

1 Internet Printing Protocol WG
2 INTERNET-DRAFT
3 <draft-ietf-ipp-not-spec-10.txt>
4 Updates RFC 2910 and 2911
5 [Target Category: standards track]
6 Expires: April 10, 2003
7

R. Herriot
consultant
T. Hastings
Xerox Corporation
October 10, 2002

8 Internet Printing Protocol (IPP):
9 **Event Notifications and Subscriptions**
10

11 Copyright (C) The Internet Society (2002). All Rights Reserved.

12 Status of this Memo

13 This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC
14 2026. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas,
15 and its working groups. Note that other groups may also distribute working documents as Internet-
16 Drafts.

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

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

22 **Abstract**

23 This document describes an OPTIONAL extension to the Internet Printing Protocol/1.1: Model and
24 Semantics (RFC 2911, RFC 2910). This extension allows a client to subscribe to printing related
25 Events. Subscriptions are modeled as *Subscription Objects*. The Subscription Object specifies that
26 when one of the specified *Events* occurs, the Printer delivers an asynchronous *Event Notification* to the
27 specified *Notification Recipient* via the specified Push or Pull *Delivery Method* (i.e., protocol).

28 A client associates Subscription Objects with a particular Job by performing the Create-Job-
29 Subscriptions operation or by submitting a Job with subscription information. A client associates
30 Subscription Objects with the Printer by performing a Create-Printer-Subscriptions operation. Four
31 other operations are defined for Subscription Objects: Get-Subscriptions-Attributes, Get-Subscriptions,
32 Renew-Subscription, and Cancel-Subscription.

33 **Table of Contents**

34	1 Introduction.....	7
35	1.1 Notification Overview	7
36	2 Models for Notification	10
37	2.1 Model for Simple Notification (Normative).....	10
38	2.2 Additional Models for Notification (Informative).....	10
39	3 Terminology.....	10
40	3.1 Conformance Terminology.....	10
41	3.2 Other Terminology.....	11
42	4 Object Relationships	13
43	4.1 Printer and Per-Printer Subscription Objects.....	13
44	4.2 Printer, Job and Per-Job Subscription Objects.....	13
45	5 Subscription Object.....	13
46	5.1 Rules for Support of Subscription Template Attributes	14
47	5.2 Rules for Processing Subscription Template Attributes	15
48	5.3 Subscription Template Attributes	18
49	5.3.1 notify-recipient-uri (uri).....	19
50	5.3.1.1 notify-schemes-supported (1setOf uriScheme).....	20
51	5.3.2 notify-pull-method (type2 keyword).....	20
52	5.3.2.1 notify-pull-method-supported (1setOf type2 keyword)	20
53	5.3.3 notify-events (1setOf type2 keyword).....	20
54	5.3.3.1 notify-events-default (1setOf type2 keyword)	21
55	5.3.3.2 notify-events-supported (1setOf type2 keyword).....	21
56	5.3.3.3 notify-max-events-supported (integer(2:MAX))	21
57	5.3.3.4 Standard Values for Subscribed Events.....	21
58	5.3.3.4.1 No Events.....	22
59	5.3.3.4.2 Subscribed Printer Events.....	22
60	5.3.3.4.3 Subscribed Job Events	23
61	5.3.3.5 Rules for Matching of Subscribed Events	24
62	5.3.3.5.1 Rules for Matching of Printer Events	24
63	5.3.3.5.2 Rules for Matching of Job Events.....	25
64	5.3.3.5.3 Special Cases for Matching Rules	25
65	5.3.4 notify-attributes (1setOf type2 keyword).....	26
66	5.3.4.1 notify-attributes-supported (1setOf type2 keyword).....	27
67	5.3.5 notify-user-data (octetString(63))	27
68	5.3.6 notify-charset (charset)	28
69	5.3.7 notify-natural-language (naturalLanguage).....	28
70	5.3.8 notify-lease-duration (integer(0:67108863)).....	28
71	5.3.8.1 notify-lease-duration-default (integer(0:67108863))	29

