

1 Internet Printing Protocol WG R. Herriot (~~editor~~)
2 INTERNET-DRAFT ~~consultant~~
3 <draft-ietf-ipp-not-spec-098.txt> T. Hastings ~~M. Shepherd~~
4 Updates RFC 2910 and 2911 Xerox Corporation
5 [Target Category: standards track] ~~June 27, 2002~~ ~~November 19, 2001~~
6 Expires: ~~December 27~~ ~~May 19~~, 2002 ~~R. deBry~~
7 ~~Utah Valley State College~~
8 ~~S. Isaacson~~
9 ~~Novell, Inc.~~
10 ~~J. Martin~~
11 ~~Underseore~~
12 ~~R. Bergman~~
13 ~~Hitachi Koki Imaging Solutions~~

14
15 Internet Printing Protocol (IPP):
16 **Event Notifications and Subscriptions**
17

18 Copyright (C) The Internet Society (2002~~+~~). All Rights Reserved.

19 Status of this Memo

20 This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of
21 [\[RFC_2026\]](#). Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its
22 areas, and its working groups. Note that other groups may also distribute working documents as
23 Internet-Drafts.

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

27 The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.html>~~txt~~
28 The list of Internet-Draft Shadow Directories can be accessed as <http://www.ietf.org/shadow.html>.

29 **Abstract**

30 This document describes an OPTIONAL extension to the Internet Printing Protocol [/1.1: Model and](#)
31 [Semantics/1.0 \(IPP\) \[RFC2566, RFC2565\]](#) and IPP/1.1 ~~{(RFC_2911, RFC_2910)}~~. This extension
32 allows a client to subscribe to printing related Events. Subscriptions are modeled as *Subscription*
33 *Objects*. The Subscription Object specifies that when one of the specified *Events* occurs, the Printer
34 sends an asynchronous *Event Notification* to the specified *Notification Recipient* via the specified [Push](#)
35 [or Pull Delivery Method](#) (i.e., protocol).

36 A client associates Subscription Objects with a particular Job by performing the Create-Job-
37 Subscriptions operation or by submitting a Job with subscription information. A client associates
38 Subscription Objects with the Printer by performing a Create-Printer-Subscriptions operation. Four

39 other operations are defined for Subscription Objects: Get-Subscriptions-Attributes, Get-Subscriptions,
40 Renew-Subscription, and Cancel-Subscription.
41

41 **Table of Contents**

| | | |
|----|---|----|
| 42 | 1 Introduction..... | 7 |
| 43 | 1.1 Notification Overview | 8 |
| 44 | 2 Models for Notification | 10 |
| 45 | 2.1 Model for Notification (Simple Case) | 10 |
| 46 | 2.2 Model for Notification with Cascading Printers..... | 10 |
| 47 | 2.3 Distributed Model for Notification | 11 |
| 48 | 2.4 Extended Notification Recipient..... | 11 |
| 49 | 3 Terminology..... | 11 |
| 50 | 3.1 Conformance Terminology | 11 |
| 51 | 3.2 Other Terminology..... | 12 |
| 52 | 4 Object Relationships | 14 |
| 53 | 4.1 Printer and Per-Printer Subscription Objects..... | 14 |
| 54 | 4.2 Printer, Job and Per-Job Subscription Objects..... | 14 |
| 55 | 5 Subscription Object..... | 15 |
| 56 | 5.1 Rules for Support of Subscription Template Attributes | 15 |
| 57 | 5.2 Rules for Processing Subscription Template Attributes | 16 |
| 58 | 5.3 Subscription Template Attributes | 19 |
| 59 | 5.3.1 notify-recipient-uri (uri)..... | 20 |
| 60 | 5.3.2 notify-pull-method (type2 keyword)..... | 21 |
| 61 | 5.3.3 notify-events (1setOf type2 keyword)..... | 21 |
| 62 | 5.3.3.1 Standard Values for Subscribed Events..... | 22 |
| 63 | 5.3.3.1.1 No Events..... | 22 |
| 64 | 5.3.3.1.2 Subscribed Printer Events | 22 |
| 65 | 5.3.3.1.3 Subscribed Job Events | 24 |
| 66 | 5.3.3.2 Rules for Matching of Subscribed Events | 25 |
| 67 | 5.3.3.2.1 Rules for Matching of Printer Events | 25 |
| 68 | 5.3.3.2.2 Rules for Matching of Job Events..... | 25 |
| 69 | 5.3.3.2.3 Special Cases for Matching Rules | 26 |
| 70 | 5.3.4 notify-attributes (1setOf type2 keyword)..... | 26 |
| 71 | 5.3.5 notify-user-data (octetString(63)) | 27 |
| 72 | 5.3.6 notify-charset (charset) | 28 |
| 73 | 5.3.7 notify-natural-language (naturalLanguage)..... | 28 |
| 74 | 5.3.8 notify-lease-duration (integer(0:67108863))..... | 29 |
| 75 | 5.3.9 notify-time-interval (integer(0:MAX)) | 30 |
| 76 | 5.4 Subscription Description Attributes..... | 31 |
| 77 | 5.4.1 notify-subscription-id (integer (1:MAX))..... | 31 |
| 78 | 5.4.2 notify-sequence-number (integer (0:MAX))..... | 32 |
| 79 | 5.4.3 notify-lease-expiration-time (integer(0:MAX))..... | 32 |
| 80 | 5.4.4 notify-printer-up-time (integer(1:MAX))..... | 33 |

| | | |
|-----|---|----|
| 81 | 5.4.5 notify-printer-uri (uri) | 33 |
| 82 | 5.4.6 notify-job-id (integer(1:MAX)) | 33 |
| 83 | 5.4.7 notify-subscriber-user-name (name(MAX)) | 34 |
| 84 | 6 Printer Description Attributes Related to Notification | 34 |
| 85 | 6.1 printer-state-change-time (integer(1:MAX)) | 35 |
| 86 | 6.2 printer-state-change-date-time (dateTime) | 35 |
| 87 | 7 New Values for Existing Printer Description Attributes | 35 |
| 88 | 7.1 operations-supported (1setOf type2 enum)..... | 35 |
| 89 | 8 Attributes Only in Event Notifications | 36 |
| 90 | 8.1 notify-subscribed-event (type2 keyword) | 36 |
| 91 | 8.2 notify-text (text(MAX))..... | 36 |
| 92 | 9 Event Notification Content | 36 |
| 93 | 9.1 Content of Machine Consumable Event Notifications | 39 |
| 94 | 9.1.1 Event Notification Content Common to All Events | 39 |
| 95 | 9.1.2 Additional Event Notification Content for Job Events..... | 40 |
| 96 | 9.1.3 Additional Event Notification Content for Printer Events..... | 41 |
| 97 | 9.2 Content of Human Consumable Event Notification | 41 |
| 98 | 9.2.1 Event Notification Content Common to All Events | 42 |
| 99 | 9.2.2 Additional Event Notification Content for Job Events..... | 43 |
| 100 | 9.2.3 Additional Event Notification Content for Printer Events..... | 44 |
| 101 | 10 Delivery Methods..... | 44 |
| 102 | 11 Operations for Notification | 46 |
| 103 | 11.1 Subscription Creation Operations | 46 |
| 104 | 11.1.1 Create-Job-Subscriptions Operation | 46 |
| 105 | 11.1.1.1 Create-Job-Subscriptions Request | 47 |
| 106 | 11.1.1.1.1 notify-job-id (integer(1:MAX)) | 47 |
| 107 | 11.1.1.2 Create-Job-Subscriptions Response..... | 48 |
| 108 | 11.1.2 Create-Printer-Subscriptions operation..... | 49 |
| 109 | 11.1.2.1 Create-Printer-Subscriptions Request..... | 49 |
| 110 | 11.1.2.2 Create-Printer-Subscriptions Response | 49 |
| 111 | 11.1.3 Job Creation Operations – Extensions for Notification..... | 49 |
| 112 | 11.1.3.1 Job Creation Request | 50 |
| 113 | 11.1.3.2 Job Creation Response..... | 51 |
| 114 | 11.2 Other Operations | 52 |
| 115 | 11.2.1 Restart-Job Operation – Extensions for Notification..... | 52 |
| 116 | 11.2.2 Validate-Job Operation – Extensions for Notification | 52 |
| 117 | 11.2.3 Get-Printer-Attributes – Extensions for Notification..... | 52 |
| 118 | 11.2.4 Get-Subscription-Attributes operation..... | 53 |
| 119 | 11.2.4.1 Get-Subscription-Attributes Request..... | 53 |
| 120 | 11.2.4.1.1 “notify-subscription-id” (integer (1:MAX)) | 54 |

| | | |
|-----|---|----|
| 121 | 11.2.4.1.2 “requested-attributes” (1setOf keyword) | 54 |
| 122 | 11.2.4.2 Get-Subscription-Attributes Response | 54 |
| 123 | 11.2.5 Get-Subscriptions operation | 55 |
| 124 | 11.2.5.1 Get-Subscriptions Request..... | 56 |
| 125 | 11.2.5.1.1 “notify-job-id” (integer(1:MAX))..... | 56 |
| 126 | 11.2.5.1.2 “limit” (integer(1:MAX))..... | 57 |
| 127 | 11.2.5.1.3 “requested-attributes” (1setOf type2 keyword)..... | 57 |
| 128 | 11.2.5.1.4 “my-subscriptions” (boolean) | 57 |
| 129 | 11.2.5.2 Get-Subscriptions Response | 57 |
| 130 | 11.2.6 Renew-Subscription operation..... | 58 |
| 131 | 11.2.6.1 Renew-Subscription Request..... | 58 |
| 132 | 11.2.6.1.1 “notify-subscription-id” (integer (1:MAX)) | 59 |
| 133 | 11.2.6.1.2 “notify-lease-duration” (integer(0:MAX))..... | 59 |
| 134 | 11.2.6.2 Renew-Subscription Response | 59 |
| 135 | 11.2.6.2.1 “notify-lease-duration” (integer(0:MAX))..... | 60 |
| 136 | 11.2.7 Cancel-Subscription operation..... | 60 |
| 137 | 11.2.7.1 Cancel-Subscription Request..... | 61 |
| 138 | 11.2.7.1.1 “notify-subscription-id” (integer (1:MAX)) | 61 |
| 139 | 11.2.7.2 Cancel-Subscription Response | 61 |
| 140 | 12 Status Codes..... | 62 |
| 141 | 12.1 successful-ok-ignored-subscriptions (0x0003) | 62 |
| 142 | 12.2 client-error-ignored-all-subscriptions (0x0414)..... | 62 |
| 143 | 13 Status Codes in Subscription Attributes Groups..... | 63 |
| 144 | 13.1 client-error-uri-scheme-not-supported (0x040C)..... | 63 |
| 145 | 13.2 client-error-attributes-or-values-not-supported (0x040B) | 63 |
| 146 | 13.3 client-error-too-many-subscriptions (0x0415)..... | 63 |
| 147 | 13.4 successful-ok-too-many-events (0x0005) | 63 |
| 148 | 13.5 successful-ok-ignored-or-substituted-attributes (0x0001)..... | 63 |
| 149 | 14 Encodings of Additional Attribute Tags | 64 |
| 150 | 15 Conformance Requirements..... | 64 |
| 151 | 15.1 Conformance requirements for clients..... | 64 |
| 152 | 15.2 Conformance requirements for Printers..... | 64 |
| 153 | 16 Normative References..... | 65 |
| 154 | 17 Informative References | 66 |
| 155 | 18 Security Considerations | 67 |
| 156 | 18.1 Client access rights | 67 |
| 157 | 18.2 Printer security threats | 69 |
| 158 | 18.3 Notification Recipient security threats..... | 69 |

| | | |
|-----|--|----|
| 159 | 19 IANA Considerations..... | 69 |
| 160 | 19.1 Attribute Registrations..... | 70 |
| 161 | 19.2 Additional Enum Attribute Value Registrations for the “operations-supported” Printer Attribute..... | 71 |
| 162 | 19.3 Operation Registrations..... | 71 |
| 163 | 19.4 Status code Registrations..... | 72 |
| 164 | 19.5 Attribute Group tag Registrations..... | 72 |
| 165 | 19.6 Registration of Events..... | 72 |
| 166 | 19.7 Registration of Event Notification Delivery Methods..... | 73 |
| 167 | 19.7.1 Requirements for Registration of Event Notification Delivery Methods..... | 73 |
| 168 | 19.7.1.1 Required Characteristics..... | 73 |
| 169 | 19.7.1.2 Naming Requirements..... | 74 |
| 170 | 19.7.1.3 Functionality Requirements..... | 74 |
| 171 | 19.7.1.4 Usage and Implementation Requirements..... | 74 |
| 172 | 19.7.1.5 Publication Requirements..... | 74 |
| 173 | 19.7.2 Registration Procedure..... | 74 |
| 174 | 19.7.2.1 Present the proposal to the Community..... | 74 |
| 175 | 19.7.2.2 Delivery Method Reviewer..... | 75 |
| 176 | 19.7.2.3 IANA Registration..... | 75 |
| 177 | 19.7.3 Delivery Method Document Registrations..... | 75 |
| 178 | 19.7.4 Registration Template..... | 76 |
| 179 | 20 Internationalization Considerations..... | 76 |
| 180 | 21 Contributors..... | 76 |
| 181 | 22 Author’s Addresses..... | 77 |
| 182 | A. Appendix - Model for Notification with Cascading Printers..... | 79 |
| 183 | B. Appendix - Distributed Model for Notification..... | 80 |
| 184 | C. Appendix - Extended Notification Recipient..... | 81 |
| 185 | D. Appendix - Details about Conformance Terminology..... | 82 |
| 186 | E. Appendix - Object Model for Notification..... | 82 |
| 187 | E.1 Appendix - Object relationships..... | 83 |
| 188 | E.2 Printer Object and Per-Printer Subscription Objects..... | 83 |
| 189 | E.3 Job Object and Per-Job Subscription Objects..... | 84 |
| 190 | F. Appendix - Per-Job versus Per-Printer Subscription Objects..... | 84 |
| 191 | G. Appendix - Description of the base IPP documents..... | 84 |
| 192 | H. Appendix - Full Copyright Statement..... | 85 |

193
194 **Tables**

| | | |
|-----|---|----|
| 195 | Table 1 – Subscription Template Attributes | 20 |
| 196 | Table 2 – Subscription Description Attributes | 31 |
| 197 | Table 3 – Printer Description Attributes Associated with Notification..... | 34 |
| 198 | Table 4 – Operation-id assignments | 36 |
| 199 | Table 5 – Attributes in Event Notification Content..... | 40 |
| 200 | Table 6 – Additional Event Notification Content for Job Events..... | 41 |
| 201 | Table 7 – Combinations of Events and Subscribed Events for “job-impressions-completed”..... | 41 |
| 202 | Table 8 – Additional Event Notification Content for Printer Events..... | 41 |
| 203 | Table 9 – Printer Name in Event Notification Content..... | 43 |
| 204 | Table 10 – Event Name in Event Notification Content | 43 |
| 205 | Table 11 – Event Time in Event Notification Content | 43 |
| 206 | Table 12 – Job Name in Event Notification Content..... | 43 |
| 207 | Table 13 – Job State in Event Notification Content | 44 |
| 208 | Table 14 – Printer State in Event Notification Content..... | 44 |
| 209 | Table 15 – Information about the Delivery Method | 45 |
| 210 | Table 16 – Printer Conformance Requirements for Operations | 65 |

211

212 **Figures**

| | | |
|-----|---|----|
| 213 | Figure 1 – Model for Notification..... | 10 |
| 214 | Figure 2 – Model for Notification with Cascading Printers..... | 80 |
| 215 | Figure 3 – Opaque Use of a Notification Service Transparent to the Client | 81 |
| 216 | Figure 4 – Use of an Extended Notification Recipient transparent to the Printer | 82 |
| 217 | Figure 5 – Object Model for Notification | 83 |

218

219 **1 Introduction**

220 This IPP notification specification is an OPTIONAL extension to Internet Printing Protocol/1.1: [Model](#)
 221 [and Semantics/1.0 \(IPP\) \[RFC2566, RFC2565\]](#) and [IPP/1.1](#) [RFC2911, RFC2910]. See Appendix G
 222 for a description of the base IPP documents. This document in combination with the following
 223 documents is intended to meet the [most important](#) notification requirements described in [ipp-not-req]:

224

Internet Printing Protocol (IPP): “Job Progress Attributes” [ipp-prog]

225

[Internet Printing Protocol \(IPP\): “The ‘ippget’ Delivery Method for Event Notifications” \[ipp-
 226 \[get-method\\]\]\(#\)](#)

226

~~[One or more Delivery Method Documents registered with IANA \(see section 10\).](#)~~

227

228

229 [This specification REQUIRES that clients and Printers support the ‘ippget’ Pull Delivery Method \[ipp-
 230 \[get-method\\]\]\(#\). \[Conforming client and Printer implementations MAY support additional Push or Pull
 231 \\[Delivery Methods as well.\\]\\(#\\) Note: this document does not define any Delivery Methods \\[itself\\]\\(#\\), but it
 232 does define the rules for conformance for Delivery Method Documents. ~~\\[Delivery Method Documents
 233 \\\[are in preparation\\\]\\\(#\\\) \\\(see section 10\\\) and \\\[will be registered\\\]\\\(#\\\) \\\[their registration\\\]\\\(#\\\) with IANA \\\(see section
 234 19.7.3\\\).\\]\\(#\\)~~\]\(#\)](#)

235

Refer to the Table of Contents for the layout of this document.

236 1.1 Notification Overview

237 This document defines operations that a client can perform in order to create *Subscription Objects* in a
238 Printer and carry out other operations on them. A Subscription Object represents a Subscription
239 abstraction. The Subscription Object specifies that when one of the specified *Events* occurs, the Printer
240 sends an asynchronous *Event Notification* to the specified *Notification Recipient* via the specified
241 *Delivery Method* (i.e., protocol).

242 When a client (called a *Subscribing Client*) performs an operation that creates a Subscription Object,
243 the operation contains one or more *Subscription Template Attributes Groups*. Each such group holds
244 information used by the Printer to initialize a newly created Subscription Object. The Printer creates
245 one Subscription Object for each Subscription Template Attributes Group in the operation. This group
246 is like the Job Template Attributes group defined in [RFC2911]. The following is an example of the
247 information included in a Subscription Template Attributes Group (see section 5 for details on the
248 Subscription Object attributes):

- 249 1. The names of Subscribed Events that are of interest to the Notification Recipient.
- 250 2. The address (URL) of one Notification Recipient for a Push Delivery Method or the method for
251 a Pull Delivery Method.
- 252 3. The Delivery Method (i.e., the protocol) which the Printer uses to send the Event Notification.
- 253 4. Some opaque data that the Printer sends to the Notification Recipient in the Event Notification.
254 The Notification Recipient might use this opaque data as a forwarding address for the Event
255 Notification.
- 256 5. The charset to use in text fields within an Event Notification
- 257 6. The natural language to use in the text fields of the Event Notification
- 258 7. The requested lease time in seconds for the Subscription Object

259 An operation that creates a Subscription Object is called a *Subscription Creation Operation*. These
260 operations include the following operations (see section 11.1 for further details):

- 261 - **Job Creation operation:** When a client performs such an operation (Print-Job, Print-URI,
262 and Create-Job), a client can include zero or more Subscription Template Attributes Groups in
263 the request. The Printer creates one Subscription Object for each Subscription Template
264 Attributes Group in the request, and the Printer associates each such Subscription Object with
265 the newly created Job. This document extends these operations' definitions in [RFC2911] by
266 adding Subscription Template Attributes Groups in the request and Subscription Attributes
267 Groups in the response.
- 268 - **Create-Job-Subscriptions operation:** A client can include one or more Subscription
269 Template Attributes Groups in the request. The Printer creates one Subscription Object for
270 each Subscription Template Attributes Group and associates each with the job that is the
271 target of this operation.

272 - **Create-Printer-Subscriptions operation:** A client can include one or more Subscription
273 Template Attributes Groups in the request. The Printer creates one Subscription Object for
274 each Subscription Template Attributes Group and associates each with the Printer that is the
275 target of this operation.

276 For each of the above operations:

277 - the Printer associates a Subscription Object with the Printer or a specific Job. When a
278 Subscription Object is associated with a Job Object, it is called a *Per-Job Subscription Object*.
279 When a Subscription Object is associated with a Printer Object, it is called a *Per-Printer*
280 *Subscription Object*.

281 - the response contains one Subscription Attributes Group for each Subscription Template
282 Attributes Group in the request and in the same order. When the Printer successfully creates a
283 Subscription Object, its corresponding Subscription Attributes Group contains the “notify-
284 subscription-id” attribute. This attribute uniquely identifies the Subscription Object and is
285 analogous to a “job-id” for a Job object. Some operations described below use the “notify-
286 subscription-id” to identify the target Subscription Object.

287 This document defines the following additional operations (see section 11.2 for further details):

288 - **Restart-Job operation:** When a client performs the Restart-Job operation [RFC2911], the
289 Printer re-uses the same Job and its Subscription Objects.

290 - **Validate-Job operation:** When a client performs this operation, a client can include zero or
291 more Subscription Template Attributes Groups in the request. The Printer determines if it
292 could create one Subscription Object for each Subscription Template Attributes Group in the
293 request. This document extends this operation’s definition in [RFC2911] by adding
294 Subscription Template Attributes Groups in the request and Subscription Attributes Groups in
295 the response.

296 - **Get-Subscription-Attributes operation:** This operation allows a client to obtain the
297 specified attributes of a target Subscription Object.

298 - **Get-Subscriptions operation:** This operation allows a client to obtain the specified attributes
299 of all Subscription Objects associated with the Printer or a specified Job.

300 - **Renew-Subscription operation:** This operation renews the lease on the target Per-Printer
301 Subscription Object before it expires. A newly created Per-Printer Subscription Object
302 receives an initial lease. It is the duty of the client to use this operation frequently enough to
303 preserve a Per-Printer Subscription Object. The Printer deletes a Per-Printer Subscription
304 Object when its lease expires. A Per-Job Subscription Object last exactly as long as its
305 associated Job Object and thus doesn’t have a lease.

306 - **Cancel-Subscription operation:** This operation (1) cancels the lease on the specified Per-
307 Printer Subscription Object and thereby deletes the Per-Printer Subscription Object or (2)
308 deletes the Per-Job Subscription Object.

309 When an Event occurs, the Printer finds all Subscription Objects listening for the Event (see section 9
310 for details on finding such Subscription Objects). For each such Subscription Object, the Printer:

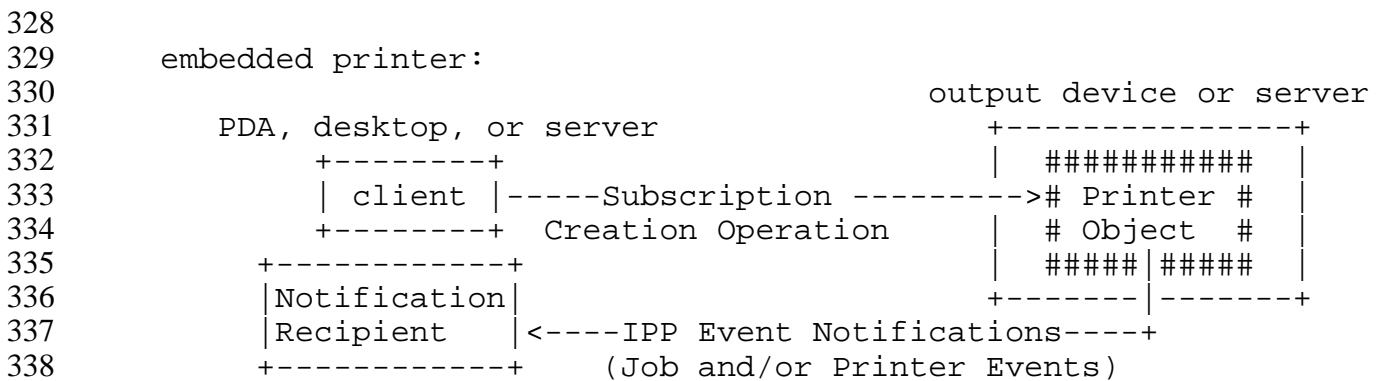
- 311 a) generates an Event Notification with information specified in section 9, AND
- 312 b) either:
 - 313 i) If the Delivery Method is a Push Delivery Method as indicated by the presence of the
314 Subscription Object’s “notify-recipient-uri” attribute, delivers the Event Notification
315 using the Delivery Method and target address identified in the Subscription Object’s
316 “notify-recipient-uri” attribute, OR
 - 317 ii) If the Delivery Method is a Pull Delivery Method as indicated by the presence of the
318 Subscription Object’s “notify-pull-method” attribute, saves Event Notification for a time
319 period called the Event Life defined by the Delivery Method, i.e., the Notification
320 Recipient is expected to fetch the Event Notifications.

