the additional attributes shown
453   in Table 6.

454   **Table 6 – 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

455 6 Additional Information about Subscription Template Attributes

456   The 'ippget' Delivery Method does not define any addition Subscription Template attributes. The
457   'ippget' Delivery Method has the same conformance requirements for Subscription Template attributes
458   as defined in [ipp-ntfy]. This section defines additional information about Subscription Template
459   attributes defined in [ipp-ntfy].

460 6.1 notify-pull-method (type2 keyword)

461   This Subscription Template attribute identifies the Pull Delivery Method to be used for the
462   Subscription Object (see [ipp-ntfy]). In order to support the 'ippget' Pull Delivery Method defined in
463   this document, the Printer MUST support this attribute with the following keyword value:

464   'ippget': indicates that the 'ippget' Pull Delivery Method is to be used for this Subscription Object.

7 Subscription Description Attributes

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

8 Additional Printer Description Attributes

470   This section defines additional Printer Description attributes for use with the 'ippget' Delivery
471   Method.

8.1 ippget-event-life (integer(15:MAX))

473   This Printer Description attribute specifies the Event Life value that the Printer assigns to each Event,
474   i.e., the number of seconds after an Event occurs during which a Printer will return that Event in an
475   Event Notification in a Get-Notifications response. After the Event Life expires for the Event, the
476   Printer MAY no longer return an Event Notification for that Event in a Get-Notifications response.

477   The Printer MUST support this attribute if it supports the 'ippget' Delivery Method. The value MUST
478   be 15 or more (at least 15 seconds) and 60 (seconds) is the RECOMMENDED value to align with the
479   PWG Job Monitoring MIB [RFC2707] jmGeneralJobPersistence and jmGeneralAttributePersistence
480   objects.

481   For example, assume the following:

482   1. a client performs a Job Creation operation that creates a Subscription Object associated with the
483   'ippget' Delivery Method, AND

484   2. an Event associated with the new Job occurs immediately after the Subscription Object is
485   created, AND

486   3. the same client or some other client performs a Get-Notifications operation such that the client
487   is *connected* N seconds after the Job Creation operation.

488   Then, if N is less than the value of this attribute, the client(s) performing the Get-Notifications
489   operations can expect not to miss any Event-Notifications, barring some unforeseen lack of memory
490   space in the Printer. Note: The client MUST initiate the Get-Notifications a time that is sufficiently
491   less that N seconds to account for network latency so that it is *connected* to the Printer before N
492   seconds elapses.

493   If a Printer supports the 'ippget' Delivery Method, it MUST keep 'completed', 'canceled', or 'aborted'
494   Job objects in the Job Retention and/or Job History phases for at least as long as this attribute's value.
495   The Printer MAY retain jobs longer that this value. See [RFC2911] section 4.3.7.1 and the discussion
496   in [ipp-ntfy] 'job-completed' event) that explains that a Notification Recipients can query the Job after

497   receiving a 'job-completed' Event Notification in order to find out other information about the job that
498   is 'completed', 'aborted', or 'canceled'. However, this attribute has no effect on the Cancel-
499   Subscription operation which deletes the Subscription object immediately, whether or not it contain the
500   "notify-pull-method" attribute with the 'ippget' keyword value. Immediately thereafter, subsequent
501   Get-Notifications Responses MUST NOT contain Event Notifications associated with the canceled
502   Subscription object.

503   # 9 New Values for Existing Printer Description Attributes

504   This section defines additional values for existing Printer Description attributes defined in [ipp-ntfy].

505   ## 9.1 notify-pull-method-supported (1setOf type2 keyword)

506   The following keyword value for the "notify-pull-method-supported" attribute is added in order to
507   support the new Delivery Method defined in this document:

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

509   ## 9.2 operations-supported (1setOf type2 enum)

510   Table 7 lists the "operation-id" value defined in order to support the new Get-Notifications operation
511   defined in this document.

512   **Table 7 – Operation-id assignments**

Value	Operation Name
0x001C	Get-Notifications

513

514   # 10 New Status Codes

515   The following status code is defined as an extension for this Delivery Method and is returned as the
516   status code of the Get-Notifications operation in Group 1 or Group 3 to N (see section 5.2).