72	5.3.8.2 notify-lease-duration-supported (1setOf (integer(0: 67108863) rangeOfInteger(0:67108863)))	
73	29	
74	5.3.9 notify-time-interval (integer(0:MAX))	30
75	5.4 Subscription Description Attributes.....	30
76	5.4.1 notify-subscription-id (integer (1:MAX)).....	31
77	5.4.2 notify-sequence-number (integer (0:MAX)).....	31
78	5.4.3 notify-lease-expiration-time (integer(0:MAX)).....	32
79	5.4.4 notify-printer-up-time (integer(1:MAX)).....	33
80	5.4.5 notify-printer-uri (uri)	33
81	5.4.6 notify-job-id (integer(1:MAX))	33
82	5.4.7 notify-subscriber-user-name (name(MAX))	34
83	6 Printer Description Attributes Related to Notification	34
84	6.1 printer-state-change-time (integer(1:MAX))	34
85	6.2 printer-state-change-date-time (dateTime)	35
86	7 New Values for Existing Printer Description Attributes	35
87	7.1 operations-supported (1setOf type2 enum).....	35
88	8 Attributes Only in Event Notifications	35
89	8.1 notify-subscribed-event (type2 keyword)	35
90	8.2 notify-text (text(MAX)).....	36
91	9 Event Notification Content.....	36
92	9.1 Content of Machine Consumable Event Notifications	38
93	9.1.1 Event Notification Content Common to All Events	39
94	9.1.2 Additional Event Notification Content for Job Events.....	40
95	9.1.3 Additional Event Notification Content for Printer Events.....	40
96	9.2 Content of Human Consumable Event Notification	41
97	9.2.1 Event Notification Content Common to All Events	41
98	9.2.2 Additional Event Notification Content for Job Events.....	43
99	9.2.3 Additional Event Notification Content for Printer Events.....	43
100	10 Delivery Methods.....	44
101	11 Operations for Notification.....	46
102	11.1 Subscription Creation Operations.....	46
103	11.1.1 Create-Job-Subscriptions Operation	46
104	11.1.1.1 Create-Job-Subscriptions Request	47
105	11.1.1.1.1 notify-job-id (integer(1:MAX))	47
106	11.1.1.2 Create-Job-Subscriptions Response.....	47
107	11.1.2 Create-Printer-Subscriptions operation.....	48
108	11.1.2.1 Create-Printer-Subscriptions Request.....	49
109	11.1.2.2 Create-Printer-Subscriptions Response	49
110	11.1.3 Job Creation Operations – Extensions for Notification.....	49
111	11.1.3.1 Job Creation Request	50

112	11.1.3.2 Job Creation Response.....	50
113	11.2 Other Operations.....	51
114	11.2.1 Restart-Job Operation – Extensions for Notification.....	51
115	11.2.2 Validate-Job Operation – Extensions for Notification	52
116	11.2.3 Get-Printer-Attributes – Extensions for Notification.....	52
117	11.2.4 Get-Subscription-Attributes operation.....	53
118	11.2.4.1 Get-Subscription-Attributes Request.....	53
119	11.2.4.1.1 “notify-subscription-id” (integer (1:MAX))	53
120	11.2.4.1.2 “requested-attributes” (1setOf keyword)	54
121	11.2.4.2 Get-Subscription-Attributes Response	54
122	11.2.5 Get-Subscriptions operation	55
123	11.2.5.1 Get-Subscriptions Request.....	56
124	11.2.5.1.1 “notify-job-id” (integer(1:MAX)).....	56
125	11.2.5.1.2 “limit” (integer(1:MAX)).....	56
126	11.2.5.1.3 “requested-attributes” (1setOf type2 keyword).....	56
127	11.2.5.1.4 “my-subscriptions” (boolean)	57
128	11.2.5.2 Get-Subscriptions Response	57
129	11.2.6 Renew-Subscription operation.....	58
130	11.2.6.1 Renew-Subscription Request.....	58
131	11.2.6.1.1 “notify-subscription-id” (integer (1:MAX))	58
132	11.2.6.1.2 “notify-lease-duration” (integer(0:MAX)).....	59
133	11.2.6.2 Renew-Subscription Response	59
134	11.2.6.2.1 “notify-lease-duration” (integer(0:MAX)).....	60
135	11.2.7 Cancel-Subscription operation.....	60
136	11.2.7.1 Cancel-Subscription Request.....	60
137	11.2.7.1.1 “notify-subscription-id” (integer (1:MAX))	61
138	11.2.7.2 Cancel-Subscription Response	61
139	12 Status Codes.....	62
140	12.1 successful-ok-ignored-subscriptions (0x0003).....	62
141	12.2 client-error-ignored-all-subscriptions (0x0414).....	62
142	13 Status Codes in Subscription Attributes Groups.....	62
143	13.1 client-error-uri-scheme-not-supported (0x040C).....	62
144	13.2 client-error-attributes-or-values-not-supported (0x040B)	63
145	13.3 client-error-too-many-subscriptions (0x0415).....	63
146	13.4 successful-ok-too-many-events (0x0005).....	63
147	13.5 successful-ok-ignored-or-substituted-attributes (0x0001).....	63
148	14 Encodings of Additional Attribute Tags.....	63
149	15 Conformance Requirements.....	63
150	15.1 Conformance requirements for clients.....	64
151	15.2 Conformance requirements for Printers.....	64
152	16 Appendix A - Model for Notification with Cascading Printers (Informative)	65