321 2 Models for Notification

322 2.1 Model for Notification (Simple Case)

323 As part of a Subscription Creation Operation, an IPP Printer (i.e., located in an output device or a
324 server) creates one or more Subscription Objects. In a Subscription Creation Operation, the client
325 specifies the Notification Recipient to which the Printer is to deliver Event Notifications. A
326 Notification Recipient can be the Subscribing Client or a third party.

327 Figure 1 shows the Notification model for a simple Client-Printer relationship.



339 **Figure 1 – Model for Notification**

340 2.2 Model for Notification with Cascading Printers

341 With this model, there is an intervening Print server between the human user and the Printer in the
342 output device. If the Printer in the output device generates an Event, the system can be configured to
343 send Event Notification either

- 344 - directly to the Notification Recipient specified by the Subscribing Client or
345 - via the Print Server to the Notification Recipient specified by the Subscribing Client.
346 See Appendix A for more details.

347 2.3 Distributed Model for Notification

348 The preceding sections (2.1 and 2.2) assume that the Notification software resides in the same device
349 or Server box as the rest of the Printer software. In many implementations, the assumption is correct.
350 However, the Notification model also permits a distributed implementation.

351 For example, the software that supports both Subscription Creation Operations and sending of Event
352 Notifications could be on hardware that is separate from the output device. To make this work, there
353 must be a symbiotic relationship between the output device software and the remote Notification
354 software. Without the remote Notification software, the output device software is not a complete
355 Printer.

356 The term “Printer” in this document includes the software on the output device or server box as well as
357 Notification software that is local to or remote from the output device.

358 Appendix B describes this example in detail.

359 2.4 Extended Notification Recipient

360 The model allows for an extended Notification Recipient that is itself a Notification service that
361 forwards each Event Notification to another recipient. The client contacts this Notification Recipient
362 to arrange for forwarding by means outside the scope of this document. The Printer need not be aware
363 that the Notification Recipient forwards Event Notifications.

364 Appendix C describes this example in detail.

365 3 Terminology

366 This section defines terminology used throughout this document. Other terminology is defined in
367 [RFC2911].

368 3.1 Conformance Terminology

369 Capitalized terms, such as **MUST**, **MUST NOT**, **REQUIRED**, **SHOULD**, **SHOULD NOT**, **MAY**,
370 **NEED NOT**, and **OPTIONAL**, have special meaning relating to conformance as defined in RFC 2119
371 [RFC2119] and [RFC2911] section 12.1. If an implementation supports the extension defined in this
372 document, then these terms apply; otherwise, they do not. These terms define conformance to *this*

373 *document only*; they do not affect conformance to other documents, unless explicitly stated otherwise.
374 See Appendix D for complete details.

375 Note: a feature that is OPTIONAL in this document becomes REQUIRED if the Printer implements a
376 Delivery Method that REQUIRES the feature.

377 **READ-ONLY** – an adjective used in an attribute definition to indicate that an IPP Printer MUST NOT
378 allow the attribute's value to be modified ~~with the Set Job Attributes or Set Printer Attributes~~
379 ~~operations (see [ipp-set]). Note: there is no Set Subscription operation so this term is not used for~~
380 ~~Subscription object attributes.~~

381 3.2 Other Terminology

382 This document uses the same terminology as [RFC2911], such as “client”, “Printer”, “attribute”,
383 “attribute value”, “keyword”, “operation”, “request”, “response”, and “support”. In addition, the
384 following terms are defined for use in this document and the Delivery Method Documents:

385 **Administrator** – A human user who establishes policy for and configures the print system.

386 **Operator** – A human user who carries out the policy established by the Administrator and controls the
387 day to day running of the print system.

388 **IPP Client (or client)** – The software component (PDA, desktop, or server) that performs an IPP
389 operation directed at an IPP Printer (located in a server or output device).

390 **Job Creation operation** – One of the operations that creates a Job object: Print-Job, Print-URI and
391 Create-Job. The Restart-Job operation [RFC2911] is not considered a Job Creation operation,
392 since the Printer re-uses the existing Job object. The Validate-Job operation is not considered a
393 Job Creation operation because no Job object is created. Therefore, when a statement also applies
394 to either the Restart-Job and/or the Validate-Job operation, they are mentioned explicitly.

395 **Event** – some occurrence (either expected or unexpected) within the printing system of a change of
396 state, condition, or configuration of a Job or Printer object. An Event occurs only at one instant in
397 time and does not span the time the physical Event takes place. For example, jam-occurred and
398 jam-cleared are two distinct, instantaneous Events, even though the jam may last for a while.

399 **Event Notification** – the information about an Event that the Printer sends when an Event occurs.

400 **Compound Event Notification** – two or more Event Notifications that a Printer sends together as a
401 single entity. The Delivery Method Document specifies whether the Delivery Method supports
402 Compound Event Notifications.

403 **Job Event** – an Event caused by some change in a particular job on the Printer, e.g., 'job-completed'.

404 **Printer Event** – an Event caused by some change in the Printer that is not specific to a job, e.g.,
405 'printer-state-changed'.

- 406 **Subscribed Event** – an Event that the Subscribing Client expresses interest in by making it a value of
407 the “notify-events” attribute on a Subscription Object.
- 408 **Subscribed Job Event** – a Subscribed Event that is a Job Event.
- 409 **Subscribed Printer Event** – a Subscribed Event that is a Printer Event.
- 410 **Notification Recipient** – the entity to which the Printer sends an Event Notification.
- 411 **Delivery Method** – the mechanism by which the Printer delivers the Event Notification, e.g., via email
412 or via an Event Notification Delivery Method protocol defined for delivering IPP Event
413 Notifications.
- 414 **Delivery Method Document** – a document, separate from this document, that defines a Delivery
415 Method.
- 416 **Push Delivery Method** –The Printer sends the Event Notification shortly after an Event occurs. For
417 some Push Delivery Methods, the Notification Recipient **MUST** send a response; for others it
418 **MUST NOT** send a response.
- 419 **Pull Delivery Method** – The Printer saves Event Notifications for some event life time and expects the
420 Notification Recipient to request Event Notifications. The Printer returns the Event Notifications
421 in a response to such a request.
- 422 **Event Life** – For a Pull Delivery Method, the length of time in seconds after an Event occurs during
423 which the Printer will return that Event in response to a request for Event Notifications. After
424 the Event Life expires, the Printer will no longer return an Event Notification for that Event in
425 such a response.
- 426 **Subscription Object** – An object containing a set of attributes that indicate: the Notification Recipient,
427 the Delivery Method, the Subscribed Events that cause the Printer to send an Event Notification,
428 and the information to send in an Event Notification.
- 429 **Per-Job Subscription Object** – A Subscription Object that is associated with a single Job. The
430 Create-Job-Subscriptions operation and Job Creation operations create such an object.
- 431 **Per-Printer Subscription Object** – A Subscription Object that is associated with the Printer as a
432 whole. The Create-Printer-Subscriptions operation creates such an object.
- 433 **Subscribing Client** – The client that creates the Subscription Object.
- 434 **Subscription Creation Operation** – An operation that creates a Subscription Object: Job Creation
435 operations, Create-Job-Subscriptions operation, Create-Printer-Subscriptions operation. In the
436 context of a Job Creation operation, a Subscription Creation Operation is the part of the Job
437 Creation operation that creates a Subscription object. The Restart-Job operation [RFC2911] is not
438 considered a Subscription Creation Operation, since the Printer re-uses the Job’s existing
439 Subscription Objects, rather than creating any new Subscription Objects.

- 440 **Subscription Creation Request** – The request portion of a Subscription Creation Operation.
- 441 **Subscription Template Attributes** – Subscription Object attributes that a client can supply in a
442 Subscription Creation Operation and associated Printer Object attributes that specify supported
443 and default values for the Subscription Object attributes.
- 444 **Subscription Description Attributes** – Subscription Object attributes that a Printer supplies during a
445 Subscription Creation Operation.
- 446 **Subscription Template Attributes Group** – The attributes group in a request that contains
447 Subscription Object attributes that are Subscription Template Attributes.
- 448 **Subscription Attributes Group** – The attributes group in a response that contains Subscription Object
449 attributes.
- 450 **Human Consumable Event Notification** – localized text for human consumption only. There is no
451 standardized format and thus programs should not try to parse this text.
- 452 **Machine Consumable Event Notification** – bytes for program consumption. The bytes are formatted
453 according to the Delivery Method document.
- 454 **Printer** – the software that supports an output device or print server (see IPP/1.1 [RFC2911] which
455 uses the terms Printer and Printer object interchangeably). This document extends the IPP/1.1
456 Printer definition to include the software that implements Subscription Creation Operations and
457 the sending of Event Notifications, even if the software for such a Printer would be distributed
458 across a network (see section 2.3).
- 459 **Notification** – when not in the phrases ‘Event Notification’ and ‘Notification Recipient’ — the
460 concepts of this specification, i.e., Events, Subscription Objects, and Event Notifications.

461 **4 Object Relationships**

462 This section defines the object relationships between the Printer, Job, and Subscription Objects. It does
463 not define the implementation. For an illustration of these relationships, see Appendix E.

464 **4.1 Printer and Per-Printer Subscription Objects**

- 465 1. A Printer object can be associated with zero or more Per-Printer Subscription Objects.
- 466 2. Each Per-Printer Subscription Object is associated with exactly one Printer object.

467 **4.2 Printer, Job and Per-Job Subscription Objects**

- 468 1. A Printer object is associated with zero or more Job objects.

- 469 2. Each Job object is associated with exactly one Printer object.
- 470 3. A Job object is associated with zero or more Per-Job Subscription Objects.
- 471 4. Each Per-Job Subscription Object is associated with exactly one Job object.

472 **5 Subscription Object**

473 A Subscribing Client creates a Subscription Object with a Subscription Creation Operation in order to
474 indicate its interest in certain Events. See section 11 for a description of these operations. When an
475 Event occurs, the Subscription Object specifies to the Printer where to send Event Notifications, how
476 to send them and what to put in them. See section 9 for details on the contents of an Event
477 Notification.

478 Using the IPP Job Template attributes as a model (see [RFC2911] section 4.2), the attributes of a
479 Subscription Object are divided into two categories: Subscription Template Attributes and Subscription
480 Description Attributes.

481 Subscription Template attributes are, in turn, like the Job Template attributes, divided into

- 482 1. Subscription Object attributes that a client can supply in a Subscription Creation Request and
- 483 2. their associated Printer Object attributes that specify supported and default values for the
484 Subscription Object attributes

485 The remainder of this section specifies general rules for Subscription Template Attributes and
486 describes each attribute in a Subscription Object.

487 **5.1 Rules for Support of Subscription Template Attributes**

488 Subscription Template Attributes are fundamental to the Notification model described in this
489 specification. The client supplies these attributes in Subscription Creation Operations and the Printer
490 uses these attributes to populate a newly created Subscription Object.

491 Subscription Objects attributes that are Subscription Template Attributes conform to the following
492 rules:

- 493 1. Each attribute's name starts with the prefix string "notify-" and this document calls such
494 attributes "notify-xxx".
- 495 2. For each "notify-xxx" Subscription Object attribute defined in column 1 of Table 1 in section
496 5.3, Table 1 specifies corresponding Printer attributes: "notify-xxx-default", "notify-xxx-
497 supported", "yyy-supported" and "notify-max-xxx-supported" defined in column 2 of Table 1.
498 Note "xxx" stands for the same string in each case and "yyy" stands for some other string.

- 499 3. If a Printer supports “notify-xxx” in column 1 of Table 1, then the Printer MUST support all
500 associated attributes specified in column 2 of Table 1. For example, Table 1 shows that if the
501 Printer supports “notify-events”, it MUST support “notify-events-default”, “notify-events-
502 supported” and “notify-max-events-supported”.
- 503 4. If a Printer does not support “notify-xxx” in column 1 of Table 1, then the Printer MUST NOT
504 support any associated “notify-yyy” attributes specified in column 2 of Table 1. For example,
505 Table 1 shows that if the Printer doesn’t support “notify-events”, it MUST NOT support “notify-
506 events-default”, “notify-events-supported” and “notify-max-events-supported”. Note this rule
507 does not apply to attributes whose names do not start with the string “notify-” and are thus
508 defined in another object and used by other attributes.
- 509 5. Most “notify-xxx” attributes have a corresponding “yyy-supported” attribute that specifies the
510 supported values for “notify-xxx”. Column 2 of Table 1 specifies the name of each “yyy-
511 supported” attribute. The naming rules of IPP/1.1 (see [RFC2911]) are used when “yyy-
512 supported” is “notify-xxx-supported”.
- 513 6. Some “notify-xxx” attributes have a corresponding “notify-xxx-default” attribute that specifies
514 the value for “notify-xxx” if the client does not supply it. Column 2 of Table 1 specifies the
515 name of each “notify-xxx-default” attribute. The naming rules of IPP/1.1 (see [RFC2911]) are
516 used.
- 517 If a client wishes to present an end user with a list of supported values from which to choose, the client
518 SHOULD query the Printer for its supported value attributes. The client SHOULD also query the
519 default value attributes. If the client then limits selectable values to only those values that are
520 supported, the client can guarantee that the values supplied by the client in the create request all fall
521 within the set of supported values at the Printer. When querying the Printer, the client MAY enumerate
522 each attribute by name in the Get-Printer-Attributes Request, or the client MAY just supply the
523 ‘subscription-template’ group name in order to get the complete set of supported attributes (both
524 supported and default attributes – see section 11.2.3).

525 5.2 Rules for Processing Subscription Template Attributes

526 This section defines a detailed set of rules that a Printer follows when it processes Subscription
527 Template Attributes in a Subscription Creation Request. These rules are similar to the rules for
528 processing Operation attributes in [RFC2911]. That is, the Printer may or may not support an attribute
529 and a client may or may not supply the attribute. Some combinations of these cases are OK. Others
530 return warnings or errors, and perhaps a list of unsupported attributes.

531 A Printer MUST implement the following behavior for processing Subscription Template Attributes in
532 a Subscription Creation Request:

- 533 1. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer supports it and
534 its value, the Printer MUST populate the attribute on the created Subscription Object.

- 535 2. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer doesn’t
536 support it or its value, the Printer MUST NOT populate the attribute on the created Subscription
537 Object with it. The Printer MUST do one of the following:
- 538 a) If the value of the “notify-xxx” attribute is unsupported, the Printer MUST return the attribute
539 with its value in the Subscription Attributes Group of the response.
- 540 b) If “notify-xxx” is an unsupported attribute, the Printer MUST return the attribute in the
541 Subscription Attributes Group of the response with the ‘unsupported’ out-of-band value.
- 542 Note: The rules of this step are the same as for Unsupported Attributes [RFC2911] section 3.1.7.
543 except that the unsupported attributes are returned in the Subscription Attributes Group rather than
544 the Unsupported Attributes Group because Subscription Creation Operations can create more than
545 one Subscription Object).
- 546 3. If a client is REQUIRED to supply a “notify-xxx” attribute from column 1 of Table 1 and the
547 Printer doesn’t support the supplied value, the Printer MUST NOT create a Subscription Object.
548 The rules for Unsupported Attributes in step #2 still apply.
- 549 4. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 and the attribute is
550 REQUIRED for the client to supply, the Printer MUST reject the Subscription Creation Operation
551 (including Job Creation operations) without creating a Subscription Object, and MUST return in
552 the response:
- 553 c) the status code ‘client-error-bad-request’ AND
- 554 d) no Subscription Attribute Groups.
- 555 5. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 that is OPTIONAL for
556 the client to supply, and column 2 of Table 1 either:
- 557 a) specifies a “notify-xxx-default” attribute, the Printer MUST behave as if the client had supplied
558 the “notify-xxx-default” attribute (see step #1) and populate the Subscription object with the
559 value of the “notify-xxx-default” attribute as part of the Subscription Creation operation (unlike
560 Job Template attributes where the Printer does not populate the Job object with defaults – see
561 [RFC2911]) OR
- 562 b) does not specify a “notify-xxx-default” attribute, the Printer MUST populate the “notify-xxx”
563 attribute on the Subscription Object according to the definition of the “notify-xxx” attribute in a
564 section 5.3. For some attributes, the “notify-xxx” is populated with the value of some other
565 attribute, and for others, the “notify-xxx” is NOT populated on the Subscription object at all.
- 566 6. A Printer MUST create a Subscription Object for each Subscription Template Attributes group in a
567 request unless the Printer:
- 568 a) encounters some attributes in a Subscription Template Attributes Group that require the Printer
569 not to create the Subscription Object OR

- 570 b) would create a Per-Job Subscription Object when it doesn't have space for another Per-Job
571 Subscription Object OR
- 572 c) would create a Per-Printer Subscription Object when it doesn't have space for another Per-
573 Printer Subscription Object.
- 574 7. A response MUST contain one Subscription Attributes Group for each Subscription Template
575 Attributes Group in the request (and in the same order) whether the Printer creates a Subscription
576 Object from the Subscription Template Attributes Group or not. However, the attributes in each
577 Subscription Attributes Group can be in any order.
- 578 8. The Printer MUST populate each Subscription Attributes Group of the response such that each
579 contains:
- 580 a) the "notify-subscription-id" attribute (see section 5.4.1), if and only if the Printer creates a
581 Subscription Object.
- 582 b) the "notify-lease-duration" attribute (see section 5.3.8), if and only if the Printer creates a Per-
583 Printer Subscription Object. The value of this attribute is the value of the Subscription Object's
584 "notify-lease-duration" attribute. This value MAY be different from the client-supplied value
585 (see section 5.3.8). If a client supplies this attribute in the creation of a Per-Job Subscription
586 Object, it MUST appear in this group with the out-of-band value 'unsupported' to indicate that
587 the Printer doesn't support it in this context.
- 588 c) all of the unsupported Subscription Template Attributes from step #2. Note, they are not
589 returned in the Unsupported Attributes Group in order to separate the unsupported attributes for
590 each Subscription Object.
- 591 d) the "notify-status-code" attribute if the Printer does not create the Subscription Object or if
592 there are unsupported attributes from step #2. The possible values of the "notify-status-code"
593 attribute are shown below (see section 13 for more details). The Printer returns the first value
594 in the list below that describes the status.
- 595 'client-error-uri-scheme-not-supported': the Subscription Object was not created because
596 the scheme of the "notify-recipient-uri" attribute is not supported. See section 13.1 for
597 more details about this status code. See step #3 in this section for the case that causes
598 this error, and the resulting step #6a) that causes the Printer not to create the
599 Subscription Object.
- 600 'client-error-attributes-or-values-not-supported': the Subscription Object was not created
601 because the method of the "notify-pull-method" attribute is not supported. See section
602 13.1 for more details about this status code. See step #3 in this section for the case that
603 causes this error, and the resulting step #6a) that causes the Printer not to create the
604 Subscription Object.
- 605 'client-error-too-many-subscriptions': the Subscription Object was not created because the
606 Printer has no space for additional Subscription Objects. The client SHOULD try again

607 later. See section 13.3 for more details about this status code. See steps #6b) and #6c)
608 in this section for the cases that causes this error.

609 ‘successful-ok-too-many-events’: the Subscription Object was created without the “notify-
610 events” values included in this Subscription Attributes Group because the “notify-
611 events” attribute contains too many values. See section 13.4 for more details about this
612 status code. See step #2 in this section and section 5.3.3 for the cases that cause this
613 status code.

614 ‘successful-ok-ignored-or-substituted-attributes’: the Subscription Object was created but
615 some supplied Subscription Template Attributes are unsupported. These unsupported
616 attributes are also in the Subscription Attributes Group. See section 13.5 for more
617 details about this status code. See step #2 in this section for the cases that cause this
618 status code.

619 9. The Printer MUST validate all Subscription Template Attributes and MUST return all unsupported
620 attributes and values in the corresponding Subscription Attributes Group of the response (see step
621 #2) unless it determines that it could not create additional Subscription Objects because of
622 condition #6b) or condition #6c). Then, the Printer NEED NOT validate these additional
623 Subscription Template Attributes and the client MUST NOT expect to find unsupported attributes
624 from step #2 in such additional Subscription Attribute Groups.

625 5.3 Subscription Template Attributes

626 This section contains the Subscription Template Attributes defined for the Subscription and Printer
627 objects.

628 Table 1 below shows the Subscription Template Attributes and has two columns:

- 629 - **Attribute in Subscription Object:** the name and attribute syntax of each Subscription Object
630 Attribute that is a Subscription Template Attribute
- 631 - **Default and Supported Printer Attributes:** the default attribute and supported Printer
632 attributes that are associated with the attribute in column 1.

633 The “notify-recipient-uri” attribute is for use with Push Delivery Methods. The “notify-pull-method”
634 attribute is for use with Pull Delivery Methods.

635 For Push Delivery Methods, a Printer MUST support all attributes in Table 1 below except for “notify-
636 pull-method” and “notify-attributes” (and “notify-pull-method-supported” and “notify-attributes-
637 supported”). For Pull Delivery Methods, a Printer MUST support all attributes in Table 1 below
638 except for “notify-recipient-uri” and “notify-attributes” (and “notify-schemes-supported” and “notify-
639 attributes-supported”). If a Printer supports both Push and Pull Delivery Methods, then it MUST
640 support both “notify-recipient-uri” and “notify-pull-method” attributes.

641 For Pull Delivery Methods, a client MUST supply “notify-recipient-uri” and MAY omit any of the rest
642 of the attributes in column 1 of Table 1 in a Subscription Creation Request. For Push Delivery

643 Methods, a client MUST supply “notify-pull-method” and MAY omit any of the rest of the attributes in
 644 column 1 of Table 1 in a Subscription Creation Request. A client MUST NOT supply both “notify-
 645 recipient-uri” and “notify-pull-method” attributes in the same Subscription Creation Request.

646 Note: The Default and Supported Printer attributes listed in column 2 of Table 1 do not have separate
 647 sections in this specification defining their semantics. Instead, the section for the corresponding
 648 Subscription Object attribute (column 1 of Table 1) contains the semantics of these Printer attributes.
 649 This approach follows the precedence of the Job Template attributes in section 4.2 of [RFC2911]
 650 where the corresponding “xxx-default” and “xxx-supported” Printer attributes are defined in the same
 651 section as the “xxx” Job attribute.

652 **Table 1 – Subscription Template Attributes**

| Attribute in Subscription Object | Default and Supported Printer Attributes |
|---|--|
| notify-recipient-uri (uri) * | notify-schemes-supported (1setOf uriScheme) |
| notify-pull-method (type2 keyword) ** | notify-pull-method-supported (1setOf type2 keyword) |
| notify-events (1setOf type2 keyword) | notify-events-default (1setOf type2 keyword) notify-events-supported (1setOf type2 keyword) notify-max-events-supported (integer(2:MAX)) |
| notify-attributes (1setOf type2 keyword) | notify-attributes-supported (1setOf type2 keyword) |
| notify-user-data (octetString(63)) | |
| notify-charset (charset) | charset-supported (1setOf charset) |
| notify-natural-language (naturalLanguage) | generated-natural-language-supported (1setOf naturalLanguage) |
| notify-lease-duration (integer(0:MAX)) | notify-lease-duration-default (integer(0:67108863)) notify-lease-duration-supported (1setOf (integer(0: 67108863) rangeOfInteger(0:67108863))) |
| notify-time-interval (integer(0:MAX)) | |

653 * “notify-recipient-uri” is for Push Delivery Methods only.