517   ## 10.1 successful-ok-events-complete (0x0007)

518   The Printer MUST return the 'successful-ok-events-complete' status code to indicate when this Get-
519   Notifications response is the last response for a Subscription object, whether or not there are Event
520   Notifications being returned. This condition occurs for **Event Wait Mode** with Notification
521   Recipients waiting for responses when the Subscription Object is: (1) canceled with a Cancel-
522   Subscription operation, (2) deleted when the Per-Printer Subscription lease time expires, or (3) when
523   the 'job-completed' event occurs for a Per-Job Subscription. This condition also occurs for a Get-

524 Notifications request that a Notification Recipient makes after the job completes, but before the Event
525 Life expires.

11 Encoding and Transport

527 This section defines the encoding and transport considerations for this Delivery Method based on
528 [RFC2910].

529 The encoding of a Get-Notifications Response is modeled the Get-Jobs Response (see [RFC2911]). In
530 a Get-Notifications Response, each Event Notification Attributes Group MUST start with an 'event-
531 notification-attributes-tag' (see the section "Encodings of Additional Attribute Tags" in [ipp-ntfy]),
532 and end with an 'end-of-attributes-tag'. In addition, for **Event Wait Mode** the multi-part/related is
533 used to separate each multiple response (in time) to a single Get-Notifications Request.

534 The Printer returns Get-Notification Response as follows:

535   1. If the Notification Recipient client did not request **Event Wait Mode** ("notify-wait" = 'false'
536   or omitted), the Printer ends the response with an 'end-of-attributes-tag' (see [RFC2911] Get-
537   Jobs encoding) as with any operation response.

538   2. If the Notification Recipient client requests **Event Wait Mode** ("notify-wait" = 'true') and the
539   Printer wishes to honor the request, the Printer MUST return the response as an application/ipp
540   part inside a multi-part/related MIME media type. When one or more additional Events occur,
541   the Printer returns each as an additional Event Notification Group using a separate
542   application/ipp part under the multi-part/related type.

543   3. If the client requested **Event Wait Mode** ("notify-wait" = 'true'), but the Printer does not wish
544   to honor the request in the initial response but wants the client explicitly poll for Event
545   Notifications, the Printer MUST return the "notify-get-interval" operation attribute (see section
546   5.2.1). The Printer returns the response as an application/ipp part which MAY be inside an
547   multi-part/related type. The client MUST accept this response and re-issue the Get-
548   Notifications request in the future indicated by the value of the "notify-get-interval" attribute
549   value..

550   4. If the client requested **Event Wait Mode** ("notify-wait" = 'true'), and the Printer initially
551   honored the request, but later wishes to leave **Event Wait Mode**, the Printer MUST return the
552   "notify-get-interval" operation attribute (see section 5.2.1). The Printer returns the response as
553   an application/ipp part which MUST be inside an multi-part/related type.

554 Note: All of the above is without either the Printer or the Notification Recipient closing the
555 connection. In fact, the connection SHOULD remain open for any subsequent IPP operations.
556 However, either the Notification Recipient or the Printer can abnormally terminate by closing the
557 connection. But, if the Printer closes the connection too soon after returning the response, the client
558 may not receive the response.

559 The Printer MAY chunk the responses, but this has no significance to the IPP semantics.

560   Note: While HTTP/1.1 allows a proxy to collect chunked responses over a period of time and return
561   them back as a single un-chunked response (with a Content Length instead). However, in practice no
562   proxy wants to have an infinite buffer. Also no proxy want to hold up responses, since user would be
563   furious.

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

566   **Table 8 – The "event-notification-attributes-tag" value**

Tag Value (Hex)	Meaning
0x07	"event-notification-attributes-tag"

567

12 Conformance Requirements

569   This section lists the conformance requirements for clients and Printers.

12.1 Conformance for IPP Printers

571   It is OPTIONAL for a Printer to support IPP Notifications as defined in [ipp-ntfy]. However, if a
572   Printer supports IPP Notifications, the Printer MUST support the 'ippget' Delivery Method as defined
573   in this document as one of its Delivery Methods. IPP Printers that conform to this specification:

574   1. MUST meet the conformance requirements defined in [ipp-ntfy] for a Pull Delivery Method;

575   2. MUST support the Get-Notifications operation defined in section 5, including **Event Wait**
576   **Mode**;

577   3. MUST support the Subscription Template object attributes as defined in section 6;

578   4. MUST support the Subscription Description object attributes as defined in section 7;

579   5. MUST support the "ippget-event-life" Printer Description attribute defined in section 8.1,
580   including retaining jobs in the Job Retention and/or Job History phases for at least as long as
581   the value specified by the Printer's "ippget-event-life";

582   6. MUST support the additional values for IPP/1.1 Printer Description attributes defined in
583   section 9;

584   7. MUST support the 'successful-ok-events-complete' status code as described in section 10.1;

585   8. MUST listen for the IPP Get-Notifications operation requests on IANA-assigned well-known
586   port 631, unless explicitly configured by system administrators or site policies;

587   9.   SHOULD NOT listen for IPP Get-Notifications operation requests on any other port, unless
588   explicitly configured by system administrators or site policies.

589   10. MUST meet the security conformance requirements as stated in section 18.4.

12.2 Conformance for IPP Clients

591   It is OPTIONAL for an IPP Client to support IPP Notifications as defined in [ipp-ntfy]. However, if a
592   client supports IPP Notifications, the client MUST support the 'ippget' Delivery Method as defined in
593   this document as one of its Delivery Methods. IPP Clients that conform to this specification:

594   1.   MUST create Subscription Objects by sending Subscription Creation operation requests
595   containing the "notify-pull-method" attribute (as opposed to the "notify-recipient-uri" attribute)
596   using the 'ippget' keyword value (see sections 6.1 and 15.2);

597   2.   MUST send IPP Get-Notifications operation requests (see section 5.1) via the port specified in
598   the associated 'ipp' URL (if present) or otherwise via IANA assigned well-known port 631;

599   3.   MUST convert the associated 'ipp' URLs for use in IPP Get-Notifications operation to their
600   corresponding 'http' URL forms for use in the HTTP layer according to the rules in section 5
601   "IPP URL Scheme" in [RFC2910].

602   4.   MUST meet the security conformance requirements as stated in section 18.5.

13 Normative References

604   [ipp-ntfy]
605   Herriot, R., and T. Hastings, "Internet Printing Protocol/1.1: IPP Event Notifications and
606   Subscriptions", <draft-ietf-ipp-not-spec-10.txt>, September 10, 2002.

607   [RFC2119]
608   S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March 1997

609   [RFC2910]
610   Herriot, R., Butler, S., Moore, P., and R. Tuner, "Internet Printing Protocol/1.1: Encoding and
611   Transport", RFC 2910, September 2000.

612   [RFC2911]
613   deBry, R., Hastings, T., Herriot, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.1:
614   Model and Semantics", RFC 2911, September 2000.

14 Informative References

[notify-req]

Hastings, T., deBry, R., and H. Lewis, "Internet Printing Protocol (IPP): Requirements for IPP Notifications"**,** <draft-ietf-ipp-not-06.txt>, work in progress, July 17, 2001.

[RFC2565]

Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.0: Encoding and Transport", RFC 2565, April 1999.

[RFC2566]

R. deBry, T. Hastings, R. Herriot, S. Isaacson, and P. Powell, "Internet Printing Protocol/1.0: Model and Semantics", RFC 2566, April 1999.

[RFC2567]

Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.

[RFC2568]

Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol", RFC 2568, April 1999.

[RFC2569]

Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC 2569, April 1999.

[RFC2616]

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

[RFC2707]

Bergman, R., Hastings, T., Isaacson, S., and H. Lewis, "Job Monitoring MIB - V1.0", November 1999.

[RFC3196]

Hastings, T., Manros, C., Zehler, P., Kugler, C., and H. Holst, "Internet Printing Protocol/1.1: Implementer's Guide", RFC3196, November 2001.

15 IANA Considerations

This section contains the exact information for IANA to add to the IPP Registries according to the procedures defined in RFC 2911 [RFC2911] section 6. The resulting registrations will be published in the http://www.iana.org/assignments/ipp-registrations registry.