654 ** “notify-pull-method” is for Pull Delivery Methods only.

655 **~~5.3.1 notify-recipient-uri (uri) OR notify-pull-method (type2 keyword)~~**

656 ~~The “notify-recipient-uri” attribute MUST be used for Push Delivery Methods and the “notify-pull-~~
 657 ~~method” attribute MUST be used for Pull Delivery Methods.~~

658 **~~5.3.1.15.3.1~~ notify-recipient-uri (uri)**

659 This attribute’s value is a URL, which is a special case of a URI. Its value consists of a scheme and an
 660 address. The address specifies the Notification Recipient and the scheme specifies the Push Delivery
 661 Method for each Event Notification associated with this Subscription Object.

662 If a Printer supports any Push Delivery Methods, a Printer MUST support this attribute and return the
663 value as supplied by the client (no case conversion or other canonicalization) in any operation response
664 that includes this attribute.

665 For a Push Delivery Method, a client MUST supply this attribute in a Subscription Creation Operation.
666 Thus there is no need for a default Printer attribute.

667 The URI scheme of the value of this attribute on a Subscription object MUST be a value of the “notify-
668 schemes-supported (1setOf uriScheme)” Printer attribute. Note: According to [RFC2396] the “:”
669 terminates the scheme and so is not part of the scheme. Therefore, values of the “notify-schemes-
670 supported” Printer attribute do not include the “:” character.

671 If the client supplies an unsupported scheme in the value of this attribute, then the Printer MUST NOT
672 create the Subscription Object and MUST return the “notify-status-code” attribute with the ‘client-
673 error-uri-scheme-not-supported’ value in the Subscription Attributes Group in the response.

674 The Printer MUST treat the address part of this attribute as opaque.

675 **5.3.1.25.3.2 notify-pull-method (type2 keyword)**

676 This attribute’s value is a type2 keyword indicating which Pull Delivery Method is to be used.

677 ~~If~~ Since a Printer **MUST** supports ~~any~~ the ‘ippget’ Pull Delivery Methods [ipp-get-method] (see section
678 15), a Printer MUST support this attribute and return the value as supplied by the client in any
679 operation response that includes this attribute.

680 For a Pull Delivery Method, a client MUST supply this attribute in a Subscription Creation Operation.
681 Thus there is no need for a default Printer attribute.

682 The keyword value of this attribute on a Subscription object MUST be a value of the “notify-pull-
683 method-supported (1setOf type2 keyword)” Printer attribute.

684 If the client supplies an unsupported method in the value of this attribute, then the Printer MUST NOT
685 create the Subscription Object and MUST return the “notify-status-code” attribute with the ‘client-
686 error-attributes-or-values-not-supported’ value in the Subscription Attributes Group in the response.

687 **5.3.3 notify-events (1setOf type2 keyword)**

688 This attribute contains a set of Subscribed Events. When an Event occurs and it “matches” a value of
689 this attribute, the Printer sends an Event Notification using information in the Subscription Object.
690 The details of “matching” are described subsection 5.3.3.2.

691 A Printer MUST support this attribute.

692 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
693 this attribute in Subscription Creation Operation, the Printer MUST populate this attribute on the
694 Subscription Object with its “notify-events-default” attribute value.

695 Each keyword value of this attribute on a Subscription Object MUST be a value of the “notify-events-
696 supported (1setOf type2 keyword)” Printer attribute.

697 The number of values of this attribute MUST NOT exceed the value of the “notify-max-events-
698 supported” attribute. A Printer MUST support at least 2 values per Subscription Object. If the number
699 of values supplied by a client in a Subscription Creation Operation exceeds the value of this attribute,
700 the Printer MUST treat extra values as unsupported values and MUST use the value of ‘successful-ok-
701 too-many-events’ for the “notify-status-code” attribute in the Subscription Attributes Group of the
702 response.

703 5.3.3.1 Standard Values for Subscribed Events

704 Each value of this attribute is a keyword and it specifies a Subscribed Event that represents certain
705 changes. Some keywords represent a subset of changes of another keyword, e.g., ‘job-completed’ is an
706 Event value which is a sub-value of ‘job-state-change’. See section 5.3.3.2 for the case where this
707 attribute contains both a value and a sub-value.

708 The values in this section are divided into three categories: No Events, Job Events and Printer Events.

709 A Printer MUST support the Events indicated as “REQUIRED” and MAY support the Events
710 indicated as “OPTIONAL”.

711 5.3.3.1.1 No Events

712 The standard and only keyword value for No Events is:

713 **‘none’**: REQUIRED – no Event Notifications for any Events. As the sole value of “notify-events-
714 supported”, this value means that the Printer does not support the sending of Event Notifications.
715 As the sole value of “notify-events-default”, this value means that a client MUST specify the
716 “notify-events” attribute in order for a Subscription Creation Operation to succeed. If the Printer
717 receives this value as the sole value of a Subscription Creation Operation, it does not create a
718 Subscription Object. If a Printer receives this value with other values of a Subscription Creation
719 Operation, the Printer MUST treat this value as an unsupported value.

720 5.3.3.1.2 Subscribed Printer Events

721 The standard keyword values for Subscribed Printer Events are:

722 **‘printer-state-changed’**: REQUIRED – the Printer changed state from any state to any other state.
723 Specifically, the value of the Printer’s “printer-state”, “printer-state-reasons” or “printer-is-
724 accepting-jobs” attributes changed.

725

726

This Subscribed Event value has the following sub-values: ‘printer-restarted’ and ‘printer-shutdown’. A client can listen for any of these sub-values if it doesn’t want to listen to all printer-state changes:

727

728

729

‘printer-restarted’: OPTIONAL – when the printer is powered up .

730

‘printer-shutdown’: OPTIONAL – when the device is being powered down .

731

732

‘printer-stopped’: REQUIRED – when the printer stops printing, i.e. the value of the “printer-state” Printer attribute becomes ‘stopped’.

733

734

‘printer-config-changed’: OPTIONAL – when the configuration of a Printer has changed, i.e., the value of the “printer-message-from-operator” or any “configuration” Printer attribute has changed. A “configuration” Printer attribute is an attribute which can change value because of some human interaction either direct or indirect, and which is not covered by one of the other Events in this section. Examples of “configuration” Printer attributes are any of the Job Template attributes, such as “xxx-supported”, “xxx-ready” and “xxx-default”. ~~Often, such a change is the result of a client performing a Set-Printer-Attributes operation (see [ipp-set]) on the Printer.~~ The client has to perform a Get-Printer-Attributes to find out the new values of these changed attributes. This Event is useful for GUI clients and drivers to update the available printer capabilities to the user.

735

736

737

738

739

740

741

742

743

This Event value has the following sub-values: ‘printer-media-changed’ and ‘printer-finishings-changed’. A client can listen for any of these sub-values if it doesn’t want to listen to all printer-configuration changes:

744

745

746

747

‘printer-media-changed’: OPTIONAL – when the media loaded on a printer has been changed, i.e., the “media-ready” attribute has changed. This Event includes two cases: an input tray that goes empty and an input tray that receives additional media of the same type or of a different type. The client must check the “media-ready” Printer attribute (see [RFC2911] section 4.2.11) separately to find out what changed.

748

749

750

751

752

‘printer-finishings-changed’: OPTIONAL – when the finisher on a printer has been changed, i.e., the “finishings-ready” attribute has changed. This Event includes two cases: a finisher that goes empty and a finisher that is refilled (even if it is not full). The client must check the “finishings-ready” Printer attribute separately to find out what changed.

753

754

755

756

‘printer-queue-order-changed’: OPTIONAL – the order of jobs in the Printer’s queue has changed, so that an application that is monitoring the queue can perform a Get-Jobs operation to determine the new order. This Event does not include when a job enters the queue (the ‘job-created’ Event covers that) and does not include when a job leaves the queue (the ‘job-completed’ Event covers that).

757

758

759

760

761 **5.3.3.1.3 Subscribed Job Events**

762 The standard keyword values for Subscribed Job Events are:

763 **‘job-state-changed’**: REQUIRED – the job has changed from any state to any other state.

764 Specifically, the Printer sends this Event whenever the value of the “job-state” attribute or “job-
765 state-reasons” attribute changes. When a Job is removed from the Job Retention or Job History
766 phases (see [RFC2911] section 4.3.7.1), no Event is generated.

767

768 This Event value has the following sub-values: ‘job-created’, ‘job-completed’ and ‘job-stopped’.
769 A client can listen for any of these sub-values if it doesn’t want to listen to all ‘job-state changes’.

770 **‘job-created’**: REQUIRED – the Printer has accepted a Job Creation operation, a Restart-
771 Job operation [RFC2911], or any job operation that creates a Job object from an existing
772 Job object. The Printer sets-populates the job’s “time-at-creation” attribute value (see
773 [RFC2911] section 4.3.14.1). The Printer puts the job in the ‘pending’, ‘pending-held’
774 or ‘processing’ states.

775 **‘job-completed’**: REQUIRED – the job has reached one of the completed states, i.e., the
776 value of the job’s “job-state” attribute has changed to: ‘completed’, ‘aborted’, or
777 ‘canceled’. The Job’s “time-at-completed” and “date-time-at-completed” (if supported)
778 attributes are set (see [RFC2911] section 4.3.14). When a Job completes, a Notification
779 Recipient MAY query the Job using the Get-Job-Attributes operation. To allow such a
780 query, the Printer retains the Job in the Job Retention and/or the Job History phases (see
781 [RFC2911] section 4.3.7.1) for a suitable amount of time that depends on
782 implementation and the Delivery Methods supported. The Printer also sends this Event
783 when a Job is removed with the Purge-Job operation (see [RFC2911] section 3.2.9). In
784 this case, the Event Notification MUST report the ‘job-state’ as ‘canceled’ and the Job
785 object is no longer present for query.

786 **‘job-stopped’**: OPTIONAL – when the job stops printing, i.e. the value of the “job-state”
787 Job attribute becomes ‘processing-stopped’.

788 **‘job-config-changed’**: OPTIONAL – when the configuration of a job has changed, i.e., the value of
789 the “job-message-from-operator” or any of the “configuration” Job attributes have changed. A
790 “configuration” Job attribute is an attribute that can change value because of some human
791 interaction either direct or indirect. Examples of “configuration” Job attributes are any of the job
792 template attributes and the “job-name” attribute. ~~Often, such a change is the result of the user or
793 the Operator performing a Set Job Attributes operation (see [ipp-set]) on the Job object.~~ The
794 client performs a Get-Job-Attributes to find out the new values of the changed attributes. This
795 Event is useful for GUI clients and drivers to update the job information to the user.

796 **‘job-progress’**: OPTIONAL – when the Printer has completed Printing a sheet. See the separate [ipp-
797 prog] specification for additional attributes that a Printer MAY send in an Event Notification
798 caused by this Event. The “notify-time-interval” attribute affects this Event by causing the Printer

799 NOT to send an Event Notification every time a ‘job-progress’ Events occurs. See section 5.3.9
800 for full details.

801 5.3.3.2 Rules for Matching of Subscribed Events

802 When an Event occurs, the Printer MUST find each Subscription object whose “notify-events”
803 attribute “matches” the Event. The rules for “matching” of Subscribed Events are described separately
804 for Printer Events and for Job Events. This section also describes some special cases.

805 5.3.3.2.1 Rules for Matching of Printer Events

806 Suppose that the Printer causes Printer Event E to occur. For each Per-Job or Per-Printer Subscription
807 S in the Printer, if E equals a value of this attribute in S or E is a sub-value of a value of this attribute in
808 S, the Printer MUST generate an Event Notification.

809 Consider the example. There are three Subscription Objects each with the Subscribed Printer Event
810 ‘printer-state-changed’. Subscription Object A is a Per-Printer Subscription Object. Subscription
811 Object B is a Per-Job Subscription Object for Job 1, and Subscription Object C is a Per-Job
812 Subscription Object for Job 2. When the Printer enters the ‘stopped’ state, the Printer sends an Event
813 Notification to the Notification Recipients of Subscription Objects A, B, and C because this is a Printer
814 Event. Note if Job 1 has already completed, the Printer would not send an Event Notification for its
815 Subscription Object, even if Job 1 is retained in the Job Retention and/or the Job History phases (see
816 [RFC2911] section 4.3.7.1).

817 5.3.3.2.2 Rules for Matching of Job Events

818 Suppose that Job J causes Job Event E to occur.

- 819 1. For each Per-Printer Subscription S in the Printer, if E equals a value of this attribute in S or E is
820 a sub-value of a value of this attribute in S, the Printer MUST generate an Event Notification.
- 821 2. For each Per-Job Subscription S associated with Job J, if E equals a value of this attribute in S
822 or E is a sub-value of a value of this attribute in S, the Printer MUST generate an Event
823 Notification.
- 824 3. For each Per-Job Subscription S that is NOT associated Job J, if E equals a value of this
825 attribute in S or E is a sub-value of a value of this attribute in, the Printer MUST NOT generate
826 an Event Notification from S.

827 Consider the example: There are three Subscription Objects listening for the Job Event ‘job-
828 completed’. Subscription Object A is a Per-Printer Subscription Object. Subscription Object B is a
829 Per-Job Subscription Object for Job 1, and Subscription Object C is a Per-Job Subscription Object for
830 Job 2. In addition, Per-Printer Subscription Object D is listening for the Job Event ‘job-state-changed’.
831 When Job 1 completes, the Printer sends an Event Notification to the Notification Recipient of

832 Subscription Object A (because it is Per-Printer) and Subscription Object B because it is a Per-Job
833 Subscription Object associated with the Job generating the Event. The Printer also sends an Event
834 Notification to the Notification Recipient of Subscription Object D because ‘job-completed’ is a sub-
835 value of ‘job-state-changed’ – the value that Subscription Object D is listening for. The Printer does
836 not send an Event Notification to the Notification Recipients of Subscription Object C because it is a
837 Per-Job Subscription Object associated with some Job other than the Job generating the Event.

838 5.3.3.2.3 Special Cases for Matching Rules

839 This section contains rule for special cases.

840 If an Event matches Subscribed Events in two different Subscription Objects and the Printer would
841 send two identical Event Notifications (except for the “notify-subscription-id” attribute) to the same
842 Notification Recipient using the same Delivery Method, the Printer MUST send both Event
843 Notifications. That is, the Printer MUST NOT try to consolidate seemingly identical Event
844 Notifications that occur in separate Subscription objects. Incidentally, the Printer MUST NOT reject
845 Subscription Creation Operations that would create this scenario.

846 If an Event matches two values of this “notify-events” attribute in a single Subscription object (e.g., a
847 value and its sub-value), a Printer MAY send one Event Notification for each matched value in the
848 Subscription Object or it MAY send only one Event Notification per Subscription Object. The rules in
849 sections 5.3.3.2.1 and 5.3.3.2.2 are purposefully ambiguous about the number of Event Notification
850 sent when Event E matches two or more values in a Subscription Object.

851 Consider the example: There are two Per-Printer Subscription Objects when a Job completes.
852 Subscription Object A has the Subscribed Job Event ‘job-state-changed’. Subscription Object B has
853 the Subscribed Job Events ‘job-state-changed’ and ‘job-completed’. The Printer sends an Event
854 Notification to the Notification Recipient of Subscription Object A with the value of ‘job-state-
855 changed’ for the “notify-subscribing-event” attribute. The Printer sends either one or two Event
856 Notifications to the Notification Recipient of Subscription Object B, depending on implementation. If
857 it sends two Event Notifications, one has the value of ‘job-state-changed’ for the “notify-subscribing-
858 event” attribute, and the other has the value of ‘job-completed’ for the “notify-subscribing-event”
859 attribute. If it sends one Event Notification, it has the value of either ‘job-state-changed’ or ‘job-
860 completed’ for the “notify-subscribing-event” attribute, depending on implementation. The algorithm
861 for choosing such a value is implementation dependent.

862 5.3.4 notify-attributes (1setOf type2 keyword)

863 This attribute contains a set of attribute names. When a Printer sends a Machine Consumable Event
864 Notification, it includes a fixed set of attributes (see section 9.1). If this attribute is present and the
865 Event Notification is Machine Consumable, the Printer also includes the attributes specified by this
866 attribute.

867 A Printer MAY support this attribute.

868 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
869 this attribute in Subscription Creation Operation or the Printer does not support this attribute, the
870 Subscription Object either (1) MAY contain the “notify-attributes” attribute with a ‘none’ value or (2)
871 NEED NOT contain the attribute at all. There is no “notify-attributes-default” Printer attribute.

872 Each keyword value of this attribute on a Subscription Object MUST be a value of the “notify-
873 attributes-supported (1setOf type2 keyword)” Printer attribute. The “notify-attributes-supported” MAY
874 contain any Printer attribute, Job attribute or Subscription Object attribute that the Printer supports in
875 an Event Notification. It MUST NOT contain any of the attributes in Section 9.1 that a Printer
876 automatically puts in an Event Notification; it would be redundant. If a client supplies an attribute in
877 Section 9.1, the Printer MUST treat it as an unsupported attribute value of the “notify-attributes”
878 attribute.

879 The following rules apply to each keyword value N of the “notify-attributes” attribute: If the value N
880 names:

- 881 a) a Subscription attribute, the Printer MUST use the attribute N in the Subscription Object that is
882 being used to generate the Event Notification.
- 883 b) a Job attribute and the Printer is generating an Event Notification from a Per-Job Subscription
884 Object S, the Printer MUST use the attribute N in the Job object associated with S.
- 885 c) a Job attribute and the Printer is generating an Event Notification from a Per-Printer Subscription
886 Object and the Event is:
- 887 • a Job Event, the Printer MUST use the attribute N in the Job object that caused the Event.
 - 888 • a Printer Event, the Printer MUST use the attribute N in the active Job.

889 If a Printer supports this attribute and a Subscription Object contains this attribute and the Delivery
890 Method generates a Machine Consumable Event Notification, the Printer MUST include in each Event
891 Notification:

- 892 a) the attributes specified in section 9.1 and
893 b) each attribute named by this attribute.

894 The Printer MUST NOT use this attribute to generate a Human Consumable Event Notification.

895 **5.3.5 notify-user-data (octetString(63))**

896 This attribute contains opaque data that some Delivery Methods include in each Machine Consumable
897 Event Notification. The opaque data might contain, for example:

- 898 - the identity of the Subscriber
- 899 - a path or index to some Subscriber information

900 - a key that identifies to the Notification Recipient the ultimate recipient of the Event
901 Notification

902 - the id for a Notification Recipient that had previously registered with an Instant Messaging
903 Service

904 A Printer MUST support this attribute.

905 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
906 this attribute in the Subscription Creation Operation, the Subscription Object either (1) MAY contain
907 the “notify-user-data” attribute with a zero length value or (2) NEED NOT contain the attribute at all.
908 There is no “notify-user-data-default” Printer attribute.

909 There is no “notify-user-data-supported” Printer attribute. Rather, any octetString whose length does
910 not exceed 63 octets is a supported value. If the length exceeds 63 octets, the Printer MUST treat it as
911 an unsupported value.

912 **5.3.6 notify-charset (charset)**

913 This attribute specifies the charset to be used in the Event Notification content sent to the Notification
914 Recipient, whether the Event Notification content is Machine Consumable or Human Consumable.

915 A Printer MUST support this attribute.

916 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
917 this attribute in Subscription Creation Operation or supplies an unsupported value, the Printer MUST
918 populate this attribute in the Subscription Object with the value of the “attributes-charset” operation
919 attribute, which is a REQUIRED attribute in all IPP requests (see [RFC2911]). If the value of the
920 “attributes-charset” attribute is unsupported, the Printer MUST populate this attribute in the
921 Subscription Object with the value of the Printer’s “charset-configured” attribute. There is no “notify-
922 charset-default” Printer attribute.

923 The value of this attribute on a Subscription Object MUST be a value of the “charset-supported (1setOf
924 charset)” Printer attribute.

925 **5.3.7 notify-natural-language (naturalLanguage)**

926 This attribute specifies the natural language to be used in any human consumable text in the Event
927 Notification content sent to the Notification Recipient, whether the Event Notification content is
928 Machine Consumable or Human Consumable.

929 A Printer MUST support this attribute.

930 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
931 this attribute in Subscription Creation Operation or supplies an unsupported value, the Printer MUST
932 populate this attribute in the Subscription Object with the value of the “attributes-natural-language”

933 operation attribute, which is a REQUIRED attribute in all IPP requests (see [RFC2911]). If the value
934 of the “attributes-natural-language” attribute is unsupported, the Printer MUST populate this attribute
935 in the Subscription Object with the value of the Printer’s “natural-language-configured” attribute.
936 There is no “notify-natural-language-default” Printer attribute.

937 The value of this attribute on a Subscription Object MUST be a value of the “generated-natural-
938 language-supported (1setOf type2 naturalLanguage)” Printer attribute.

939 **5.3.8 notify-lease-duration (integer(0:67108863))**

940 This attribute specifies the duration of the lease (in seconds) associated with the Per-Printer
941 Subscription Object at the time the Subscription Object was created or the lease was renewed. The
942 duration of the lease is infinite if the value is 0, i.e., the lease never expires. See section 5.4.3 on
943 “notify-lease-expiration-time (integer(0:MAX))” for more details.

944 This attribute is not present on a Per-Job Subscription Object because the Subscription Object lasts
945 exactly as long as the associated Job object. See discussion of the ‘job-completed’ event in section
946 5.3.3.1.3 about retention of the Job object after completion.

947 A Printer MUST support this attribute.

948 For a Subscription Object Creation operation of a Per-Job Subscription Object, the client MUST NOT
949 supply this attribute. If the client does supply this attribute, the Printer MUST treat it as an
950 unsupported attribute.

951 For a Subscription Creation Operation of a Per-Printer Subscription Object or a Renew-Subscription
952 operation, a client MAY supply this attribute. If the client does not supply this attribute, the Printer
953 MUST populate this attribute with its “notify-lease-duration-default” (0:67108863) attribute value. If
954 the client supplies this attribute with an unsupported value, the Printer MUST populate this attribute
955 with a supported value, and this value SHOULD be as close as possible to the value requested by the
956 client. Note: this rule implies that a Printer doesn’t assign the value of 0 (infinite) unless the client
957 requests it.

958 After the Printer has populated this attribute with a supported value, the value represents the “granted
959 duration” of the lease in seconds and the Printer [sets updates](#) the value of the Subscription Object’s
960 “notify-lease-expiration-time” attribute as specified in section 5.4.3.

961 The value of this attribute on a Subscription Object MUST be a value of the “notify-lease-duration-
962 supported” (1setOf (integer(0:67108863) | rangeOfInteger(0:67108863))) Printer attribute.

963 A Printer MAY require authentication in order to return the value of 0 (the lease never expires) as one
964 of the values of “notify-lease-duration-supported”, and to allow 0 as a value of the “notify-lease-
965 duration” attribute.

966 Note: The maximum value 67,108,863 is 2 raised to the 26 power minus 1 and is about 2 years in
967 seconds. The value is considerably less than MAX so that there is virtually no chance of an overflow
968 when it is added to “printer-up-time” to produce “notify-lease-expiration-time”.

969 **5.3.9 notify-time-interval (integer(0:MAX))**

970 The ‘job-progress’ Event occurs each time that a Printer completes a sheet. Some Notification
971 Recipients do not want to receive an Event Notification every time this Event occurs. This attribute
972 allows a Subscribing Client to request how often it wants to receive Event Notifications for ‘job-
973 progress’ Events. The value of this attribute MAY be any nonnegative integer (0,MAX) indicating the
974 minimum number of seconds between ‘job-progress’ Event Notifications.

975 The Printer MUST support this attribute if and only if the Printer supports the ‘job-progress’ Event.

976 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
977 this attribute in the Subscription Creation Operation, the Subscription Object either (1) MAY contain
978 the “notify-time-interval” attribute with a ‘0’ value or (2) NEED NOT contain this attribute at all.
979 There is no “notify-time-interval-default” Printer attribute.