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

15.1 Attribute Registrations

The following table lists the attributes defined in this document. This is to be registered according to the procedures in RFC 2911 [RFC2911] section 6.2.

```
Printer Description attributes:               Reference   Section:
--------------------------------              ---------   -------
ippget-event-life (integer(15:MAX))           [RFCNNNN]   8.1
```

15.2 Delivery Method and Additional keyword attribute value registrations for existing attributes

This section lists additional keyword attribute value registrations for use with existing attributes defined in other documents. These are to be registered according to the procedures in RFC 2911 [RFC2911] section 6.1. According to [ipp-ntfy] section 24.7.3, Pull Delivery Method registrations are the keyword attribute value registrations for the "notify-pull-method" and "notify-pull-method-supported" attributes.

```
Attribute (attribute syntax)
  Values:                                     Reference   Section:
  -----------------------                     ---------   -------
notify-pull-method (type2 keyword)            [ipp-ntfy]  5.3.2
notify-pull-method-supported (1setOf type2 keyword)
                                              [ipp-ntfy]  5.3.2.1
  ippget                                      [RFCNNNN]   9.1
```

15.3 Additional enum attribute values

The following table lists the enum attribute values defined in this document. These are to be registered according to the procedures in RFC 2911 [RFC2911] section 6.1.

```
Attribute (attribute syntax)
  Value    Name                               Reference   Section
  ------   ------------------------------     ---------   -------
operations-supported (type2 enum)             [RFC2911]   4.4.15
  0x001C   Get-Notifications                  [RFCNNNN]   9.2
```

15.4 Operation Registrations

The following table lists the operations defined in this document. This is to be registered according to the procedures in RFC 2911 [RFC2911] section 6.4.

```
Operations:                                   Reference   Section
-----------                                   ---------   -------
Get-Notifications operation                   [RFCNNNN]   5
```

15.5 Status code Registrations

The following table lists the status codes defined in this document. This is to be registered according to the procedures in RFC 2911 [RFC2911] section 6.6.

```
Status codes:                                    Reference   Section
-------------                                    ---------   -------
successful-ok-events-complete (0x0007)           [RFCNNNN]   10.1
```

16 Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in RFC 2028. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

17 Internationalization Considerations

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

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

18 Security Considerations

The IPP Model and Semantics document [RFC2911 section 8] discusses high-level security requirements (Client Authentication, Server Authentication and Operation Privacy). The IPP Transport and Encoding document [RFC2910 section 8] discusses the security requirements for the IPP protocol. Client Authentication is the mechanism by which the client proves its identity to the server in a secure manner. Server Authentication is the mechanism by which the server proves its identity to the client in a secure manner. Operation Privacy is defined as a mechanism for protecting operations from eavesdropping.

719   The 'ippget' Delivery Method with its Get-Notifications operations leverages the security mechanism
720   that are used in IPP/1.1 [RFC2910 and RFC2911] without adding any additional security mechanisms
721   in order to maintain the same security support as IPP/1.1.

722   The access control model for the Get-Notifications operation defined in this document is the same as
723   the access control model for the Get-Job-Attributes operation (see [RFC2911] section 3.2.6). The
724   primary difference is that a Get-Notifications operation is directed at Subscription Objects rather than
725   at Job objects, and a returned attribute group contains Event Notification attributes rather than Job
726   object attributes.

727   **18.1 Notification Recipient client access rights**

728   The Notification Recipient client MUST have the following access rights to the Subscription object(s)
729   targeted by the Get-Notifications operation request:

730   The authenticated user (see [RFC2911] section 8.3) performing this operation MUST be (1) the
731   owner of each Subscription Object identified by the "notify-subscription-ids" operation attribute
732   (see section 5.1.1), (2) an operator or administrator of the Printer (see [RFC2911] Sections 1 and
733   8.5), or (3) be otherwise authorized by the Printer's administrator-configured security policy to
734   request Event Notifications from the target Subscription Object(s). Furthermore, the Printer's
735   security policy MAY limit the attributes returned by the Get-Notifications operation, in a manner
736   similar to the Get-Job-Attributes operation (see [RFC2911] end of section 3.3.4.2).

737   **18.2 Printer security threats**

738   Because the Get-Notifications operation is sent in the same direction as Job Creation operations,
739   usually by the same client, this Event Notification Delivery Method poses no additional authentication,
740   authorization, privacy, firewall, or port assignment issues above those for the IPP Get-Job-Attributes
741   and Get-Printer-Attributes operations (see [RFC2911] sections 3.2.6 and 3.2.5).