980 There is no “notify-time-interval-supported” Printer attribute.

981 If the ‘job-progress’ Event occurs and a Subscription Object contains the ‘job-progress’ Event as a
982 value of the ‘notify-events’ attribute, there are two cases to consider:

- 983 1. This attribute is not present on the Subscription Object or has the value of 0. The Printer MUST
984 generate and send an Event Notification (as is the case with other Events).
- 985 2. This attribute is present with a nonzero value of N:
 - 986 a) If the Printer has not sent an Event Notification for the ‘job-progress’ Event for the associated
987 Subscription Object within the past N seconds, the Printer MUST send an Event Notification for
988 the Event that just occurred. Note when the Printer completes the first page of a Job, this rule
989 implies that the Printer sends an Event Notification for a Per-Job Subscription Object.
 - 990 b) Otherwise, the Printer MUST NOT generate or send an Event Notification for the associated
991 Subscription Object. The Printer MUST NOT increase the value of the “notify-sequence-
992 number” Subscription Object attribute (i.e., the sequence of values of the “notify-sequence-
993 number” attribute counts the Event Notifications that the Printer sent and not the Events that do
994 not cause an Event Notification to be sent).

995 It is RECOMMENDED that a Subscribing Client use this attribute when it subscribes to the ‘job-
996 progress’ Event, and that the value be sufficiently large to limit the frequency with which the Printer
997 sends Event Notifications requests.

998 This attribute MUST NOT effect any Events other than ‘job-progress’.

999 5.4 Subscription Description Attributes

1000 Subscription Description Attributes are those attributes that a Printer adds to a Subscription Object at
1001 the time of its creation.

1002 A Printer MUST support all attributes in this Table 2.

1003 A client MUST NOT supply the attributes in Table 2 in a Subscription Template Attributes Group of a
1004 Subscription Creation Operation. ~~If the client supplies them, the Printer MUST NOT set them and~~
1005 ~~MUST treat them as unsupported attributes.~~ There are no corresponding default or supported
1006 attributes.

1007 **Table 2 – Subscription Description Attributes**

| |
|---|
| Subscription Object attributes: |
| notify-subscription-id (integer(1:MAX)) |
| notify-sequence-number (integer(0:MAX)) |
| notify-lease-expiration-time (integer(0:MAX)) |
| notify-printer-up-time (integer(1:MAX)) |
| notify-printer-uri (uri) |
| notify-job-id (integer(1:MAX)) |
| notify-subscriber-user-name (name(MAX)) |

1008

1009 5.4.1 notify-subscription-id (integer (1:MAX))

1010 This attribute identifies a Subscription Object instance with a number that is unique within the context
1011 of the Printer. The Printer generates this value at the time it creates the Subscription Object.

1012 A Printer MUST support this attribute.

1013 The Printer MAY assign the value of this attribute sequentially as it creates Subscription Objects.
1014 However, if there is no security on Subscription objects, sequential assignment exposes the system to a
1015 passive traffic monitoring threat.

1016 The Printer SHOULD avoid re-using recent values of this attribute during continuous operation of the
1017 Printer as well as across power cycles. Then a Subscribing Client is unlikely to find that a stale
1018 reference accesses a new Subscription Object.

1019 The 0 value is not permitted in order to allow for compatibility with “job-id” and with SNMP index
1020 values, which also cannot be 0.

1021 5.4.2 notify-sequence-number (integer (0:MAX))

1022 The value of this attribute indicates the number of times that the Printer has generated and attempted to
1023 send an Event Notification for this Subscription object. When an Event Notification contains this
1024 attribute, the Notification Recipient can determine whether it missed some Event Notifications (i.e.,
1025 numbers skipped) or received duplicates (i.e., same number twice).

1026 A Printer MUST support this attribute.

1027 When the Printer creates a Subscription Object, it MUST ~~set-populate the value of~~ this attribute ~~to~~ with
1028 a value of 0. This value indicates that the Printer has not sent any Event Notifications for this
1029 Subscription Object.

1030 Each time the Printer sends a newly generated Event Notification, it MUST increase the value of this
1031 attribute by 1. For some Delivery Methods, the Printer MUST include this attribute in each Event
1032 Notification, and the value MUST be the value after it is increased by 1. That is, the value of this
1033 attribute in the first Event Notification after Subscription object creation MUST be 1, the second
1034 MUST be 2, etc. If a Delivery Method is defined such that the Notification Recipient returns a
1035 response, the Printer can re-try sending an Event Notification a certain number of times with the same
1036 sequence number when the Notification Recipient fails to return a response.

1037 If a Subscription Object lasts long enough to reach the value of MAX, its next value MUST be 0, i.e., it
1038 wraps.

1039 5.4.3 notify-lease-expiration-time (integer(0:MAX))

1040 This attribute specifies the time in the future when the lease on the Per-Printer Subscription Object will
1041 expire, i.e. the “printer-up-time” value at which the lease will expire. If the value is 0, the lease never
1042 expires.

1043 A Printer MUST support this attribute.

1044 When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present – the
1045 Subscription Object lasts exactly as long as the associated Job object. See also the discussion of the
1046 ‘job-completed’ event in section 5.3.3.1.3 about retention of the Job object after completion so that a
1047 Notification Recipient can query the Job object after receiving the ‘job-completed’ Event Notification.

1048 When the Printer creates a Per-Printer Subscription Object, it populates this attribute with a value that
1049 is the sum of the values of the Printer’s “printer-up-time” attribute and the Subscription Object’s
1050 “notify-lease-duration” attribute with the following exception. If the value of the Subscription Object’s
1051 “notify-lease-duration” attribute is 0 (i.e., no expiration time), then the value of this attribute MUST be
1052 set to 0 (i.e., no expiration time).

1053 When the Printer powers up, it MUST ~~set-populate the value of~~ this attribute in each persistent
1054 Subscription Object with a value using the algorithm in the previous paragraph.

1055 When the “printer-up-time” equals the value of this attribute, the Printer MUST delete the Subscription
1056 Object. A client can extend a lease of a Per-Printer Subscription Object with the Renew-Subscription
1057 operation (see section 11.2.6).

1058 Note: In order to compute the number of seconds remaining in a lease for a Per-Printer Subscription
1059 Object, a client can subtract the Subscription’s “notify-printer-up-time” attribute (see section 5.4.4)
1060 from the Subscription’s “notify-lease-expiration-time” attribute.

1061 **5.4.4 notify-printer-up-time (integer(1:MAX))**

1062 This attribute is an alias for the Printer’s “printer-up-time” attribute “ (see [RFC2911] section 4.4.29).
1063 In other words, when this attribute is queried with the Get-Subscriptions or Get-Subscription-Attributes
1064 operations (see sections 11.2.4 and 11.2.5), the value returned is the current value of the Printer’s
1065 “printer-up-time” attribute, rather than the time at which the Subscription Object was created.

1066 A Printer MUST support this attribute.

1067 When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present. When
1068 the Printer creates a Per-Printer Subscription Object, this attribute MUST be present.

1069 Note: this attribute exists in a Per-Printer Subscription Object so that a client using the Get-
1070 Subscription-Attributes or Get-Subscription operations can convert the Per-Printer Subscription’s
1071 “notify-lease-expiration-time” attribute to wall clock time with one request. If the value of the “notify-
1072 lease-expiration-time” attribute is not 0 (i.e., no expiration time), then the difference between the
1073 “notify-lease-expiration-time” attribute and the “notify-printer-up-time” is the remaining number of
1074 seconds on the lease from the current time.

1075 **5.4.5 notify-printer-uri (uri)**

1076 This attribute identifies the Printer object that created this Subscription Object.

1077 A Printer MUST support this attribute.

1078 During a Subscription Creation Operation, the Printer MUST populate this attribute with the value of
1079 the “printer-uri” operation attribute in the request. From the Printer URI, the client can, for example,
1080 determine what security scheme was used.

1081 **5.4.6 notify-job-id (integer(1:MAX))**

1082 This attribute specifies whether the containing Subscription Object is a Per-Job or Per-Printer
1083 Subscription Object, and for Per-Job Subscription Objects, it specifies the associated Job.

1084 A Printer MUST support this attribute.

1085 If this attribute is not present, the Subscription Object MUST be a Per-Printer Subscription. If this
 1086 attribute is present, the Subscription Object MUST be a Per-Job Subscription Object and this attribute
 1087 MUST identify the Job with which the Subscription Object is associated.

1088 Note: This attribute could be useful to a Notification Recipient that receives an Event Notification
 1089 generated from a Per-Job Subscription Object and caused by a Printer Event. The Event Notification
 1090 gives access to the Printer and the Subscription Object. The Event Notification gives access to the
 1091 associated Job only via this attribute. See discussion of the ‘job-completed’ event in section 5.3.3.1.3
 1092 about retention of the Job object after completion so that a Notification Recipient can query the Job
 1093 object after receiving the ‘job-completed’ Event Notification.

1094 5.4.7 notify-subscriber-user-name (name(MAX))

1095 This attribute contains the name of the user who performed the Subscription Creation Operation.

1096 A Printer MUST support this attribute.

1097 The Printer ~~sets~~ **MUST populate** this attribute ~~to~~ **with** the most authenticated printable name that it can
 1098 obtain from the authentication service over which the Subscription Creation Operation was received.
 1099 The Printer uses the same mechanism for determining the value of this attribute as it does for a Job’s
 1100 “job-originating-user-name” (see [RFC2911] section 4.3.6).

1101 Note: To help with authentication, a Subscription Object may have additional private attributes about
 1102 the user, e.g., a credential of a principal. Such private attributes are implementation-dependent and not
 1103 defined in this document.

1104 6 Printer Description Attributes Related to Notification

1105 This section defines the Printer Description attributes that are related to Notification. Table 3 lists the
 1106 Printer Description attributes, indicates the Printer support required for conformance, and whether or
 1107 not the attribute is READ-ONLY (see section 3.1):

1108 **Table 3 – Printer Description Attributes Associated with Notification**

| Printer object attributes: | REQUIRED | READ-ONLY |
|--|----------|-----------|
| printer-state-change-time (integer(1:MAX)) | No | Yes |
| printer-state-change-date-time (dateTime) | No | Yes |

1109

1110 6.1 printer-state-change-time (integer(1:MAX))

1111 This OPTIONAL attribute records the most recent time at which the ‘printer-state-changed’ Printer
1112 Event occurred whether or not any Subscription objects were listening for this event. This attribute
1113 helps a client or operator to determine how long the Printer has been in its current state.

1114 A Printer MAY support this attribute and if so, the attribute MUST be READ-ONLY.

1115 On power-up, the Printer MUST ~~set-populate the value of~~ this attribute ~~to-be~~with the value of its
1116 “printer-up-time” attribute, so that it always has a value. Whenever the ‘printer-state-changed’ Printer
1117 Event occurs, the Printer MUST ~~set-update~~ this attribute ~~to~~-with the value of the Printer’s “printer-up-
1118 time” attribute.

1119 6.2 printer-state-change-date-time (dateTime)

1120 This OPTIONAL attribute records the most recent time at which the ‘printer-state-changed’ Printer
1121 Event occurred whether or not there were any Subscription Objects listening for this event. This
1122 attribute helps a client or operator to determine how long the Printer has been in its current state.

1123 A Printer MAY support this attribute and if so, the attribute MUST be READ-ONLY.

1124 On power-up, the Printer MUST ~~populate set the value of~~ this attribute ~~to-be~~with the value of its
1125 “printer-current-time” attribute, so that it always has a value (see [RFC2911] section 4.4.30 on
1126 “printer-current-time”). Whenever the ‘printer-state-changed’ Printer Event occurs, the Printer MUST
1127 ~~set-update~~ this attribute ~~to~~-with the value of the Printer’s “printer-current-time” attribute.

1128 7 New Values for Existing Printer Description Attributes

1129 This section contains those attributes for which additional values are added.

1130 7.1 operations-supported (1setOf type2 enum)

1131 The following “operation-id” values are added in order to support the new operations defined in this
1132 document:

1133

Table 4 – Operation-id assignments

| Value | Operation Name |
|--------|------------------------------|
| 0x0016 | Create-Printer-Subscriptions |
| 0x0017 | Create-Job-Subscriptions |
| 0x0018 | Get-Subscription-Attributes |
| 0x0019 | Get-Subscriptions |
| 0x001A | Renew-Subscription |
| 0x001B | Cancel-Subscription |

1134 8 Attributes Only in Event Notifications

1135 This section contains those attributes that exist only in Event Notifications and do not exist in any
1136 objects.

1137 8.1 notify-subscribed-event (type2 keyword)

1138 This attribute indicates the Subscribed Event that caused the Printer to send this Event Notification.
1139 This attribute exists only in Event Notifications.

1140 This attribute MUST contain one of the values of the “notify-events” attribute in the Subscription
1141 Object, i.e., one of the Subscribed Event values. Its value is the Subscribed Event that “matches” the
1142 Event that caused the Printer to send this Event Notification. This Subscribed Event value may be
1143 identical to the Event or the Event may be a sub-value of the Subscribed Event. For example, the ‘job-
1144 completed’ Event (which is a sub-event of the ‘job-state-changed’ event) would cause the Printer to
1145 send an Event Notification for either the ‘job-completed’ or ‘job-state-changed’ Subscribed Events and
1146 to send the ‘job-completed’ or ‘job-state-changed’ value for this attribute, respectively,. See section
1147 5.3.3.2 for the “matching” rules of Subscribed Events and for additional examples.

1148 The Delivery Method Document specifies whether the Printer includes the value of this attribute in an
1149 Event Notification.

1150 8.2 notify-text (text(MAX))

1151 This attribute contains a Human Consumable text message (see section 9.2). This message describes
1152 the Event and is encoded as plain text, i.e., ‘text/plain’ with the charset specified by Subscription
1153 Object’s “notify-charset” attribute.

1154 The Delivery Method Document specifies whether the Printer includes this attribute in an Event
1155 Notification.

1156 9 Event Notification Content

1157 This section defines the Event Notification content that the Printer sends when an Event occurs.

1158 When an Event occurs, the Printer MUST find each Subscription object whose “notify-events”
1159 attribute “matches” the Event. See section 5.3.3.2 for details on “matching”. For each matched
1160 Subscription Object, the Printer MUST create an Event Notification with the content and format that
1161 the Delivery Method Document specifies. The content contains the value of attributes specified by the
1162 Delivery Method Document. The Printer obtains the values immediately after the Event occurs. For
1163 example, if the “printer-state” attribute changes from ‘idle’ to ‘processing’, the Event ‘printer-state-
1164 changed’ occurs and the Printer puts various attributes into the Event Notification, including “printer-
1165 up-time” and “printer-state” with the values that they have immediately after the Event occurs, i.e., the
1166 value of “printer-state” is ‘processing’.

1167 **Event Notification Ordering:**

1168 When a Printer sends Event Notifications, the Event Notifications from any given Subscription Object
1169 MUST be in time stamp order, i.e., in order of increasing “printer-up-time” attribute value in the Event
1170 Notification (see Table 5). These Event Notifications MAY be interleaved with those from other
1171 Subscription Objects, as long as those others are also in time stamp order. The Printer MUST observe
1172 these ordering requirements whether sending multiple pending Events as multiple separate Event
1173 Notifications or together in a single Compound Event Notification.

1174 If a Subscribing Client wants the Printer to send certain Event Notifications in time stamp order, the
1175 Subscribing Client uses a single Subscription Object. Even so, depending on the underlying transport,
1176 the actual order that a Notification Recipient receives separate Event Notifications may differ from the
1177 order sent by the Printer (e.g., email).

1178 Example: Consider two Per-Printer Subscription Objects: SO1 and SO2. SO1 requests ‘job-state-
1179 changed’ events and SO2 requests ‘printer-state-changed’ events. The number in parens is the time
1180 stamp. The following Event Notification sequences are the only ones that conform to the ordering
1181 requirements for the Printer to send the Event Notifications:

1182 (a) SO1: ‘job-created’ (1000), SO1: ‘job-stopped’ (1005), SO1: ‘job-completed’ (1009), SO2: ‘printer-
1183 stopped’ (1005)

1184 (b) SO1: ‘job-created’ (1000), SO1: ‘job-stopped’ (1005), SO2: ‘printer-stopped’ (1005), SO1: ‘job-
1185 completed’ (1009)

1186 (c) SO1: ‘job-created’ (1000), SO2: ‘printer-stopped’ (1005), SO1: ‘job-stopped’ (1005), SO1: ‘job-
1187 completed’ (1009)

1188 (d) SO2: ‘printer-stopped (1005), SO1: ‘job-created’ (1000), SO1: ‘job-stopped’ (1005), SO1: ‘job-
1189 completed’ (1009)

1190 Examples (b) and (c) are interleaved; examples (a) and (d) are not interleaved and are not appropriate
1191 for some Delivery Methods.

1192 If two different Events occur simultaneously, or nearly so (e.g., “printer-up-time” has the same value
1193 for both), the Printer MUST create a separate Event Notification for each Event, even if the associated
1194 Subscription Object is the same for both Events. However, the Printer MAY combine these distinct

1195 Event Notifications into a single Compound Event Notification if the Delivery Method supports
1196 Compound Event Notifications. For example, suppose that two nearly-simultaneously Events
1197 represent two successive ‘printer-state-changed’ Events, one from ‘idle’ to ‘processing’ and another
1198 from ‘processing’ to ‘stopped’. These two Events have the same name but are different instances of
1199 the Event. Then the Printer MUST create a separate Event Notification for each Event and SHOULD
1200 accurately report the “printer-state” of the first Event as ‘processing’ and the second Event as
1201 ‘stopped’.

1202 If a Subscription Object contains more than one Subscribed Event, and several Events occur in quick
1203 succession each matching a different Subscribed Event in the Subscription Object, the Printer MUST
1204 NOT generate a single Event Notification from several of these Events, but MAY combine distinct
1205 Event Notifications into a single Compound Event Notification if the Delivery Method supports
1206 Compound Event Notifications.

1207 After the Printer has created the Event Notification, the Printer delivers it via either a:

1208 Push Delivery Method: The Printer sends the Event Notification shortly after an Event occurs.
1209 For some Push Delivery Methods, the Notification Recipient MUST send a response; for others
1210 it MUST NOT send a response.

1211 Pull Delivery Method: The Printer saves Event Notifications for some Event Life and expects
1212 the Notification Recipient to request Event Notifications. The Printer returns the Event
1213 Notifications in a response to such a request.

1214 If an error that meets the following conditions occurs, the Printer MUST cancel the Subscription
1215 Object.

- 1216 a) the error occurs during the sending of an Event Notification generated from Subscription Object S
1217 AND
- 1218 b) the error would continue to occur every time the Printer sends an Event Notification generated from
1219 Subscription Object S in the future.

1220 For example, if the address of the “notify-recipient-uri” of Subscription Object A references a non-
1221 existent target and the Printer determines this fact, it MUST delete Subscription Object A.

1222 The next two sections describe the values that a Printer sends in the content of Machine Consumable
1223 and Human Consumable Event Notifications, respectively.

1224 The tables in the sub-sections of this section contain the following columns:

- 1225 a) **Source Value:** the name of the attribute that supplies the value for the Event Notification.
1226 Asterisks in this field refer to a note below the table.
- 1227 b) **Sends:** if the Printer supports the value (column 1) on the Source Object (column 3) the
1228 Delivery Method MUST specify:

1229 **MUST:** that the Printer **MUST** send the value.

1230 **SHOULD:** either that the Printer **MUST** send the value or that the value is incompatible
1231 with the Delivery Method.

1232 **MAY:** that the Printer **MUST**, **SHOULD**, **MAY**, **MUST NOT**, **SHOULD NOT**, or **NEED**
1233 **NOT** send the value. The Delivery Method specifies the level of conformance for the
1234 Printer.

1235 c) **Source Object:** the object from which the source value comes. If the object is “Event
1236 Notification”, the Printer fabricates the value when it sends the Event Notification. See section
1237 8.

1238 **9.1 Content of Machine Consumable Event Notifications**

1239 This section defines the attributes that a Delivery Method **MUST** mention in a Delivery Method
1240 Document when specifying the Machine Consumable Event Notification’s contents.

1241 This document does not define the order of attributes in Event Notifications. However, Delivery
1242 Method Documents **MAY** define the order of some or all of the attributes.

1243 A Delivery Method Document **MUST** specify additional attributes (if any) that a Printer
1244 implementation sends in a Machine Consumable Event Notification.

1245 Notification Recipients **MUST** be able to accept Event Notifications containing attributes they do not
1246 recognize. What a Notification Recipient does with an unrecognized attribute is implementation-
1247 dependent. Notification Recipients **MAY** attempt to display unrecognized attributes anyway or **MAY**
1248 ignore them.

1249 The next three sections define the attributes in Event Notification Contents that are:

1250 1. for all Events

1251 2. for Job Events only

1252 3. for Printer Events only

1253 **9.1.1 Event Notification Content Common to All Events**

1254 This section lists the attributes that a Delivery Method Document **MUST** specify for all Events.

1255 Table 5 lists potential values in each Event Notification.

1256

Table 5 – 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)) | SHOULD | Subscription |
| notify-charset (charset) | SHOULD | Subscription |
| notify-natural-language (naturalLanguage) | SHOULD | Subscription |
| notify-user-data (octetString(63)) ** | SHOULD | Subscription |
| notify-text (text) | SHOULD | Event Notification |
| attributes from the “notify-attributes” attribute *** | MAY | Printer |
| attributes from the “notify-attributes” attribute *** | MAY | Job |
| attributes from the “notify-attributes” attribute *** | MAY | Subscription |

1257

1258

1259

*A Printer MUST send this value only if and only if it supports the Printer’s “printer-current-time” attribute.

1260

1261

1262

** If the Subscription Object does not contain a “notify-user-data” attribute and the Delivery Method Document REQUIRES the Printer to send the “notify-user-data” source value in the Event Notification, the Printer MUST send an octet-string of length 0.

1263

1264

1265

1266

1267

1268

1269

*** The last three rows represent additional attributes that a client MAY request via the “notify-attributes” attribute. A Printer MAY support the “notify-attributes” attribute. The Delivery Method MUST say that the Printer MUST, SHOULD, MAY, MUST NOT, SHOULD NOT, or NEED NOT support the “notify-attributes” attribute and specific values of this attribute. The Delivery Method MAY say that support for the “notify-attributes” is conditioned on support of the attribute by the Printer or it MAY say that Printer MUST support the “notify-attributes” attribute if the Printer supports the Delivery Method.

1270

9.1.2 Additional Event Notification Content for Job Events

1271

1272

This section lists the additional attributes that a Delivery Method Document MUST specify for Job Events. See Table 6.

1273

Table 6 – Additional 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 |

1274

1275

1276

* The Printer MUST send the “job-impressions-completed” attribute in an Event Notification only for the combinations of Events and Subscribed Events shown in Table 7.

1277

Table 7 – 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’ |

1278

1279

9.1.3 Additional Event Notification Content for Printer Events

1280

1281

This section lists the additional attributes that a Delivery Method Document MUST specify for Printer Events. See Table 8.

1282

Table 8 – Additional 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 |

1283

1284

9.2 Content of Human Consumable Event Notification

1285

1286

1287

This section defines the information that a Delivery Method MUST mention in a Delivery Method Document when specifying the Human Consumable Event Notifications contents or the value of the “notify-text” attribute.

1288

Such a Delivery Method MUST specify the following information and a Printer SHOULD send it:

1289

a) the Printer name (see Table 9)

1290

b) the time of the Event (see Table 11)

- 1291 c) for Printer Events only:
1292 i) the Event (see Table 10) and/or Printer state information (see Table 14)
1293 d) for Job Events only:
1294 i) the job identity (see Table 12)
1295 ii) the Event (see Table 10) and/or Job state information (see Table 13)
1296

1297 The subsections of this section specify the attributes that a Printer MUST use to obtain this
1298 information.

1299 A Delivery Method Document MUST specify additional information (if any) that a Printer
1300 implementation sends in a Human Consumable Event Notification or in the “notify-text” attribute.

1301 A client MUST NOT request additional attributes via the “notify-attributes” attribute because this
1302 attribute works only for Machine Consumable Event Notifications.

1303 Notification Recipients MUST NOT expect to be able to parse the Human Consumable Event
1304 Notification contents or the value of the “notify-text” attribute.

1305 The next three sections define the attributes in Event Notification Contents that are:

- 1306 a) for all Events
1307 b) for Job Events only
1308 c) for Printer Events only
1309

1310 **9.2.1 Event Notification Content Common to All Events**

1311 This section lists the source of the information that a Delivery Method MUST specify for all Events.

1312 There is a separate table for each piece of information. Each row in the table represents a source value
1313 for the information and the values are listed in order of preference, with the first one being the
1314 preferred one. An implementation SHOULD use the source value from the earliest row in each table.
1315 It MAY use the source value from another row instead, or it MAY combine the source values from
1316 several rows. An implementation is free to determine the best way to present this information.

1317 In all tables of this section, all rows contain a “MAY” in order to state that the Delivery Method
1318 specifies the conformance.

1319 Table 9 lists the source of the information for the Printer Name. The “printer-name” is more user-
1320 friendly unless the Notification Recipient is in a place where the Printer name is not meaningful. For
1321 example, an implementation could have the intelligence to send the value of the “printer-name”
1322 attribute to a Notification Recipient that can access the Printer via value of the “printer-name” attribute
1323 and otherwise send the value of the “notify-printer-uri” attribute.

1324

Table 9 – Printer Name in Event Notification Content

| Source Value | Sends | Source Object |
|--------------------------|-------|---------------|
| printer-name (name(127)) | MAY | Printer |
| notify-printer-uri (uri) | MAY | Subscription |

1325

1326

1327

Table 10 lists the source of the information for the Event name. A Printer MAY combine this information with state information described for Jobs in Table 13 or for Printers in Table 14.

1328

Table 10 – Event Name in Event Notification Content

| Source Value | Sends | Source Object |
|---|-------|---------------|
| notify-subscribed-event (type2 keyword) | MAY | Subscription |

1329

1330

1331

1332

1333

Table 11 lists the source of the information for the time that the Event occurred. A Printer can send this value only if it supports the Printer's "printer-current-time" attribute. If a Printer does not support the "printer-current-time" attribute, it MUST NOT send the "printer-up-time" value instead, since it is not an allowed option for human consumable information.

1334

Table 11 – Event Time in Event Notification Content

| Source Value | Sends | Source Object |
|---------------------------------|-------|---------------|
| printer-current-time (dateTime) | MAY | Printer |

1335

1336

9.2.2 Additional Event Notification Content for Job Events

1337

1338

This section lists the source of the additional information that a Delivery Method MUST specify for Job Events.

1339

1340

Table 12 lists the source of the information for the job name. The "job-name" is likely more meaningful to a user than "job-id".

1341

Table 12 – Job Name in Event Notification Content

| Source Value | Sends | Source Object |
|-------------------------|-------|---------------|
| job-name (name(MAX)) | MAY | Job |
| job-id (integer(1:MAX)) | MAY | Job |

1342

1343 Table 13 lists the source of the information for the job state. If a Printer supports the “job-state-
 1344 message” and “job-detailed-state-message” attributes, it SHOULD use those attributes for the job state
 1345 information, otherwise, it should fabricate such information from the “job-state” and “job-state-
 1346 reasons”. For some Events, a Printer MAY combine this information with Event information.

1347 **Table 13 – Job State in Event Notification Content**

| Source Value | Sends | Source Object |
|---|-------|---------------|
| job-state-message (text(MAX)) | MAY | Job |
| job-detailed-status-messages (1setOf text(MAX)) | MAY | Job |
| job-state (type1 enum) | MAY | Job |
| job-state-reasons (1setOf type2 keyword) | MAY | Job |

1348

1349 9.2.3 Additional Event Notification Content for Printer Events

1350 This section lists the source of the additional information that a Delivery Method MUST specify for
 1351 Printer Events.

1352 Table 14 lists the source of the information for the printer state. If a Printer supports the “printer-state-
 1353 message”, it SHOULD use that attribute for the job state information, otherwise it SHOULD fabricate
 1354 such information from the “printer-state” and “printer-state-reasons”. For some Events, a Printer MAY
 1355 combine this information with Event information.

1356 **Table 14 – Printer State in Event Notification Content**

| Source Value | Sends | Source Object |
|--|-------|---------------|
| printer-state-message (text(MAX)) | MAY | Printer |
| printer-state (type1 enum) | MAY | Printer |
| printer-state-reasons (1setOf type2 keyword) | MAY | Printer |
| printer-is-accepting-jobs (boolean) | MAY | Printer |

1357 10 Delivery Methods

1358 A Delivery Method is the mechanism, i.e., protocol, by which the Printer delivers an Event Notification
 1359 to a Notification Recipient. There are several potential Delivery Methods for Event Notifications,
 1360 standardized, as well as proprietary. **This specification REQUIRES that the ‘ippget’ Pull Delivery**
 1361 **Method [ipp-get-method] be supported. Conforming implementations MAY support additional Push**
 1362 **or Pull Delivery Methods as well.** This document does not define any of these delivery mechanisms.
 1363 Each Delivery Method MUST be defined in a Delivery Method Document that is separate from this
 1364 document. New Delivery Methods will be created as needed using an extension to the registration
 1365 procedures defined in [RFC2911]. Such documents are registered with IANA (see section 19.7.3).

- 1366 The following sorts of Delivery Methods are ~~expected~~possible:
- 1367 - The Notification Recipient polls for Event Notifications at intervals directed by the Printer
 - 1368 - The Printer sends Event Notifications to the Notification Recipient using http as the transport.
 - 1369 - The Printer sends an email message.
- 1370 This section specifies how to define a Delivery Method Document and what to put in such a document.
- 1371 A Delivery Method Document MUST contain an exact copy of the following paragraph, caption and
 1372 table. In addition, column 2 of the table in the Delivery Method Document MUST contain answers to
 1373 questions in column 1 for the Delivery Method. Also, the Delivery Method document MUST contain a
 1374 reference to this document and call that reference [ipp-ntfy] because the table contains an [ipp-ntfy]
 1375 reference.
- 1376 If a Printer supports this Delivery Method, the following are its characteristics.

1377 **Table 15 – 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? | |
| 2. Is the Delivery Method REQUIRED, RECOMMENDED, or OPTIONAL for an IPP Printer to support? | |
| 3. What transport and delivery protocols does the Printer use to deliver the Event Notification Content, i.e., what is the entire network stack? | |
| 4. Can several Event Notifications be combined into a Compound Event Notification? | |
| 5. Is the Delivery Method initiated by the Notification Recipient (pull), or by the Printer (push)? | |
| 6. Is the Event Notification content Machine Consumable or Human 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? | |
| 8. What are the latency and reliability of the transport and delivery protocol? | |

| | |
|---|--|
| 9. What are the security aspects of the transport and delivery protocol, e.g., how it is handled in firewalls? | |
| 10. What are the content length restrictions? | |
| 11. What are the additional values or pieces of information that a Printer sends in an Event Notification content and the conformance requirements thereof? | |
| 12. What are the additional Subscription Template and/or Subscription Description attributes and the conformance requirements thereof? | |
| 13. What are the additional Printer Description attributes and the conformance requirements thereof? | |

1378

1379 11 Operations for Notification

1380 This section defines all of the operations for Notification. Section 7.1 assigns the “operation-id” for
 1381 each operation. The following two sub-sections define Subscription Creation Operations, and other
 1382 operations.

1383 11.1 Subscription Creation Operations

1384 This section defines the Subscription Creation Operations. The first section on Create-Job-
 1385 Subscriptions gives most of the information. The other Subscription Creation Operations refer to the
 1386 section on Create-Job-Subscriptions, even though the Create-Job-Subscriptions operation is the only
 1387 OPTIONAL operation in this document (see section 12).

1388 A Printer MUST support Create-Printer-Subscriptions and the Subscription Template Attributes Group
 1389 in Job Creation operations. It MAY support Create-Job-Subscriptions operations.

1390 11.1.1 Create-Job-Subscriptions Operation

1391 The operation creates one or more Per-Job Subscription Objects. The client supplies one or more
 1392 Subscription Template Attributes Groups each containing one or more of Subscription Template
 1393 Attributes (defined in section 5.3).

1394 Except for errors, the Printer MUST create exactly one Per-Job Subscription Object from each
 1395 Subscription Template Attributes Group in the request, even if the newly created Subscription Object
 1396 would have identical behavior to some existing Subscription Object. The Printer MUST associate each
 1397 newly created Per-Job Subscription Object with the target Job, which is specified by the “notify-job-id”
 1398 operation attribute.

1399 The Printer MUST accept the request in any of the target job’s ‘not-completed’ states, i.e., ‘pending’,
 1400 ‘pending-held’, ‘processing’, or ‘processing-stopped’. The Printer MUST NOT change the job’s “job-
 1401 state” attribute because of this operation. If the target job is in any of the ‘completed’ states, i.e.,

1402 ‘completed’, ‘canceled’, or ‘aborted, then the Printer MUST reject the request and return the ‘client-
1403 error-not-possible’ status code; the response MUST NOT contain any Subscription Attribute Groups.

1404 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section
1405 8.3) performing this operation MUST ~~either (1)~~ be the job owner, ~~or (2)~~ have Operator or
1406 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise
1407 authorized by the Printer’s administrator-configured security policy to create Per-Job Subscription
1408 Objects for the target job. Otherwise the Printer MUST reject the operation and return: the ‘client-
1409 error-forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-authorized’ status code as
1410 appropriate.

1411 11.1.1.1 Create-Job-Subscriptions Request

1412 The following groups of attributes are part of the Create-Job-Subscriptions Request:

1413 Group 1: Operation Attributes

1414 Natural Language and Character Set:

1415 The “attributes-charset” and “attributes-natural-language” attributes as described in
1416 [RFC2911] section 3.1.4.1.

1417

1418 Target:

1419 The “printer-uri” attribute which defines the target for this operation as described in
1420 [RFC2911] section 3.1.5.

1421

1422 Requesting User Name:

1423 The “requesting-user-name” attribute SHOULD be supplied by the client as described in
1424 [RFC2911] section 8.3.

1425 **11.1.1.1.1 notify-job-id (integer(1:MAX))**

1426 The client MUST supply this attribute and it MUST specify the Job object to associate the
1427 Per-Job Subscription with. The value of “notify-job-id” MUST be the value of the “job-id” of
1428 the associated Job object. If the client does not supply this attribute, the Printer MUST reject
1429 this request with a ‘client-error-bad-request’ status code.

1430

1431 Group 2-N: Subscription Template Attributes

1432 For each occurrence of this group:

1433

1434 The client MUST supply one or more Subscription Template Attributes in any order. See
1435 section 5.3 for a description of each such attribute. See section 5.2 for details on processing
1436 these attributes.

1437 **11.1.1.2 Create-Job-Subscriptions Response**

1438 The Printer **MUST** return to the client the following sets of attributes as part of a Create-Job-
1439 Subscriptions response:

1440 Group 1: Operation Attributes

1441 Status Message:

1442 In addition to the **REQUIRED** status code returned in every response, the response
1443 **OPTIONALLY** includes a "status-message" (text(255)) and/or a "detailed-status-message"
1444 (text(MAX)) operation attribute as described in [RFC2911] sections 13 and 3.1.6.

1445
1446 In this group, the Printer can return any status codes defined in [RFC2911] and section 12.
1447 The following is a description of the important status codes:

- 1448
1449 **successful-ok:** the Printer created all Subscription Objects requested (see [RFC2911]).
1450 **successful-ok-ignored-subscriptions:** the Printer created some Subscription Objects
1451 requested but some failed. The Subscription Attributes Groups with a "notify-status-
1452 code" attribute are the ones that failed (see section 12.1).
1453 **client-error-ignored-all-subscriptions:** the Printer created no Subscription Objects
1454 requested and all failed. The Subscription Attributes Groups with a "notify-status-
1455 code" attribute are the ones that failed (see section 12.2).
1456 **client-error-not-possible:** For this operation and other Per-Job Subscription operations,
1457 this error can occur because the specified Job has already completed (see [RFC2911]),
1458 whether or not the Job is retained in the Job Retention and/or Job History phases (see
1459 [RFC2911] section 4.3.7.1).

1460
1461 Natural Language and Character Set:

1462 The "attributes-charset" and "attributes-natural-language" attributes as described in
1463 [RFC2911] section 3.1.4.2.

1464
1465 Group 2: Unsupported Attributes

1466 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes. This group
1467 does not contain any unsupported Subscription Template Attributes; they are returned in the
1468 Subscription Attributes Group (see below).

1469
1470 Group 3-N: Subscription Attributes

1471 These groups **MUST** be returned unless the Printer is unable to interpret the entire request,
1472 e.g., the "status-code" parameter returned in Group 1 has the value: 'client-error-bad-request'.

1473
1474 "notify-status-code" (type2 enum):

1475 Indicates the status of this subscription (see section 13 for the status code definitions).
1476 Section 5.2 defines when this attribute **MUST** be present in this group.

1477

1478 See section 5.2 for details on the contents of each occurrence of this group.
1479

1480 11.1.2 Create-Printer-Subscriptions operation

1481 The operation is identical to Create-Job-Subscriptions with exceptions noted in this section.

1482 The operation creates Per-Printer Subscription Objects instead of Per-Job Subscription Objects, and
1483 associates each newly created Per-Printer Subscription Object with the Printer specified by the
1484 operation target rather than with a specific Job.

1485 The Printer MUST accept the request in any of its states, i.e., ‘idle’, ‘processing’, or ‘stopped’. The
1486 Printer MUST NOT change its “printer-state” attribute because of this operation.

1487 Access Rights: To create Per-Printer Subscription Objects, the authenticated user (see [RFC2911]
1488 section 8.3) performing this operation MUST have (1) Operator or Administrator access rights for this
1489 Printer (see [RFC2911] sections 1 and 8.5), or (2) be otherwise authorized by the Printer’s
1490 administrator-configured security policy to create Per-Printer Subscription Objects for this Printer.
1491 Otherwise, the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-
1492 not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

1493 11.1.2.1 Create-Printer-Subscriptions Request

1494 The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.1) except that the
1495 Operation Attributes group MUST NOT contain the “notify-job-id” attribute. If the client does supply
1496 the “notify-job-id” attribute, then the Printer MUST treat it as any other unsupported Operation
1497 attribute and MUST return it in the Unsupported Attributes group.

1498 11.1.2.2 Create-Printer-Subscriptions Response

1499 The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.2).

1500 11.1.3 Job Creation Operations – Extensions for Notification

1501 This document extends the Job Creation operations (see section 3.2) to create Subscription Objects as a
1502 part of the operation.

1503 The Job Creation operations are identical to Create-Job-Subscriptions operation with exceptions noted
1504 in this section.

1505 Unlike the Create-Job-Subscriptions operation, a Job Creation operation associates the newly created
1506 Subscription Objects with the Job object created by this operation. The operation succeeds if and only
1507 if the Job creation succeeds. If the Printer does not create some or all of the requested Subscription
1508 Objects, the Printer MUST return a ‘successful-ok-ignored-subscriptions’ status-code instead of a

1509 'successful-ok' status-code, but the Printer MUST NOT reject the operation because of a failure to
1510 create Subscription Objects.

1511 If the Job Creation operation includes a Job Template group, the client MUST supply it after the
1512 Operation Attributes group and before the first Subscription Template Attributes Group.

1513 If a Printer does not support this Notification specification, then it MUST treat the Subscription
1514 Attributes Group like an unknown group and ignore it (see [RFC2911] section 5.2.2). Because the
1515 Printer ignores the Subscription Attributes Group, it doesn't return them in the response either, thus
1516 indicating to the client that the Printer doesn't support Notification.

1517 After completion of a successful Job Creation operation, the Printer generates a 'job-created' event (see
1518 section 5.3.3.1.3).

1519 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section
1520 8.3) performing this operation MUST either have permission to create Jobs on the Printer or have
1521 Operator or Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5). Otherwise
1522 the Printer MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-
1523 authenticated', or 'client-error-not-authorized' status code as appropriate.

1524 **11.1.3.1 Job Creation Request**

1525 The groups for this operation are sufficiently different from the Create-Job-Subscriptions operation that
1526 they are all presented here. The following groups of attributes are supplied as part of a Job Creation
1527 Request:

1528 Group 1: Operation Attributes

1529 Same as defined in [RFC2911] for Print-Job, Print-URI, and Create-Job requests.

1530

1531 Group 2: Job Template Attributes

1532 The client OPTIONALLY supplies a set of Job Template attributes as defined in [RFC2911]
1533 section 4.2.

1534

1535 Group 3 to N: Subscription Template Attributes

1536 The same as Group 2-N in Create-Job-Subscriptions. See section 11.1.1.1.

1537 Group N+1: Document Content (Print-Job only)

1538 The client MUST supply the document data to be processed.

1539

1540 **11.1.3.2 Job Creation Response**

1541 The Printer MUST return to the client the following sets of attributes as part of a Print-Job, Print-URI,
1542 and Create-Job Response:

1543 Group 1: Operation Attributes

1544 Status Message:

1545

1546 As defined in [RFC2911] for Print-Job, Print-URI, and Create-Job requests.

1547

1548 In this group, the Printer can return any status codes defined in [RFC2911] and section 12.
1549 The following is a description of the important status codes:

1550

1551 **successful-ok:** the Printer created the Job and all Subscription Objects requested (see
1552 [RFC2911]).

1553 **successful-ok-ignored-subscriptions:** the Printer created the Job and not all of the
1554 Subscription Objects requested (see section 12.1). This status-code hides
1555 'successful-ok-xxx' status-codes that could reveal problems in Job creation. The
1556 Printer MUST NOT return the 'client-error-ignored-all-subscriptions' status code for
1557 Job Creation operations because the Printer returns an error status-code only when it
1558 fails to create a Job.

1559

1560 Natural Language and Character Set:

1561 The "attributes-charset" and "attributes-natural-language" attributes as described in
1562 [RFC2911] section 3.1.4.2.

1563

1564 Group 2: Unsupported Attributes

1565 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes. This group
1566 does not contain any unsupported Subscription Template Attributes; they are returned in the
1567 Subscription Attributes Group (see below).

1568

1569 Group 3: Job Object Attributes

1570 The "job-id" of the Job Object just created, etc., as defined in [RFC2911] for Print-Job, Print-
1571 URI, and Create-Job requests.

1572

1573 Group 4 to N: Subscription Attributes

1574 These groups MUST be returned if and only if the client supplied Subscription Template
1575 Attributes and the operation was accepted.

1576 See section 5.2 for details on the contents of each occurrence of this group.

1577

1578 11.2 Other Operations

1579 This section defines other operations on Subscription objects.

1580 11.2.1 Restart-Job Operation – Extensions for Notification

1581 The Restart-Job operation [RFC2911] is neither a Job Creation operation nor a Subscription Creation
1582 operation (see section 3.2). For the Restart-Job operation, the client MUST NOT supply any Job
1583 Subscription Attributes Groups. The Printer MUST treat any supplied Job Subscription Attributes as
1584 unsupported attributes.

1585 For this operation, the Printer does not return a job-id or any Subscription Attributes groups because
1586 the Printer reuses the existing Job object with the same job-id and the existing Per-Job Subscription
1587 Objects with the same subscription-ids. However, after successful completion of this operation, the
1588 Printer generates a 'job-created' event (see section 5.3.3.1.3).

1589 11.2.2 Validate-Job Operation – Extensions for Notification

1590 A client can test whether one or more Subscription Objects could be created using the Validate-Job
1591 operation. The client supplies one or more Subscription Template Attributes Groups (defined in
1592 section 5.3), just as in a Job Creation request.

1593 A Printer MUST support this extension to this operation.

1594 The Printer MUST accept requests that are identical to the Job Creation request defined in section
1595 11.1.3.1, except that the request MUST NOT contain document data.

1596 The Printer MUST return the same groups and attributes as the Print-Job operation (section 11.1.3.1)
1597 with the following exceptions. The Printer MUST NOT return a Job Object Attributes Group because
1598 no Job is created. The Printer MUST NOT return the "notify-subscription-id" attribute in any
1599 Subscription Attribute Group because no Subscription Object is created.

1600 If the Printer would succeed in creating a Subscription Object, the corresponding Subscription
1601 Attributes Group either has no 'status-code' attribute or a 'status-code' attribute with a value of
1602 'successful-ok-too-many-events' or 'successful-ok-ignored-or-substituted-attributes' (see sections 5.2
1603 and 13). The status-codes have the same meaning as in Job Creation except the results state what
1604 "would happen".

1605 The Printer MUST validate Subscription Template Attributes Groups in the same manner as the Job
1606 Creation operations.

1607 11.2.3 Get-Printer-Attributes – Extensions for Notification

1608 This operation is extended so that it returns Printer attributes defined in this document.

1609 A Printer MUST support this extension to this operation.

1610 In addition to the requirements of [RFC2911] section 3.2.5, a Printer MUST support the following
1611 additional values for the “requested-attributes” Operation attribute in this operation and return such
1612 attributes in the Printer Object Attributes group of its response.

1613 1. **Subscription Template Attributes:** Each supported attribute in column 2 of Table 1.

1614 2. **New Printer Description Attributes:** Each supported attribute in section 6.

1615 3. **New Group Name:** The ‘subscription-template’ group name, which names all supported
1616 Subscription Template Attribute in column 2 of Table 1. This group name is also used in the
1617 Get-Subscription-Attributes and Get-Subscriptions operation with an analogous meaning.

1618 4. **Extended Group Name:** The ‘all’ group name, which names all Printer attributes according to
1619 [RFC2911] section 3.2.5. In this extension ‘all’ names all attributes specified in [RFC2911]
1620 plus those named in items 1 and 2 of this list.

1621 11.2.4 Get-Subscription-Attributes operation

1622 This operation allows a client to request the values of the attributes of a Subscription Object.

1623 A Printer MUST support this operation.

1624 This operation is almost identical to the Get-Job-Attributes operation (see [RFC2911] section 3.3.4).
1625 The only differences are that the operation is directed at a Subscription Object rather than a Job object,
1626 and the returned attribute group contains Subscription Object attributes rather than Job object
1627 attributes.

1628 Access Rights: The authenticated user (see [RFC2911] section 8.3) performing this operation MUST
1629 (1) be the Subscription Object owner, (2) have Operator or Administrator access rights for this Printer
1630 (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer’s administrator-
1631 configured security policy to query the Subscription Object for the target job. Otherwise the Printer
1632 MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-not-authenticated’, or
1633 ‘client-error-not-authorized’ status code as appropriate. Furthermore, the Printer’s security policy
1634 MAY limit which attributes are returned, in a manner similar to the Get-Job-Attributes operation (see
1635 [RFC2911] end of section 3.3.4.2).

1636 11.2.4.1 Get-Subscription-Attributes Request

1637 The following groups of attributes are part of the Get-Subscription-Attributes request:

1638 Group 1: Operation Attributes

1639 Natural Language and Character Set:
1640 The “attributes-charset” and “attributes-natural-language” attributes as described in section
1641 [RFC2911] 3.1.4.1.
1642

1643 Target:
1644 The “printer-uri” attribute which defines the target for this operation as described in
1645 [RFC2911] section 3.1.5.
1646

1647 Requesting User Name:
1648 The “requesting-user-name” attribute SHOULD be supplied by the client as described in
1649 [RFC2911] section 8.3.

1650 **11.2.4.1.1 “notify-subscription-id” (integer (1:MAX))**

1651 The client MUST supply this attribute. The Printer MUST support this attribute. This
1652 attribute specifies the Subscription Object from which the client is requesting attributes. If the
1653 client omits this attribute, the Printer MUST reject this request with the ‘client-error-bad-
1654 request’ status code.