742   **18.3 Notification Recipient security threats**

743   Unwanted Events Notifications (spam): Unlike Push Event Notification Delivery Methods in which
744   the IPP Printer initiates the Event Notification, with the Pull Delivery Method defined in this
745   document, the Notification Recipient is the client who initiates the Get-Notifications operation (see
746   section 5). Therefore, there is no chance of "spam" notifications with this method.

747   Note: when a client stays connected to a Printer using the Event Wait Mode (see section 5.1.3) in
748   order to receive Event Notifications as they occur, such a client can close down the IPP connection at
749   any time, and so can avoid future unwanted Event Notifications at any time.

750   It is true that client has control about whether to ask for Event Notifications. However, if the client
751   subscribes to an event, and does a Get-Notifications request, the client gets all events for the
752   Subscription Object in the sequence number range (see section 5.1.2), not just the ones the client
753   wants. If a client subscribes to a Per-Printer Subscription job event, such as 'job-completed', and

754   someone then starts and cancels thousands of jobs, the client would have to receive these events in
755   addition to the ones the client is interested in. A client can protect itself better by subscribing to his
756   own jobs using a Per-Job Subscription, rather than creating a Per-Printer subscription whose Job
757   events apply to all jobs.

758   **18.4 Security requirements for Printers**

759   For the Get-Notifications operation defined in this document, the same Printer conformance
760   requirements apply for supporting and using Client Authentication, Server Authentication and
761   Operation Privacy as stated in [RFC2910] section 8 for all IPP operations.

762   **18.5 Security requirements for clients**

763   For the Get-Notifications operation defined in this document, the same client conformance
764   requirements apply for supporting and using Client Authentication, Server Authentication and
765   Operation Privacy as stated in [RFC2910] section 8 for all IPP operations.

766   # 19 Contributors

767   Carl Kugler and Harry Lewis contributed the basic idea of in-band "smart polling" coupled with
768   multiple responses for a single operation on the same connection, one response for each event as it
769   occurs. Without their continual persuasion, we would not have arrived at this Delivery Method
770   specification and would not have been able to agree on a single REQUIRED Delivery Method for IPP.

771   Carl Kugler
772   IBM
773   P.O. Box 1900
774   Boulder, CO 80301-9191
775
776   Phone:
777   Fax:
778   e-mail: kugler@us.ibm.com
779

780   # 20 Authors' Addresses

781
782   Robert Herriot
783   706 Colorado Ave.
784   Palo Alto, CA 94303
785
786   Phone: 650-327-4466
787   Fax: 650-327-4466

788   email: bob@herriot.com
789
790   Thomas N. Hastings
791   Xerox Corporation
792   737 Hawaii St. ESAE 231
793   El Segundo CA 90245
794
795   Phone: 310-333-6413
796   Fax: 310-333-5514
797   email: hastings@cp10.es.xerox.com
798
799   Harry Lewis
800   IBM
801   P.O. Box 1900
802   Boulder, CO 80301-9191
803
804   Phone: 303-924-5337
805   FAX:
806   e-mail: harryl@us.ibm.com
807
808
809   IPP Web Page: http://www.pwg.org/ipp/
810   IPP Mailing List: ipp@pwg.org
811
812   To subscribe to the ipp mailing list, send the following email:
813   1) send it to majordomo@pwg.org
814   2) leave the subject line blank
815   3) put the following two lines in the message body:
816   subscribe ipp
817   end
818
819   Implementers of this specification document are encouraged to join the IPP Mailing List in order to
820   participate in any discussions of clarification issues and review of registration proposals for additional
821   attributes and values. In order to reduce spam the mailing list rejects mail from non-subscribers, so
822   you must subscribe to the mailing list in order to send a question or comment to the mailing list.

823 **21 Description of Base IPP documents (Informative)**

824   The base set of IPP documents includes:

825   Design Goals for an Internet Printing Protocol [RFC2567]
826   Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
827   Internet Printing Protocol/1.1: Model and Semantics [RFC2911]
828   Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
829   Internet Printing Protocol/1.1: Implementer's Guide [RFC3196]

830   Mapping between LPD and IPP Protocols [RFC2569]

831

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

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

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

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

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

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

857   ## 22 Full Copyright Statement

875   **Acknowledgement**
876