1655 **11.2.4.1.2 “requested-attributes” (1setOf keyword)**

1656 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute.
1657 This attribute specifies the attributes of the specified Subscription Object that the Printer
1658 MUST return in the response. Each value of this attribute is either an attribute name (defined
1659 in sections 5.3 and 5.4) or an attribute group name. The attribute group names are:

- 1660
- 1661 - ‘subscription-template’: all attributes that are both defined in section 5.3 and present on
1662 the specified Subscription Object (column 1 of Table 1).
- 1663 - ‘subscription-description’: all attributes that are both defined in section 5.4 and present
1664 on the specified Subscription Object (Table 2).
- 1665 - ‘all’: all attributes that are present on the specified Subscription Object.
1666

1667 A Printer MUST support all these group names.

1668 If the client omits this attribute, the Printer MUST respond as if this attribute had been
1669 supplied with a value of ‘all’.
1670

1671 **11.2.4.2 Get-Subscription-Attributes Response**

1672 The Printer returns the following sets of attributes as part of the Get-Subscription-Attributes Response:

1673 Group 1: Operation Attributes

1674 Status Message:
1675 Same as [RFC2911].
1676

1677 Natural Language and Character Set:
1678 The “attributes-charset” and “attributes-natural-language” attributes as described in
1679 [RFC2911] section 3.1.4.2. The “attributes-natural-language” MAY be the natural language
1680 of the Subscription Object, rather than the one requested.
1681

1682 Group 2: Unsupported Attributes

1683 See [RFC2911] section 3.1.7 and section 3.2.5.2 for details on returning Unsupported
1684 Attributes.

1685
1686 The response NEED NOT contain the "requested-attributes" operation attribute with any
1687 supplied keyword values that were requested by the client but are not supported by the IPP
1688 object. If the Printer object does return unsupported attributes referenced in the "requested-
1689 attributes" operation attribute, the values of the “requested-attributes” attribute returned
1690 MUST include only the unsupported keywords that were requested by the client. If the client
1691 had requested a group name, such as 'all', the resulting unsupported attributes returned MUST
1692 NOT include attribute keyword names described in the standard but not supported by the
1693 implementation.
1694

1695 Group 3: Subscription Attributes

1696 This group contains a set of attributes with their current values. Each attribute returned in this
1697 group:

- 1698 a) MUST be specified by the “requested-attributes” attribute in the request, AND
1699
1700 b) MUST be present on the specified Subscription Object AND
1701
1702 c) MUST NOT be restricted by the security policy in force. For example, a Printer MAY
1703 prohibit a client who is not the creator of a Subscription Object from seeing some or all
of its attributes. See [RFC2911] end of section 3.3.4.2 and section 8.

1704 The Printer can return the attributes of the Subscription Object in any order. The client
1705 MUST accept the attributes in any order.
1706

1707 11.2.5 Get-Subscriptions operation

1708 This operation allows a client to retrieve the values of attributes of all Subscription Objects belonging
1709 to a Job or Printer.

1710 A Printer MUST supported this operation.

1711 This operation is similar to the Get-Subscription-Attributes operation, except that this Get-
1712 Subscriptions operation returns attributes from possibly more than one object.

1713 This operation is similar to the Get-Jobs operation (see [RFC2911] section 3.2.6), except that the
1714 operation returns Subscription Objects rather than Job objects.

1715 Access Rights: To query Per-Job Subscription Objects of the specified job (client supplied the “notify-
1716 job-id” operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)
1717 performing this operation MUST (1) be the Subscription Object owner, (2) have Operator or
1718 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise
1719 authorized by the Printer’s administrator-configured security policy to query the Subscription Object
1720 for the target job. To query Per-Printer Subscription Objects of the Printer (client omits the “notify-
1721 job-id” operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)
1722 performing this operation MUST (1) have Operator or Administrator access rights for this Printer (see
1723 [RFC2911] sections 1 and 8.5), or (2) be otherwise authorized by the Printer’s administrator-
1724 configured security policy to query Per-Printer Subscription Objects for the target Printer. Otherwise
1725 the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-not-
1726 authenticated’, or ‘client-error-not-authorized’ status code as appropriate. Furthermore, the Printer’s
1727 security policy MAY limit which attributes are returned, in a manner similar to the Get-Jobs and Get-
1728 Printer-Attributes operations (see [RFC2911] end of sections 3.2.6.2 and 3.2.5.2).

1729 11.2.5.1 Get-Subscriptions Request

1730 The following groups of attributes are part of the Get-Subscriptions request:

1731 Group 1: Operation Attributes

1732 Natural Language and Character Set:

1733 The “attributes-charset” and “attributes-natural-language” attributes as described in
1734 [RFC2911] section 3.1.4.1.

1735

1736 Target:

1737 The “printer-uri” attribute which defines the target for this operation as described in
1738 [RFC2911] section 3.1.5.

1739

1740 Requesting User Name:

1741 The “requesting-user-name” attribute SHOULD be supplied by the client as described in
1742 [RFC2911] section 8.3.

1743 11.2.5.1.1 “notify-job-id” (integer(1:MAX))

1744 If the client specifies this attribute, the Printer returns the specified attributes of all Per-Job
1745 Subscription Objects associated with the Job whose “job-id” attribute value equals the value
1746 of this attribute. If the client does not specify this attribute, the Printer returns the specified
1747 attributes of all Per-Printer Subscription Objects. Note: there is no way to get all Per-Job
1748 Subscriptions known to the Printer in a single operation. A Get-Jobs operation followed by a
1749 Get-Subscriptions operation for each Job will return all Per-Job Subscriptions.

1750 11.2.5.1.2 “limit” (integer(1:MAX))

1751 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. It
1752 is an integer value that determines the maximum number of Subscription Objects that a client
1753 will receive from the Printer even if the “my-subscriptions” attribute constrains which
1754 Subscription Objects are returned. The limit is a “stateless limit” in that if the value supplied
1755 by the client is ‘N’, then only the first ‘N’ Subscription Objects are returned in the Get-
1756 Subscriptions Response. There is no mechanism to allow for the next ‘M’ Subscription
1757 Objects after the first ‘N’ Subscription Objects. If the client does not supply this attribute, the
1758 Printer responds with all applicable Subscription Objects.

1759 11.2.5.1.3 “requested-attributes” (1setOf type2 keyword)

1760 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute.
1761 This attribute specifies the attributes of the specified Subscription Objects that the Printer
1762 MUST return in the response. Each value of this attribute is either an attribute name (defined
1763 in sections 5.3 and 5.4) or an attribute group name (defined in section 11.2.4.1). If the client
1764 omits this attribute, the Printer MUST respond as if the client had supplied this attribute with
1765 the one value: ‘notify-subscription-id’.

1766 11.2.5.1.4 “my-subscriptions” (boolean)

1767 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. If
1768 the value is ‘false’, the Printer MUST consider the Subscription Objects from all users as
1769 candidates. If the value is ‘true’, the Printer MUST return the Subscription Objects created by
1770 the requesting user of this request. If the client does not supply this attribute, the Printer
1771 MUST respond as if the client had supplied the attribute with a value of ‘false’. The means
1772 for authenticating the requesting user and matching the Subscription Objects is similar to that
1773 for Jobs which is described in [RFC2911] section 8.
1774

1775 11.2.5.2 Get-Subscriptions Response

1776 The Printer returns the following sets of attributes as part of the Get-Subscriptions Response:

1777 Group 1: Operation Attributes

1778 Status Message:
1779 Same as [RFC2911].

1780
1781 Natural Language and Character Set:
1782 The “attributes-charset” and “attributes-natural-language” attributes as described in
1783 [RFC2911] section 3.1.4.2.

1784
1785 Group 2: Unsupported Attributes

1786 Same as for Get-Subscription-Attributes.

1787

1788

Groups 3 to N: Subscription Attributes

1789

The Printer responds with one Subscription Attributes Group for each requested Subscription Object (see the “notify-job-id” attribute in the Operation Attributes Group of this operation).

1790

1791

1792

The Printer returns Subscription Objects in any order.

1793

1794

If the “limit” attribute is present in the Operation Attributes group of the request, the number of Subscription Attributes Groups in the response MUST NOT exceed the value of the “limit” attribute.

1795

1796

1797

1798

If there are no Subscription Objects associated with the specified Job or Printer, the Printer MUST return zero Subscription Attributes Groups and it MUST NOT treat this case as an error, i.e., the status-code MUST be ‘successful-ok’ unless something else causes the status code to have some other value.

1799

1800

1801

1802

1803

See the Group 3 response (Subscription Attributes Group) of the Get-Subscription-Attributes operation (section 11.2.4.2) for the attributes that a Printer returns in this group.

1804

1805

1806

11.2.6 Renew-Subscription operation

1807

This operation allows a client to request the Printer to extend the lease on a Per-Printer Subscription Object.

1808

1809

The Printer MUST support this operation.

1810

The Printer MUST accept this request for a Per-Printer Subscription Object in any of the target Printer’s states, i.e., ‘idle’, ‘processing’, or ‘stopped’, but MUST NOT change the Printer’s “printer-state” attribute.

1811

1812

1813

The Printer MUST reject this request for a Per-Job Subscription Object because it has no lease (see section 5.4.3). The status code returned MUST be ‘client-error-not-possible’.

1814

1815

Access Rights: The authenticated user (see [RFC2911] section 8.3) performing this operation MUST ~~either~~ (1) be the owner of the Per-Printer Subscription Object, (2) ~~or~~ have Operator or Administrator access rights for the Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer’s administrator-configured security policy to renew Per-Printer Subscription Objects for the target Printer. Otherwise, the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

1816

1817

1818

1819

1820

1821

11.2.6.1 Renew-Subscription Request

1822

The following groups of attributes are part of the Renew-Subscription Request:

1823 Group 1: Operation Attributes

1824 Natural Language and Character Set:

1825 The “attributes-charset” and “attributes-natural-language” attributes as described in
1826 [RFC2911] section 3.1.4.1.

1827

1828 Target:

1829 The “printer-uri” attribute which defines the target for this operation as described in
1830 [RFC2911] section 3.1.5.

1831

1832 Requesting User Name:

1833 The “requesting-user-name” (name(MAX)) attribute SHOULD be supplied by the client as
1834 described in [RFC2911] section 8.3.

1835

1836 11.2.6.1.1 “notify-subscription-id” (integer (1:MAX))

1837 The client MUST supply this attribute. The Printer MUST support this attribute. This
1838 attribute specifies the Per-Printer Subscription Object whose lease the Printer MUST renew.
1839 If the client omits this attribute, the Printer MUST reject this request with the ‘client-error-
1840 bad-request’ status code.

1841

1842 Group 2: Subscription Template Attributes

1843 11.2.6.1.2 “notify-lease-duration” (integer(0:MAX))

1844 The client MAY supply this attribute. It indicates the number of seconds to renew the lease
1845 for the specified Subscription Object. A value of 0 requests an infinite lease (which MAY
1846 require Operator access rights). If the client omits this attribute, the Printer MUST use the
1847 value of the Printer’s “notify-lease-duration-default” attribute. See section 5.3.8 for more
1848 details.

1849

1850 11.2.6.2 Renew-Subscription Response

1851 The Printer returns the following sets of attributes as part of the Renew-Subscription Response:

1852 Group 1: Operation Attributes

1853 Status Message:

1854 Same as [RFC2911].

1855

1856 The following are some of the status codes returned (see [RFC2911]):

1857

1858 **successful-ok:** The operation successfully renewed the lease on the Subscription Object
1859 for the requested duration.

1860 **successful-ok-ignored-or-substituted-attributes:** The operation successfully renewed
1861 the lease on the Subscription Object for some duration other than the amount
1862 requested.

1863 **client-error-not-possible:** The operation failed because the “notify-subscription-id”
1864 Operation attribute identified a Per-Job Subscription Object.

1865 **client-error-not-found:** The operation failed because the “notify-subscription-id”
1866 Operation attribute identified a non-existent Subscription Object.

1867

1868 Natural Language and Character Set:

1869 The “attributes-charset” and “attributes-natural-language” attributes as described in
1870 [RFC2911] section 3.1.4.2. The “attributes-natural-language” MAY be the natural language
1871 of the Subscription Object, rather than the one requested.

1872

1873 Group 2: Unsupported Attributes

1874 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes.

1875

1876 Group 3: Subscription Attributes

1877 The Printer MUST return the following Subscription Attribute:

1878 **11.2.6.2.1 “notify-lease-duration” (integer(0:MAX))**

1879 The value of this attribute MUST be the number of seconds that the Printer has granted for the
1880 lease of the Subscription Object (see section 5.3.8 for details, such as the value of this
1881 attribute when the Printer doesn’t support the requested value).

1882 **11.2.7 Cancel-Subscription operation**

1883 This operation allows a client to delete a Subscription Object and stop the Printer from sending more
1884 Event Notifications. Once performed, there is no way to reference the Subscription Object.

1885 A Printer MUST supported this operation.

1886 The Printer MUST accept this request in any of the target Printer’s states, i.e., ‘idle’, ‘processing’, or
1887 ‘stopped’, but MUST NOT change the Printer’s “printer-state” attribute.

1888 If the specified Subscription Object is a Per-Job Subscription Object, the Printer MUST accept this
1889 request in any of the target Job’s states, but MUST NOT change the Job’s “job-state” attribute or affect
1890 the Job.

1891 Note: There is no way to change any attributes on a Subscription Object, except the “notify-lease-
1892 duration” attribute (using the Renew-Subscription operation). In order to change other attributes, a
1893 client performs a Subscription Creation Operation and Cancel-Subscription operation on the old
1894 Subscription Object. If the client wants to avoid missing Event Notifications, it performs the

1895 Subscription Creation Operation first. If this order would create too many Subscription Objects on the
1896 Printer, the client reverses the order.

1897 *Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation MUST
1898 ~~either (1)~~ be the owner of the Subscription Object, ~~(2) or~~ have Operator or Administrator access rights
1899 for the Printer (see [RFC2911] sections 1 and 8.5), ~~or (3) be otherwise authorized by the Printer's~~
1900 ~~administrator-configured security policy to cancel the target Subscription Object.~~ Otherwise, the
1901 Printer MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-
1902 authenticated', or 'client-error-not-authorized' status code as appropriate.

1903 11.2.7.1 Cancel-Subscription Request

1904 The following groups of attributes are part of the Cancel-Subscription Request:

1905 Group 1: Operation Attributes

1906 Natural Language and Character Set:

1907 The "attributes-charset" and "attributes-natural-language" attributes as described in
1908 [RFC2911] section 3.1.4.1.

1909

1910 Target:

1911 The "printer-uri" attribute which defines the target for this operation as described in
1912 [RFC2911] section 3.1.5.

1913

1914 Requesting User Name:

1915 The "requesting-user-name" attribute SHOULD be supplied by the client as described in
1916 [RFC2911] section 8.3.

1917 11.2.7.1.1 "notify-subscription-id" (integer (1:MAX))

1918 The client MUST supply this attribute. The Printer MUST support this attribute. This
1919 attribute specifies the Subscription Object that the Printer MUST cancel. If the client omits
1920 this attribute, the Printer MUST reject this request with the 'client-error-bad-request' status
1921 code.

1922

1923 11.2.7.2 Cancel-Subscription Response

1924 The Printer returns the following sets of attributes as part of the Cancel-Subscription Response:

1925 Group 1: Operation Attributes

1926 Status Message:

1927 Same as [RFC2911].

1928

1929 The following are some of the status codes returned (see [RFC2911]):

1930

1931

successful-ok: The operation successfully canceled (deleted) the Subscription Object.

1932

client-error-not-found: The operation failed because the “notify-subscription-id”

1933

Operation attribute identified a non-existent Subscription Object.

1934

1935

Natural Language and Character Set:

1936

The “attributes-charset” and “attributes-natural-language” attributes as described in

1937

[RFC2911] section 3.1.4.2. The “attributes-natural-language” MAY be the natural language

1938

of the Subscription Object, rather than the one requested.

1939

1940

Group 2: Unsupported Attributes

1941

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

1942

1943

12 Status Codes

1944

The following status codes are defined as extensions for Notification and are returned as the value of

1945

the “status-code” parameter in the Operation Attributes Group of a response (see [RFC2911] section

1946

3.1.6.1). Operations in this document can also return the status codes defined in section 13 of

1947

[RFC2911]. The ‘successful-ok’ status code is an example of such a status code.

1948

12.1 successful-ok-ignored-subscriptions (0x0003)

1949

The Subscription Creation Operation was unable to create all requested Subscription Objects.

1950

For a Create-Job-Subscriptions or Create-Printer-Subscriptions operation, this status code means that

1951

the Printer created one or more Subscription Objects, but not all requested Subscription Objects.

1952

For a Job Creation operation, this status code means that the Printer created the Job along with zero or

1953

more Subscription Objects. The Printer returns this status code even if other job attributes are

1954

unsupported or in conflict. That is, if an IPP Printer finds a warning that would allow it to return

1955

‘successful-ok-ignored-subscriptions’ and either ‘successful-ok-ignored-or-substituted-attributes’

1956

and/or ‘successful-ok-conflicting-attributes’, it MUST return ‘successful-ok-ignored-subscriptions’.

1957

12.2 client-error-ignored-all-subscriptions (0x0414)

1958

This status code is the same as ‘successful-ok-ignored-subscriptions’ except that only the Create-Job-

1959

Subscriptions and Create-Printer-Subscriptions operation return it. They return this status code only

1960

when the Printer creates zero Subscription Objects.

1961 **13 Status Codes in Subscription Attributes Groups**

1962 This section contains values of the “notify-status-code” (type2 enum) attribute that the Printer returns
1963 in a Subscription Attributes Group in a response when the corresponding Subscription Object:

1964 1. is not created or

1965 2. is created and some of the client-supplied attributes are not supported.

1966 The following sections are ordered in decreasing order of importance of the status-codes.

1967 **13.1 client-error-uri-scheme-not-supported (0x040C)**

1968 This status code is defined in [RFC2911]. This document extends its meaning and allows it to be in a
1969 Subscription Attributes Group of a response.

1970 The scheme of the client-supplied URI in a “notify-recipient-uri” Subscription Template Attribute in a
1971 Subscription Creation Operation is not supported. See section 5.3.1.

1972 **13.2 client-error-attributes-or-values-not-supported (0x040B)**

1973 This status code is defined in [RFC2911]. This document extends its meaning and allows it to be in a
1974 Subscription Attributes Group of a response.

1975 The method of the client-supplied keyword in a “notify-pull-method” Subscription Template Attribute
1976 in a Subscription Creation Operation is not supported. See section 5.3.2.

1977 **13.3 client-error-too-many-subscriptions (0x0415)**

1978 The number of Subscription Objects supported by the Printer would be exceeded if this Subscription
1979 Object were created (see section 5.2).

1980 **13.4 successful-ok-too-many-events (0x0005)**

1981 The client supplied more Events in the “notify-events” operation attribute of a Subscription Creation
1982 Operation than the Printer supports, as indicated in its “notify-max-events-supported” Printer attribute
1983 (see section 5.3.3).

1984 **13.5 successful-ok-ignored-or-substituted-attributes (0x0001)**

1985 This status code is defined in [RFC2911]. This document extends its meaning to include unsupported
1986 Subscription Template Attributes and it can appear in a Subscription Attributes Group.

1987 14 Encodings of Additional Attribute Tags

1988 This section assigns values to two attributes tags as extensions to the encoding defined in [RFC2910]).

1989 The “subscription-attributes-tag” delimits Subscription Template Attributes Groups in requests and
 1990 Subscription Attributes Groups in responses.

1991 The “event-notification-attributes-tag” delimits Event Notifications in Delivery Methods that use an
 1992 IPP-like encoding.

1993 The following table specifies the values for the delimiter tags:

| Tag Value (Hex) | Meaning |
|-----------------|-------------------------------------|
| 0x06 | “subscription-attributes-tag” |
| 0x07 | “event-notification-attributes-tag” |

1994 15 Conformance Requirements

1995 It is OPTIONAL for IPP clients and Printers to implement this Event Notification specification.

1996 15.1 Conformance requirements for clients

1997 If this Event Notification specification is implemented by a client, the client MUST support the
 1998 ‘ippget’ Pull Delivery Method and meet the conformance requirements as defined in [ipp-get-method]
 1999 for clients. A client MAY support additional Delivery Methods.

2000 15.2 Conformance requirements for Printers

2001 If this Event Notification specification is implemented by a Printer, the Printers MUST:

- 2002 - meet the Conformance Requirements detailed in section 5 of [RFC2911].
- 2003 - support the Subscription Template Attributes Group in requests and the Subscription
 2004 Attributes Group in responses.
- 2005 - support all of the following attributes:
 - 2006 a. REQUIRED Subscription Object attributes in section 5.
 - 2007 b. REQUIRED Printer Description object attributes in section 6.
 - 2008 c. REQUIRED attributes in Event Notification content in section 8.

2009 - support the ‘ippget’ Pull Delivery Method and meet the conformance requirements as defined
 2010 in [ipp-get-method] for Printers. The Printer MAY support additional Push and Pull Delivery
 2011 Methods.

- 2012 - send Event Notifications that conform to the requirements of section 9 and the requirements
 2013 of the Delivery Method Document for each supported Delivery Method (the conformance
 2014 requirements for Delivery Method Documents is specified in section 10).
- 2015 - for all of the Job Creation Operations that the Printer supports, MUST support the
 2016 REQUIRED extensions for notification defined in section 11.1.3.
- 2017 - meet the conformance requirements for operations as described in Table 16 and meet the
 2018 requirements for Printers as specified in the indicated sub-sections of section 11:

2019 **Table 16 – Printer Conformance Requirements for Operations**

| Operation | Printer Conformance Requirements |
|---|----------------------------------|
| Create-Printer-Subscriptions (section 11.1.2) | REQUIRED |
| Create-Job-Subscriptions (section 11.1.1) | OPTIONAL |
| Get-Subscription-Attributes (section 11.2.3) | REQUIRED |
| Get-Subscriptions (section 11.2.5) | REQUIRED |
| Renew-Subscription (section 11.2.6) | REQUIRED |
| Cancel-Subscription (section 11.2.7) | REQUIRED |

2020

2021 **16 Normative References**

2022 ~~[IANA-CON]~~

2023 ~~—— Narte, T. and Alvestrand, H.T.: Guidelines for Writing an IANA Considerations Section in RFCs,~~
 2024 ~~BCP 26, RFC 2434, October 1998.~~

2025 ~~[ipp-not-req]~~

2026 ~~—— deBry, R., Lewis, H., Hastings, T., “Internet Printing Protocol/1.1: Requirements for IPP~~
 2027 ~~Notifications”, <draft-ietf-ipp-not-06.txt>, work in progress, July 17, 2001.~~

2028 ~~[ipp-get-method]~~

2029 ~~—— Herriot, R., and T. Hastings, Kugler, C., and H. Lewis, “Internet Printing Protocol (IPP): The~~
 2030 ~~‘ippget’ Delivery Method for Event Notifications”, <draft-ietf-ipp-notify-get-07.txt>, June 27, 2002.~~

2031 ~~[ipp-prog]~~

2032 ~~Hastings, T., Bergman, R., and H. Lewis, H., “IPP: Job Progress Attributes”, <draft-ietf-ipp-job-~~
 2033 ~~prog-03.txt> work in progress, July 17, 2001.~~

2034 ~~[ipp-set]~~

2035 ~~—— Kugler, C., Hastings, T., Herriot, R., Lewis, H., “Internet Printing Protocol (IPP): Job and Printer Set~~
 2036 ~~Operations”, <draft-ietf-ipp-job-printer-set-ops-04.txt>, work in progress, July 17, 2001.~~

2037 ~~[RFC2026]~~

2038 ~~—— S. Bradner, “The Internet Standards Process — Revision 3”, RFC 2026, October 1996.~~

- 2039 [RFC2119]
2040 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997
- 2041 [RFC2396]
2042 Berners-Lee, T., Fielding, R., and L. Masinter, L., "Uniform Resource Identifiers (URI): Generic
2043 Syntax", RFC 2396, August 1998.
- 2044 ~~[RFC2565]~~
2045 ~~Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.0: Encoding and~~
2046 ~~Transport", RFC 2565, April 1999.~~
- 2047 ~~[RFC2566]~~
2048 ~~deBry, R., , Hastings, T., Herriot, R., Isaacson, S., Powell, P., "Internet Printing Protocol/1.0: Model~~
2049 ~~and Semantics", RFC 2566, April 1999.~~
- 2050 ~~[RFC2567]~~
2051 ~~Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.~~
- 2052 ~~[RFC2568]~~
2053 ~~Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",~~
2054 ~~RFC 2568, April 1999.~~
- 2055 ~~[RFC2569]~~
2056 ~~Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC~~
2057 ~~2569, April 1999.~~
- 2058 [RFC2717]
2059 R. Petke and I. King, "Registration Procedures for URL Scheme Names", RFC 2717, November
2060 1999.
- 2061 [RFC2910]
2062 Herriot, R., Butler, S., Moore, P., and R. Turner, R., "Internet Printing Protocol/1.1: Encoding and
2063 Transport", RFC 2910, September 2000.
- 2064 [RFC2911]
2065 deBry, R., ,Hastings, T., Herriot, R., Isaacson, S., and P. Powell, P., "Internet Printing Protocol/1.1:
2066 Model and Semantics", RFC 2911, September 2000.

2067 **17 Informative References**

- 2068 [IANA-CON]
2069 Narte, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs",
2070 BCP 26, RFC 2434, October 1998.

- 2071 [\[ipp-not-req\]](#)
2072 [deBry, R., Lewis, H., and T. Hastings, "Internet Printing Protocol/1.1: Requirements for IPP](#)
2073 [Notifications", <draft-ietf-ipp-not-06.txt>, work in progress, July 17, 2001.](#)
- 2074 [\[RFC2565\]](#)
2075 [Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.0: Encoding and](#)
2076 [Transport", RFC 2565, April 1999.](#)
- 2077 [\[RFC2566\]](#)
2078 [deBry, R., , Hastings, T., Herriot, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.0:](#)
2079 [Model and Semantics", RFC 2566, April 1999.](#)
- 2080 [\[RFC2567\]](#)
2081 [Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.](#)
- 2082 [\[RFC2568\]](#)
2083 [Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",](#)
2084 [RFC 2568, April 1999.](#)
- 2085 [\[RFC2569\]](#)
2086 [Herriot, R., Hastings, T., Jacobs, N., and J. Martin, "Mapping between LPD and IPP Protocols",](#)
2087 [RFC 2569, April 1999.](#)
- 2088 [\[RFC2616\]](#)
2089 [Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee,](#)
2090 ["Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.](#)
- 2091 [\[RFC3196\]](#)
2092 [Hastings, T., Manros, C., Zehler, P., Kugler, C., and H. Holst, "Internet Printing Protocol/1.1:](#)
2093 [Implementer's Guide", RFC3196, November 2001.](#)

2094 **18 Security Considerations**

2095 Clients submitting Notification requests to the IPP Printer have the same security issues as submitting
2096 an IPP/1.1 print job request ([see \[RFC2911\] section 3.2.1 and section 8](#)). The same mechanisms used
2097 by IPP/1.1 can therefore be used by the client Notification submission. Operations that require
2098 authentication can use the HTTP authentication. Operations that require privacy can use the
2099 HTTP/TLS privacy. As with IPP/1.1 Print Job Objects, if there is no security on Subscription Objects,
2100 sequential assignment of subscription-ids exposes the system to a passive traffic monitoring threat.

2101 **18.1 Client access rights**

2102 The ~~Notification-Subscription Object~~ access control model ~~should be is similar the same as to~~ the IPP
2103 access control model for Job objects. The client MUST have the following access rights for the
2104 indicated Subscription operations:

- 2105 1. Create-Job-Subscriptions (see section 11.1.1): A Per-Job Subscription object is associated with
2106 a Job. To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section
2107 8.3) performing this operation MUST (1) be the job owner, (2) have Operator or Administrator
2108 access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized
2109 by the Printer's administrator-configured security policy to create Per-Job Subscription Objects
2110 for the target job.
- 2111 2. Create-Printer-Subscriptions (see section 11.1.2): A Per-Printer Subscription object is
2112 associated with the Printer. To create Per-Printer Subscription Objects, the authenticated user
2113 (see [RFC2911] section 8.3) performing this operation MUST (1) have Operator or
2114 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5) or (2) be
2115 otherwise authorized by the Printer's administrator-configured security policy to create Per-
2116 Printer Subscription Objects for this Printer. ~~Creating a Per-Printer Subscription Object is~~
2117 ~~associated with a user. Some Subscription Objects (e.g., those that have a lifetime longer than a~~
2118 ~~job) can be done only by privileged users (users having Operator and/or Administrator access~~
2119 ~~rights), if that is the authorization policy.~~
- 2120 3. Get-Subscription-Attributes (see section 11.2.4): The access control model for this operation is
2121 the same as that of the Get-Job-Attributes operation (see [RFC2911] section 3.3.4). The
2122 primary difference is that a Get-Subscription-Attributes operation is directed at a Subscription
2123 Object rather than at a Job object, and a returned attribute group contains Subscription Object
2124 attributes rather than Job object attributes. To query the specified Subscription Object, the
2125 authenticated user (see [RFC2911] section 8.3) performing this operation MUST (1) be the
2126 Subscription Object owner, (2) have Operator or Administrator access rights for this Printer
2127 (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer's
2128 administrator-configured security policy to query the Subscription Object for the target job.
2129 ~~The system may limit the listing of items to only those items owned by the user. Furthermore,~~
2130 ~~the Printer's security policy MAY limit which attributes are returned, in a manner similar to the~~
2131 ~~Get-Job-Attributes operation (see [RFC2911] end of section 3.3.4.2).~~
- 2132 4. Get-Subscriptions (see section 11.2.5): The access control model for this operation is the same
2133 as that of the Get-Jobs operation (see [RFC2911] section 3.2.6). The primary difference is that
2134 the operation is directed at Subscription Objects rather than at Job objects, and the returned
2135 attribute groups contain Subscription Object attributes rather than Job object attributes. To
2136 query Per-Job Subscription Objects of the specified job (client supplied the "notify-job-id"
2137 operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)
2138 performing this operation MUST (1) be the Subscription Object owner, (2) have Operator or
2139 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be
2140 otherwise authorized by the Printer's administrator-configured security policy to query the
2141 Subscription Object for the target job. To query Per-Printer Subscription Objects of the Printer
2142 (client omits the "notify-job-id" operation attribute - see section 11.2.5.1.1), the authenticated
2143 user (see [RFC2911] section 8.3) performing this operation MUST (1) have Operator or
2144 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (2) be
2145 otherwise authorized by the Printer's administrator-configured security policy to query Per-
2146 Printer Subscription Objects for the target Printer. ~~The system may limit the listing of items to~~
2147 ~~only those items owned by the user. Furthermore, the Printer's security policy MAY limit~~

2148 which attributes are returned, in a manner similar to the Get-Job-Attributes operation (see
2149 [RFC2911] end of section 3.2.6.2).

2150 5. Renew-Subscriptions (see section 11.2.6): The authenticated user (see [RFC2911] section 8.3)
2151 performing this operation MUST (1) be the owner of the Per-Printer Subscription Object, (2)
2152 have Operator or Administrator access rights for the Printer (see [RFC2911] sections 1 and
2153 8.5), or (3) be otherwise authorized by the Printer's administrator-configured security policy to
2154 renew Per-Printer Subscription Objects for the target Printer

2155 6. Cancel-Subscription (see section 11.2.7): The authenticated user (see [RFC2911] section 8.3)
2156 performing this operation MUST (1) be the owner of the Subscription Object, (2) have Operator
2157 or Administrator access rights for the Printer (see [RFC2911] sections 1 and 8.5), or (3) be
2158 otherwise authorized by the Printer's administrator-configured security policy to cancel the
2159 target Subscription Object.~~Only the creator or an Operator can cancel the Subscription Object.~~

2160 The standard security concerns (delivery to the right user, privacy of content, tamper proof content)
2161 apply to ~~the each~~ Delivery Method. ~~IPP should use the security mechanism of the Delivery Method~~
2162 ~~used.~~ Some ~~d~~Delivery ~~mechanisms~~ Methods are more secure than others. Each Delivery Method
2163 Document MUST discuss its Security Considerations.~~Therefore, sensitive Event Notifications should~~
2164 ~~use the Delivery Method that has the strongest security.~~

2165 18.2 Printer security threats

2166 Notification trap door: If a Printer supports the OPTIONAL "notify-attributes" Subscription Template
2167 attribute (see section 5.3.4) where the client can request that the Printer return any specified Job,
2168 Printer, and Subscription object attributes, the Printer MUST apply the same security policy to these
2169 requested attributes in the Get-Notifications request as it does for the Get-Jobs, Get-Job-Attributes,
2170 Get-Printer-Attributes, and Get-Subscription-Attributes requests.

2171 18.3 Notification Recipient security threats

2172 Unwanted Events Notifications (spam): For any Push Delivery Method, Bby far the biggest security
2173 concern is the abuse of notification: sending unwanted Event Notifications to third parties (i.e., spam).
2174 The problem is made worse by notification addresses that may be redistributed to multiple parties ~~(e.g.,~~
2175 ~~mailing lists)~~. There exist scenarios where third party notification is ~~required~~ used (see Scenario #2
2176 and #3 in [ipp-not-req]). ~~The~~ Any fully secure solution would require active agreement of all recipients
2177 before sending out anything. ~~However, requirement #9 in [ipp-req] ("There is no requirement for IPP~~
2178 ~~Printer receiving the print request to validate the identity of an Event recipient") argues against this.~~
2179 ~~Certain systems may decide to disallow third party Event Notifications (a traditional fax model).~~

2180 **19 IANA Considerations**

2181 This section contains the registration information for IANA to add to the various IPP Registries
2182 according to the procedures defined in RFC 2911 [RFC2911] section 6 to cover the definitions in this

2183 document. In addition, this section defines how Events and Delivery Methods will be registered when
2184 they are defined in other documents.

2185 *Note to RFC Editors: Replace RFC NNNN below with the RFC number for this document, so that it*
2186 *accurately reflects the content of the information for the IANA Registry.*

2187 19.1 Attribute Registrations

2188 The following table lists all the attributes defined in this document. These are to be registered
2189 according to the procedures in RFC 2911 [RFC2911] section 6.2.

| 2190 | Subscription Template attributes: | Ref. | Section: |
|------|---|----------|----------|
| 2191 | notify-recipient-uri (uri) | RFC NNNN | 5.3.1 |
| 2192 | notify-schemes-supported (1setOf uriScheme) | RFC NNNN | 5.3.1 |
| 2193 | notify-pull-method (type2 keyword) | RFC NNNN | 5.3.2 |
| 2194 | notify-pull-method-supported (1setOf type2 keyword) | | |
| 2195 | | RFC NNNN | 5.3.2 |
| 2196 | notify-events (1setOf type2 keyword) | RFC NNNN | 5.3.3 |
| 2197 | notify-events-default (1setOf type2 keyword) | RFC NNNN | 5.3.3 |
| 2198 | notify-events-supported (1setOf type2 keyword) | RFC NNNN | 5.3.3 |
| 2199 | notify-max-events-supported (integer(2:MAX)) | RFC NNNN | 5.3.3 |
| 2200 | notify-attributes (1setOf type2 keyword) | RFC NNNN | 5.3.4 |
| 2201 | notify-attributes-supported (1setOf type2 keyword) | | |
| 2202 | | RFC NNNN | 5.3.4 |
| 2203 | notify-user-data (octetString(63)) | RFC NNNN | 5.3.5 |
| 2204 | notify-charset (charset) | RFC NNNN | 5.3.6 |
| 2205 | notify-natural-language (naturalLanguage) | RFC NNNN | 5.3.7 |
| 2206 | notify-lease-duration (integer(0:67108863)) | RFC NNNN | 5.3.8 |
| 2207 | notify-lease-duration-default (integer(0:67108863)) | | |
| 2208 | | RFC NNNN | 5.3.8 |
| 2209 | notify-lease-duration-supported (1setOf (integer(0: 67108863) | | |
| 2210 | rangeOfInteger(0:67108863))) | RFC NNNN | 5.3.8 |
| 2211 | notify-time-interval (integer(0:MAX)) | RFC NNNN | 5.3.9 |
| 2212 | | | |
| 2213 | Subscription Description Attributes: | | |
| 2214 | notify-subscription-id (integer (1:MAX)) | RFC NNNN | 5.4.1 |
| 2215 | notify-sequence-number (integer (0:MAX)) | RFC NNNN | 5.4.2 |
| 2216 | notify-lease-expiration-time (integer(0:MAX)) | RFC NNNN | 5.4.3 |
| 2217 | notify-printer-up-time (integer(1:MAX)) | RFC NNNN | 5.4.4 |
| 2218 | notify-printer-uri (uri) | RFC NNNN | 5.4.5 |
| 2219 | notify-job-id (integer(1:MAX)) | RFC NNNN | 5.4.6 |
| 2220 | notify-subscriber-user-name (name(MAX)) | RFC NNNN | 5.4.7 |
| 2221 | | | |
| 2222 | Printer Description Attributes: | | |
| 2223 | printer-state-change-time (integer(1:MAX)) | RFC NNNN | 6.1 |
| 2224 | printer-state-change-date-time (dateTime) | RFC NNNN | 6.2 |
| 2225 | | | |
| 2226 | Attributes Only in Event Notifications | | |
| 2227 | notify-subscribed-event (type2 keyword) | RFC NNNN | 8.1 |

2228 notify-text (text(MAX)) RFC NNNN 8.2
 2229
 2230 The resulting attribute registrations will be published in the
 2231 ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attributes/
 2232 area.
 2233

2234 19.2 Additional Enum Attribute Value Registrations for the “operations-supported” Printer 2235 Attribute

2236 The following table lists all the new enum attribute values defined in this document as additional type2
 2237 enum values for use with the “operations-supported” Printer Description attribute. These are to be
 2238 registered according to the procedures in RFC 2911 [RFC2911] section 6.1.

| 2239 | type2 enum Attribute Values: | Value | Ref. | Section: |
|------|------------------------------|--------|----------|----------|
| 2240 | Create-Printer-Subscriptions | 0x0016 | RFC NNNN | 7.1 |
| 2241 | Create-Job-Subscriptions | 0x0017 | RFC NNNN | 7.1 |
| 2242 | Get-Subscription-Attributes | 0x0018 | RFC NNNN | 7.1 |
| 2243 | Get-Subscriptions | 0x0019 | RFC NNNN | 7.1 |
| 2244 | Renew-Subscription | 0x001A | RFC NNNN | 7.1 |
| 2245 | Cancel-Subscription | 0x001B | RFC NNNN | 7.1 |

2246
 2247 The resulting enum attribute value registrations will be published in the
 2248 ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/operations-supported/
 2249 area.
 2250

2251 19.3 Operation Registrations

2252 The following table lists all of the operations defined in this document. These are to be registered
 2253 according to the procedures in RFC 2911 [RFC2911] section 6.4.

| 2254 | Operations: | Ref. | Section: |
|------|--|----------|----------|
| 2255 | Create-Job-Subscriptions Operation | RFC NNNN | 11.1.1 |
| 2256 | Create-Printer-Subscriptions Operation | RFC NNNN | 11.1.2 |
| 2257 | Job Creation Operations - Extensions | RFC NNNN | 11.1.3 |
| 2258 | Validate-Job Operation - Extensions | RFC NNNN | 11.2.2 |
| 2259 | Get-Printer-Attributes - Extensions | RFC NNNN | 11.2.3 |
| 2260 | Get-Subscription-Attributes Operation | RFC NNNN | 11.2.4 |
| 2261 | Get-Subscriptions Operation | RFC NNNN | 11.2.5 |
| 2262 | Renew-Subscription Operation | RFC NNNN | 11.2.6 |
| 2263 | Cancel-Subscription Operation | RFC NNNN | 11.2.7 |

2264
 2265 The resulting operation registrations will be published in the
 2266 ftp://ftp.iana.org/in-notes/iana/assignments/ipp/operations/
 2267 area.
 2268

2269 19.4 Status code Registrations

2270 The following table lists all the status codes defined in this document. These are to be registered
2271 according to the procedures in RFC 2911 [RFC2911] section 6.6.

| 2272 | Status codes: | Ref. | Section: |
|------|--|----------|----------|
| 2273 | successful-ok-ignored-subscriptions (0x0003) | RFC NNNN | 12.1 |
| 2274 | client-error-ignored-all-subscriptions (0x0414) | RFC NNNN | 12.2 |
| 2275 | | | |
| 2276 | Status Codes in Subscription Attributes Groups: | | |
| 2277 | client-error-uri-scheme-not-supported (0x040C) | RFC NNNN | 13.1 |
| 2278 | client-error-attributes-or-values-not-supported (0x040B) | | |
| 2279 | | RFC NNNN | 13.2 |
| 2280 | client-error-too-many-subscriptions (0x0415) | RFC NNNN | 13.3 |
| 2281 | successful-ok-too-many-events (0x0005) | RFC NNNN | 13.4 |
| 2282 | successful-ok-ignored-or-substituted-attributes (0x0001) | | |
| 2283 | | RFC NNNN | 13.5 |
| 2284 | | | |

2285 The resulting status code registrations will be published in the
2286 <ftp://ftp.iana.org/in-notes/iana/assignments/ipp/status-codes/>
2287 area.
2288

2289 19.5 Attribute Group tag Registrations

2290 The following table lists all the attribute group tags defined in this document. These are to be
2291 registered according to the procedures in RFC 2911 [RFC2911] section 6.5.

| 2292 | Attribute Group Tags: | Tag Value: | Ref. | Section: |
|------|-----------------------------------|------------|----------|----------|
| 2293 | subscription-attributes-tag | 0x06 | RFC NNNN | 14 |
| 2294 | event-notification-attributes-tag | 0x07 | RFC NNNN | 14 |
| 2295 | | | | |

2296 The resulting attribute group tag registrations will be published in the
2297 <ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-group-tags/>
2298 area.
2299

2300 19.6 Registration of Events

2301 When other document define additional type2 keywords to be used with the “notify-events”
2302 Subscription Template attribute (see section 5.3.3), these event keywords will be registered according
2303 to the procedures of [RFC2911] section 7.1 as additional attribute values for use with the “notify-
2304 events” Subscription Template attribute, i.e., the "notify-events", "notify-events-default", and "notify-
2305 events-supported" attributes.

2306 Therefore, the IPP Registry entry for an Event will be of the form:

| 2307 | type2 enum Attribute Values: | Ref. | Section: |
|------|------------------------------|------|----------|
|------|------------------------------|------|----------|

2308 <scheme name> RFC xxxx m.n
2309

2310 The resulting type2 keyword attribute values will be published in the
2311 ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/notify-events/
2312 area.
2313

2314 19.7 Registration of Event Notification Delivery Methods

2315 This section describes the requirements and procedures for registration and publication of Event
2316 Notification Delivery Methods and for the submission of such proposals.

2317 19.7.1 Requirements for Registration of Event Notification Delivery Methods

2318 Registered IPP Event Notification Delivery Methods are expected to follow a number of requirements
2319 described below.

2320 19.7.1.1 Required Characteristics

2321 A Delivery Method Document MUST either (1) contain all of the semantics of the Delivery Method or
2322 (2) contain the IPP Delivery Method registration requirements and a profile of some other protocol that
2323 in combination is the Delivery Method (e.g., mailto). ~~In either case, t~~The Delivery Method Document
2324 (and any documents it requires) MUST define ~~either (1) a URL for a Push Delivery Method or a~~
2325 ~~keyword for a Pull Delivery method and be a standards track, informational, or experimental RFC~~ that
2326 the meets the requirements of [RFC2717]. ~~or (2) a keyword for a Pull Delivery method.~~

2327
2328 IPP Event Notification Delivery Method Documents MUST meet the requirements of this document
2329 (see sections 9 and 10).

2330 In addition, a Delivery Method Document MUST contain the following information:

2331
2332 Type of registration: IPP Event Notification Delivery Method
2333 Name of this delivery method:
2334 Proposed URL scheme name of this Push Delivery Method or the keyword name of this Pull
2335 Delivery Method:
2336 Name of proposer:
2337 Address of proposer:
2338 Email address of proposer:
2339 Is this delivery method REQUIRED or OPTIONAL for conformance to the IPP Event Notification
2340 and Subscriptions document:
2341 Is this delivery method defining Machine Consumable and/or Human Consumable content:
2342

2343 **19.7.1.2 Naming Requirements**

2344 Exactly one (URL scheme or keyword) name **MUST** be assigned to each Delivery Method.

2345 Each assigned name **MUST** uniquely identify a single Delivery Method. All Push Delivery Method
2346 names **MUST** conform to the rules for URL scheme names, according to [RFC2396] and [RFC2717]
2347 for schemes in the IETF tree. All Pull Delivery Method names **MUST** conform to the rules for
2348 keywords according to [RFC2911].

2349 **19.7.1.3 Functionality Requirements**

2350 Delivery Methods **MUST** function as a protocol that is capable of delivering (push or pull) IPP Event
2351 Notifications to Notification Recipients.

2352 **19.7.1.4 Usage and Implementation Requirements**

2353 Use of a large number of Delivery Methods may hamper interoperability. However, the use of a large
2354 number of undocumented and/or unlabelled Delivery Methods hampers interoperability even more.

2355 A Delivery Method should therefore be registered **ONLY** if it adds significant functionality that is
2356 valuable to a large community, **OR** if it documents existing practice in a large community. Note that
2357 Delivery Methods registered for the second reason should be explicitly marked as being of limited or
2358 specialized use and should only be used with prior bilateral agreement.

2359 **19.7.1.5 Publication Requirements**

2360 Delivery Method Documents **MUST** be published in a standards track, informational, or experimental
2361 RFCs.

2362 **19.7.2 Registration Procedure**

2363 The IPP WG is developing a small number of Delivery Methods which are intended to be published as
2364 standards track RFCs. However, some parties may wish to register additional Delivery Methods in the
2365 future. This section describes the procedures for these additional Delivery Methods.

2366 **19.7.2.1 Present the proposal to the Community**

2367 First the Delivery Method Document **MUST** be an Internet-Draft with a target category of standards
2368 track, informational, or experimental. The same **MUST** be true for any documents that it references.

2369 Send the proposed Delivery Method Document proposal to the “ipp@pwg.org” mailing list. This
2370 mailing list has been established by [RFC2911] for reviewing proposed registrations and discussing

2371 other IPP matters. Proposed Delivery Method Documents are not formally registered and MUST NOT
2372 be used until approved.

2373 The intent of the public posting is to solicit comments and feedback on the definition and suitability of
2374 the Delivery Method and the name chosen for it over a four week period.

2375 **19.7.2.2 Delivery Method Reviewer**

2376 The Delivery Method Reviewer is the same person who has been appointed by the IETF Application
2377 Area Director(s) as the IPP Designated Expert according to [RFC2911] and [IANA-CON]. When the
2378 four week period is over and the IPP Designated Expert is convinced that consensus has been achieved,
2379 the IPP Designated Expert either approves the request for registration or rejects it. Rejection may
2380 occur because of significant objections raised on the list or objections raised externally.

2381 Decisions made by the Reviewer must be posted to the ipp@pwg.org mailing list within 14 days.
2382 Decisions made by the Reviewer may be appealed to the IESG.

2383 **19.7.2.3 IANA Registration**

2384 Provided that the Delivery Method registration proposal has either passed review or has been
2385 successfully appealed to the IESG, the IANA will register the Delivery Method and make it available to
2386 the community.

2387 **19.7.3 Delivery Method Document Registrations**

2388 Each Push Delivery Method Document defines a URI scheme which is registered as an additional value
2389 of the “notify-schemes-supported” Printer attribute. These uriScheme values will be registered
2390 according to the procedures of [RFC2911] section 7.1 for additional attribute values. Therefore, the
2391 IPP Registry entry for a Push Delivery Method will be of the form:

| | | | |
|------|-----------------------------|----------|----------|
| 2392 | uriScheme Attribute Values: | Ref. | Section: |
| 2393 | <scheme name> | RFC xxxx | m.n |

2394
2395 The resulting Delivery Method URI schemes will be published in the
2396 <ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/notify-schemes-supported/>
2397 area.
2398

2399 Each Pull Delivery Method Document defines a keyword method which is registered as an additional
2400 value of the “notify-pull-method-supported” Printer attribute. These keyword values will be registered
2401 according to the procedures of [RFC2911] section 7.1 for additional attribute values. Therefore, the
2402 IPP Registry entry for a Pull Delivery Method will be of the form:

| | | | |
|------|---------------------------|----------|----------|
| 2403 | keyword Attribute Values: | Ref. | Section: |
| 2404 | <method name> | RFC xxxx | m.n |

2405

2406 The resulting Delivery Method URI schemes will be published in the
2407 ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/notify-pull-method-supported/
2408 area.
2409

2410 **19.7.4 Registration Template**

2411 To: ipp@pwg.org
2412 Subject: Registration of a new Delivery Method

2413
2414 Delivery Method name:

2415
2416 (All Push Delivery Method names must be suitable for use as the value of a URL scheme in the IETF
2417 tree and all Pull Delivery Method names must be suitable IPP keywords according to [RFC2911])

2418
2419 Published specification(s):

2420
2421 (A specification for the Delivery Method must be openly available that accurately describes what is
2422 being registered.)

2423
2424 Person & email address to contact for further information:

2425 **20 Internationalization Considerations**

2426 This IPP Notification specification continues support for the internationalization of [RFC2911] of
2427 attributes containing text strings and names. Allowing a Subscribing Client to specify a different
2428 natural language and charset for each Subscription Object increases the internationalization support.

2429 The Printer MUST be able to localize the content of Human Consumable Event Notifications and to
2430 localize the value of “notify-text” attribute in Machine Consumable Event Notifications that it sends to
2431 Notification Recipients. For localization, the Printer MUST use the value of the “notify-charset”
2432 attribute and the “notify-natural-language” attribute in the Subscription Object supplied by the
2433 Subscribing Client.

2434 **21 Contributors**

2435 The following people made significant contributions to the design and review of this specification:

2436 Scott A. Isaacson
2437 Novell, Inc.
2438 122 E 1700 S
2439 Provo, UT 84606

2440
2441 Phone: 801-861-7366
2442 Fax: 801-861-2517

2443 [e-mail: sisaacson@novell.com](mailto:sisaacson@novell.com)
2444
2445 [Roger deBry](#)
2446 [Utah Valley State College](#)
2447 [Orem, UT 84058](#)
2448
2449 [Phone: \(801\) 222-8000](#)
2450 [EMail: debryro@uvsc.edu](mailto:debryro@uvsc.edu)
2451
2452 [Jay Martin](#)
2453 [Underscore Inc.](#)
2454 [9 Jacqueline St.](#)
2455 [Hudson, NH 03051-5308](#)
2456 [603-889-7000](#)
2457 [fax: 775-414-0245](#)
2458 [e-mail: jkm@underscore.com](mailto:jkm@underscore.com)
2459
2460 [Michael Shepherd](#)
2461 [Xerox Corporation](#)
2462 [800 Phillips Road MS 128-51E](#)
2463 [Webster, NY 14450](#)
2464
2465 [Phone: 716-422-2338](#)
2466 [Fax: 716-265-8871](#)
2467 [e-mail: mshepherd@crt.xerox.com](mailto:mshepherd@crt.xerox.com)
2468
2469 [Ron Bergman](#)
2470 [Hitachi Koki Imaging Solutions](#)
2471 [1757 Tapo Canyon Road](#)
2472 [Simi Valley, CA 93063-3394](#)
2473
2474 [Phone: 805-578-4421](#)
2475 [Fax: 805-578-4001](#)
2476 [Email: rbergma@hitachi-hkis.com](mailto:rbergma@hitachi-hkis.com)

2477 **22 Author's Addresses**

2478 Robert Herriot
2479 [706 Colorado Ave.](#)
2480 [Palo Alto, CA 94303](#)
2481
2482 [Phone: 650-327-4466](#)
2483 [2066 Byron St.](#)
2484 [Palo Alto, CA 94301](#)
2485

2486 ~~Phone: 650-326-8279~~
2487 Fax: 650-327-4466
2488 Email: bob@herriot.com
2489
2490 Tom Hastings
2491 Xerox Corporation
2492 737 Hawaii St. ESAE 231
2493 El Segundo, CA 90245
2494
2495 Phone: 310-333-6413
2496 Fax: 310-333-5514
2497 e-mail: hastings@cp10.es.xerox.com
2498
2499 ~~Scott AIsaacson~~
2500 ~~Novell, Inc.~~
2501 ~~122 E 1700 S~~
2502 ~~Provo, UT 84606~~
2503
2504 ~~Phone: 801-861-7366~~
2505 ~~Fax: 801-861-2517~~
2506 ~~e-mail: sisaacson@novell.com~~
2507
2508 ~~Roger deBry~~
2509 ~~Utah Valley State College~~
2510 ~~Orem, UT 84058~~
2511
2512 ~~Phone: (801) 222-8000~~
2513 ~~EMail: debryro@uvsc.edu~~
2514
2515 ~~Jay Martin~~
2516 ~~Underscore Inc.~~
2517 ~~9 Jacqueline St.~~
2518 ~~Hudson, NH 03051-5308~~
2519 ~~603-889-7000~~
2520 ~~fax: 775-414-0245~~
2521 ~~e-mail: jkm@underscore.com~~
2522
2523 ~~Michael Shepherd~~
2524 ~~Xerox Corporation~~
2525 ~~800 Phillips Road MS 128-51E~~
2526 ~~Webster, NY 14450~~
2527
2528 ~~Phone: 716-422-2338~~
2529 ~~Fax: 716-265-8871~~
2530 ~~e-mail: mshepherd@crt.xerox.com~~

2531

2532

[Ron Bergman](#)

2533

[Hitachi Koki Imaging Solutions](#)

2534

[1757 Tapo Canyon Road](#)

2535

[Simi Valley, CA 93063-3394](#)

2536

2537

[Phone: 805-578-4421](#)

2538

[Fax: 805-578-4001](#)

2539

[Email: rbergma@hitachi-hkis.com](#)

2540

2541

IPP Web Page: <http://www.pwg.org/ipp/>

2542

IPP Mailing List: ipp@pwg.org

2543

2544

To subscribe to the ipp mailing list, send the following email:

2545

1) send it to majordomo@pwg.org

2546

2) leave the subject line blank

2547

3) put the following two lines in the message body:

2548

subscribe ipp

2549

end

2550

2551

Implementers of this specification document are encouraged to join the IPP Mailing List in order to participate in any discussions of clarification issues and review of registration proposals for additional attributes and values. In order to reduce spam the mailing list rejects mail from non-subscribers, so you must subscribe to the mailing list in order to send a question or comment to the mailing list.

2552

2553

2554

2555

A. Appendix - Model for Notification with Cascading Printers

2556

With this model (see Figure 2), there is an intervening Print server between the human user and the output-device. So the system effectively has two Printers. There are two cases to consider.

2557

2558

1. When the Printer 1 (in the server) generates Events, the system behaves like the client and Printer in Figure 1. In this case, Printer 1 sends Event Notifications that are shown as Event Notifications (A) of Figure 2,.

2559

2560

2561

2. When the Printer 2 (in the output-device) generates Events, there are two possible system configurations:

2562

2563

a) Printer 1 forwards the client-supplied Subscription Creation Operations to the downstream Printer 2 and lets Printer 2 send the Event Notifications directly to the Notification Recipients supplied by the Client (Event Notifications(C) in the diagram).

2564

2565

2566

b) Printer 1 performs the client-supplied Subscription Creation Operations and also forwards the Subscription Creation Operations to Printer 2 with the Notification Recipient changed to be the Printer 1. When an Event occurs in Printer 2, Printer 2 sends the Event Notification (B) to Notification Recipient of Printer 1, which relays the received Event Notification (B) to the client-supplied Notification Recipient (as Event Notifications(A) in the diagram). Note, when a

2567

2568

2569

2570


```

2605                                     *****
2606                                     *
2607                                     * Printer (including
2608                                     * the distributed
2609                                     * Notification Service)
2610                                     *
2611                                     * output device or server
2612                                     * +-----+
2613 PDA, desktop, or server             * + ##### +
2614     +-----+                       * | # partial # |
2615     | client |---IPP Subscription---># Printer # |
2616     +-----+   Creation operation * | # Object # |
2617                                     * | #####|#####|
2618                                     * +-----+-----+
2619                                     * | Subscriptions
2620                                     * | OR Event
2621                                     * | Notifications
2622     +-----+                       * +-----v-----+
2623     |Notification|   IPP-defined      * | Notification |
2624     |Recipient   |<--Event Notifications---| Service   |
2625     +-----+                       * +-----+
2626                                     *
2627                                     *****
2628 *** = Implementation configuration opaque boundary
2629

```

Figure 3 – Opaque Use of a Notification Service Transparent to the Client

C. Appendix - Extended Notification Recipient

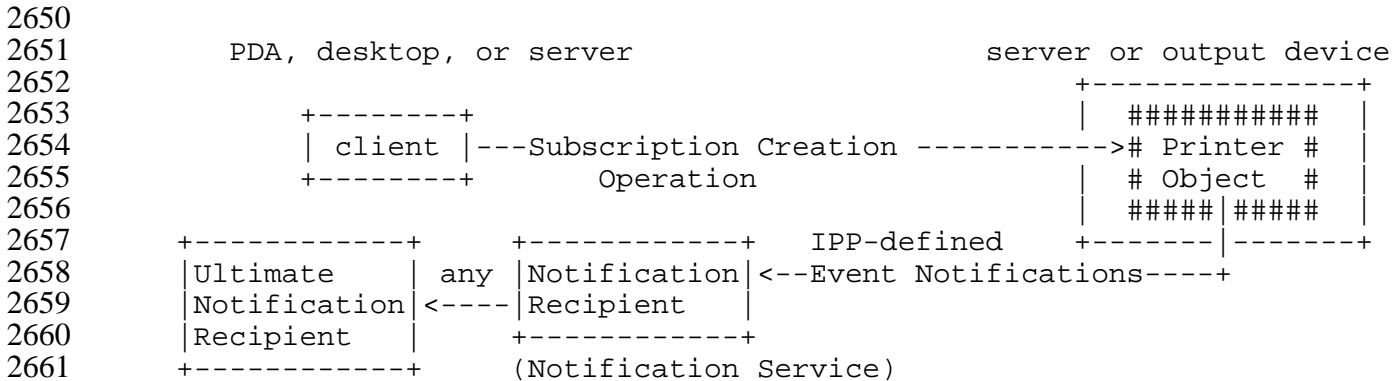
The model allows for an extended Notification Recipient that is itself a notification service that forwards each Event Notification to another recipient (called the Ultimate Notification Recipient in this section). The Delivery Method to the Ultimate Recipient is probably different from the Delivery Method used by the Printer to the extended Notification Recipient.

This extended Notification Recipient is transparent to the Printer but not to the client.

When a client performs a Subscription Creation Operation, it specifies the extended Notification Recipient as it would any Notification Recipient. In addition, the client specifies the Ultimate Notification Recipient in the Subscription Creation Operation in a manner specified by the extended Notification Recipient. Typically, it is either some bytes in the value of “notify-user-data” or some additional parameter in the value of “notify-recipient-uri”. The client also subscribes directly with the extended Notification Recipient (by means outside this document), since it is a notification service in its own right.

The IPP Printer treats the extended Notification Recipient like any other Notification Recipient and the IPP Printer is not aware of the forwarding. The Delivery Method that the extended Notification Recipient uses for delivering the Event Notification to the Ultimate Notification Recipient is beyond the scope of this document and is transparent to the IPP Printer.

2648 Examples of this extended Notification Recipient are paging, immediate messaging services, general
2649 notification services, and NOS vendors' infrastructure. Figure 4 shows this approach.



2662 Figure 4 – Use of an Extended Notification Recipient transparent to the Printer

2663 D. Appendix - Details about Conformance Terminology

2664 The following paragraphs provide more details about conformance terminology.

2665 **REQUIRED** - an adjective used to indicate that a conforming IPP Printer implementation MUST
2666 support the indicated operation, object, attribute, attribute value, status code, or out-of-band value
2667 in requests and responses. See [RFC2911] "Appendix A - Terminology for a definition of
2668 "support". *Since support of this entire Notification specification is OPTIONAL for
2669 conformance to ~~IPP/1.0~~ IPP/1.1, the use of the term REQUIRED in this document means
2670 "REQUIRED if this OPTIONAL Notification specification is implemented".*

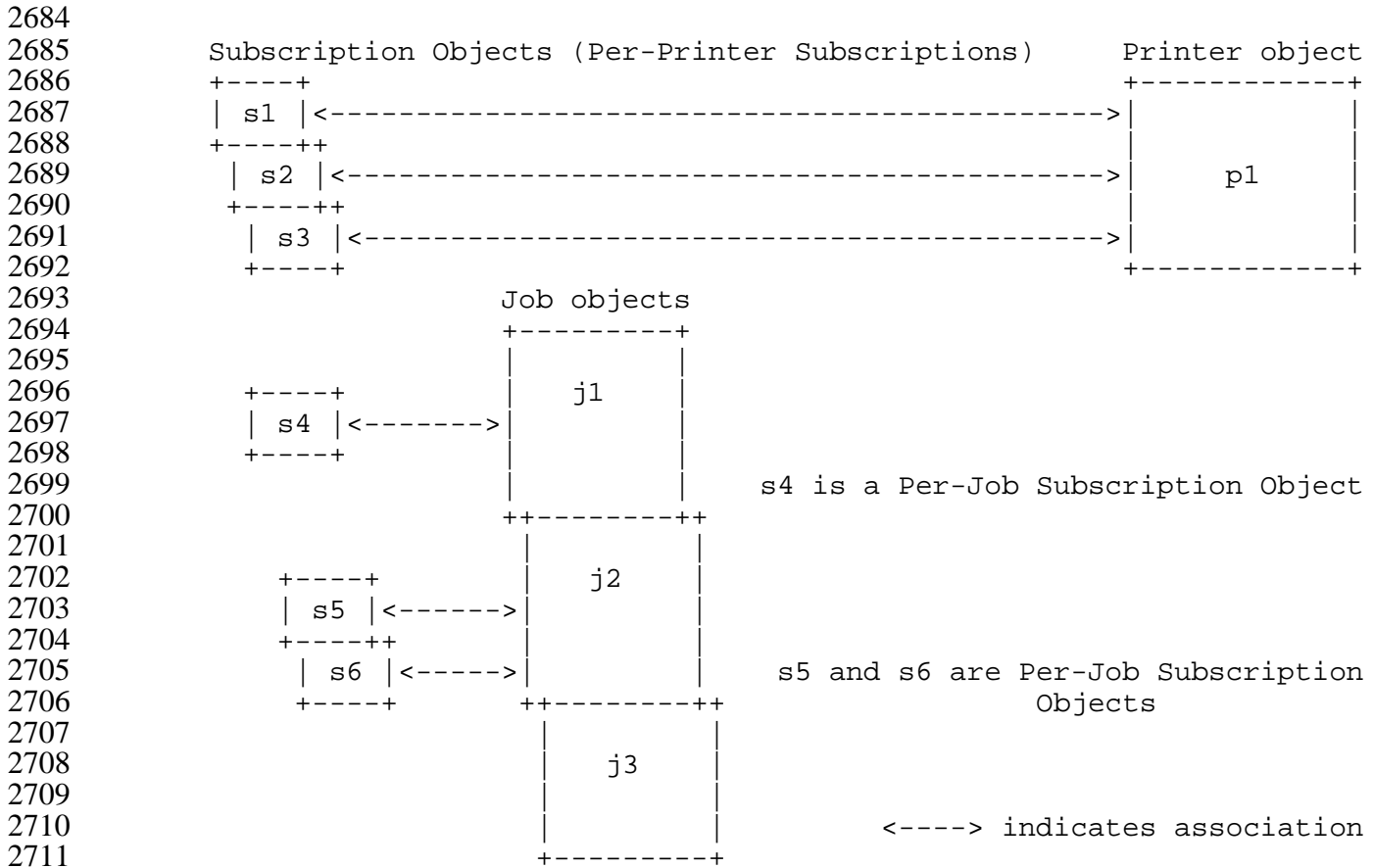
2671 **RECOMMENDED** - an adjective used to indicate that a conforming IPP Printer implementation is
2672 recommended to support the indicated operation, object, attribute, attribute value, status code, or
2673 out-of-band value in requests and responses. *Since support of this entire Notification
2674 specification is OPTIONAL for conformance to ~~IPP/1.0~~ IPP/1.1, the use of the term
2675 RECOMMENDED in this document means "RECOMMENDED if this OPTIONAL
2676 Notification specification is implemented".*

2677 **OPTIONAL** - an adjective used to indicate that a conforming IPP Printer implementation MAY, but is
2678 NOT REQUIRED to, support the indicated operation, object, attribute, attribute value, status code,
2679 or out-of-band value in requests and responses.

2680 E. Appendix - Object Model for Notification

2681 This section describes the Notification object model that adds a Subscription Object which together
2682 with the Job and Printer object provide the complete Notification semantics.

2683 The object relationships can be seen pictorially as:



2712 **Figure 5 – Object Model for Notification**

2713 s1, s2, and s3 are Per-Printer Subscription Objects and can identify Printer and/or Job Events.
 2714 s4, s5, and s6 are Per-Job Subscription Objects and can identify Printer and/or Job Events.

2715 E.1 Appendix - Object relationships

2716 This sub-section defines the object relationships between the Printer, Job, and Subscription Objects by
 2717 example. Whether Per-Printer Subscription Objects are actually contained in a Printer object or are
 2718 just bi-directionally associated with them in some way is IMPLEMENTATION DEPENDENT and is
 2719 transparent to the client. Similarly, whether Per-Job Subscription Objects are actually contained in a
 2720 Job object or are just bi-directionally associated with them in some way is IMPLEMENTATION
 2721 DEPENDENT and is transparent to the client. The object relationships are defined as follows:

2722 E.2 Printer Object and Per-Printer Subscription Objects

- 2723 1. The Printer object contains (is associated with) zero or more Per-Printer Subscription Objects
 2724 (p1 contains s1-s3 Per-Printer Subscription Objects).
- 2725 2. Each Per-Printer Subscription Object (s1, s2, and s3) is contained in (or is associated with)
 2726 exactly one Printer object (p1).

2727 **E.3 Job Object and Per-Job Subscription Objects**

- 2728 1. A Job object (j1, j2, j3) is associated with zero or more Per-Job Subscription Objects (s4-s6).
2729 Job j1 is associated with Per-Job Subscription Object s4, Job j2 is associated with Per-Job
2730 Subscription Objects s5 and s6, and Job j3 is not associated with any Per-Job Subscription
2731 Object.
- 2732 2. Each Per-Job Subscription Object is associated with exactly one Job object.

2733 **F. Appendix - Per-Job versus Per-Printer Subscription Objects**

2734 Per-Job and Per-Printer Subscription Objects are quite similar. Either type of Subscription Object can
2735 subscribe to Job Events, Printer Events, or both. Both types of Subscription Objects can be queried
2736 using the Get-Subscriptions and Get-Subscription-Attributes operations and canceled using the Cancel-
2737 Subscription operation. Both types of Subscription Objects create Subscription Objects which have the
2738 same Subscription Object attributes defined. However, there are some semantic differences between
2739 Per-Job Subscription Objects and Per-Printer Subscription Objects. A Per-Job Subscription Object is
2740 established by the client when submitting a job and after creating the job using the Create-Job-
2741 Subscriptions operation by specifying the “job-id” of the Job with the “notify-job-id” attribute. A Per-
2742 Printer Subscription Object is established between a client and a Printer using the Create-Printer-
2743 Subscriptions operation. Some specific differences are:

- 2744 1. A client usually creates one or more Per-Job Subscription Objects as part of the Job Creation
2745 operations (Create-Job, Print-Job, and Print-URI), rather than using the OPTIONAL Create-Job-
2746 Subscriptions operation, especially since Printer implementations NEED NOT support the
2747 Create-Job-Subscriptions operation, since it is OPTIONAL.
- 2748 2. For Per-Job Subscription Objects, the Subscription Object is only valid while the job is “not-
2749 complete” (see sections 5.4.3) while for the Per-Printer Subscription Objects, the Subscription
2750 Object is valid until the time (in seconds) that the Printer returned in the “notify-lease-
2751 expiration-time” operation attribute.
- 2752 3. Job Events in a Per-Job Subscription Object apply only to “one job” (the Job created by the Job
2753 Creation operation or references by the Create-Job-Subscriptions operation) while Job Events in
2754 a Per-Printer Subscription Object apply to ALL jobs contained in the IPP Printer.

2755 **G. Appendix - Description of the base IPP documents**

2756 The base set of IPP documents includes:

- 2757 Design Goals for an Internet Printing Protocol [RFC2567]
2758 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
2759 Internet Printing Protocol/1.1: Model and Semantics [RFC2911]
2760 Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
2761 Internet Printing Protocol/1.1: Implementer's Guide [[RFC3196IPP-HG](#)]
2762 Mapping between LPD and IPP Protocols [RFC2569]

2763

2764 The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed
2765 printing functionality, and it enumerates real-life scenarios that help to clarify the features that need to
2766 be included in a printing protocol for the Internet. It identifies requirements for three types of users:
2767 end users, operators, and administrators. It calls out a subset of end user requirements that are satisfied
2768 in IPP/1.0 [RFC2566, RFC2565]. A few OPTIONAL operator operations have been added to IPP/1.1
2769 [RFC2911, RFC2910].

2770

2771 The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document
2772 describes IPP from a high level view, defines a roadmap for the various documents that form the suite
2773 of IPP specification documents, and gives background and rationale for the IETF IPP working group's
major decisions.

2774

2775 The "Internet Printing Protocol/1.1: Model and Semantics" document describes a simplified model
2776 with abstract objects, their attributes, and their operations. The model introduces a Printer and a Job.
2777 The Job supports multiple documents per Job. The model document also addresses how security,
internationalization, and directory issues are addressed.

2778

2779 The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the
2780 abstract operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It also
2781 defines the encoding rules for a new Internet MIME media type called "application/ipp". This
2782 document also defines the rules for transporting over HTTP a message body whose Content-Type is
"application/ipp". This document defines the 'ipp' scheme for identifying IPP printers and jobs.

2783

2784 The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to
2785 implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some
2786 of the considerations that may assist them in the design of their client and/or IPP object
2787 implementations. For example, a typical order of processing requests is given, including error
checking. Motivation for some of the specification decisions is also included.

2788

2789 The "Mapping between LPD and IPP Protocols" document gives some advice to implementers of
gateways between IPP and LPD (Line Printer Daemon) implementations.

2790 H. Appendix - Full Copyright Statement

2791

Copyright (C) The Internet Society (1998,1999,2000,2001,[2002](#)). All Rights Reserved

2792

2793 This document and translations of it may be copied and furnished to others, and derivative works that
2794 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published
2795 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright
2796 notice and this paragraph are included on all such copies and derivative works. However, this
2797 document itself may not be modified in any way, such as by removing the copyright notice or
2798 references to the Internet Society or other Internet organizations, except as needed for the purpose of
2799 developing Internet standards in which case the procedures for copyrights defined in the Internet
Standards process must be followed, or as required to translate it into languages other than English.

2800 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or
2801 its successors or assigns.

2802 This document and the information contained herein is provided on an “AS IS” basis and THE
2803 INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL
2804 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
2805 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
2806 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
2807 PARTICULAR PURPOSE.

2808 **Acknowledgement**

2809
2810 Funding for the RFC Editor function is currently provided by the Internet Society.