153	17 Appendix B - Distributed Model for Notification (Informative).....	66
154	18 Appendix C - Extended Notification Recipient (Informative)	67
155	19 Appendix D - Details about Conformance Terminology (Normative).....	68
156	20 Appendix E - Object Model for Notification (Normative).....	68
157	20.1 Object relationships	69
158	20.2 Printer Object and Per-Printer Subscription Objects	69
159	20.3 Job Object and Per-Job Subscription Objects.....	70
160	21 Appendix F - Per-Job versus Per-Printer Subscription Objects (Normative).....	70
161	22 Normative References.....	70
162	23 Informative References	71
163	24 IANA Considerations.....	72
164	24.1 Attribute Registrations.....	72
165	24.2 Additional Enum Attribute Value Registrations.....	73
166	24.3 Operation Registrations	73
167	24.4 Status code Registrations.....	74
168	24.5 Attribute Group tag Registrations.....	74
169	24.6 Registration of Events.....	74
170	24.7 Registration of Event Notification Delivery Methods	75
171	24.7.1 Requirements for Registration of Event Notification Delivery Methods	75
172	24.7.1.1 Required Characteristics	75
173	24.7.1.2 Naming Requirements	76
174	24.7.1.3 Functionality Requirements	76
175	24.7.1.4 Usage and Implementation Requirements	76
176	24.7.1.5 Publication Requirements	76
177	24.7.2 Registration Procedure.....	76
178	24.7.2.1 Present the proposal to the Community.....	76
179	24.7.2.2 Delivery Method Reviewer.....	77
180	24.7.2.3 IANA Registration	77
181	24.7.3 Delivery Method Document Registrations	77
182	24.7.4 Registration Template.....	78
183	25 Internationalization Considerations	78
184	26 Security Considerations	78
185	26.1 Client access rights	79
186	26.2 Printer security threats	80
187	26.3 Notification Recipient security threats.....	80
188	27 Contributors	80

189	28 Author's Addresses	81
190	29 Appendix G - Description of the base IPP documents (Informative)	82
191	30 Appendix H - Full Copyright Statement (Informative).....	83
192		
193	Tables	
194	Table 1 – Subscription Template Attributes	19
195	Table 2 – Subscription Description Attributes	31
196	Table 3 – Printer Description Attributes Associated with Notification.....	34
197	Table 4 – Operation-id assignments	35
198	Table 5 – Attributes in Event Notification Content.....	39
199	Table 6 – Additional Event Notification Content for Job Events.....	40
200	Table 7 – Combinations of Events and Subscribed Events for “job-impressions-completed”.....	40
201	Table 8 – Additional Event Notification Content for Printer Events.....	41
202	Table 9 – Printer Name in Event Notification Content.....	42
203	Table 10 – Event Name in Event Notification Content	42
204	Table 11 – Event Time in Event Notification Content	42
205	Table 12 – Job Name in Event Notification Content.....	43
206	Table 13 – Job State in Event Notification Content	43
207	Table 14 – Printer State in Event Notification Content	44
208	Table 15 – Information about the Delivery Method	45
209	Table 16 – Printer Conformance Requirements for Operations	65
210		
211	Figures	
212	Figure 1 – Model for Notification.....	10
213	Figure 2 – Model for Notification with Cascading Printers.....	66
214	Figure 3 – Opaque Use of a Notification Server Transparent to the Client.....	67
215	Figure 4 – Use of an Extended Notification Recipient transparent to the Printer	68
216	Figure 5 – Object Model for Notification	69
217		

218 1 Introduction

219 This IPP notification specification is an OPTIONAL extension to Internet Printing Protocol/1.1: Model
220 and Semantics [RFC2911, RFC2910]. See Appendix 29 for a description of the base IPP documents.
221 This document in combination with the following documents is intended to meet the most important
222 notification requirements described in [ipp-not-req]:

223 Internet Printing Protocol (IPP): “Job Progress Attributes” [RFC3381]

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

226
227 This specification REQUIRES that clients and Printers support the ‘ippget’ Pull Delivery Method [ipp-
228 get-method]. Conforming client and Printer implementations MAY support additional Push or Pull
229 Delivery Methods as well. Note: this document does not define any Delivery Methods itself, but it
230 does define the rules for conformance for Delivery Method Documents and their registration with
231 IANA (see section 24.7.3).

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

233 1.1 Notification Overview

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

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

- 246 1. The names of Subscribed Events that are of interest to the Notification Recipient.
- 247 2. The address (URL) of one Notification Recipient for a Push Delivery Method or the method for
248 a Pull Delivery Method.
- 249 3. The Delivery Method (i.e., the protocol) which the Printer uses to deliver the Event Notification.
- 250 4. Some opaque data that the Printer delivers to the Notification Recipient in the Event
251 Notification. For example, the Notification Recipient might use this opaque data as a
252 forwarding address for the Event Notification.
- 253 5. The charset to use in text fields within an Event Notification

254 6. The natural language to use in the text fields of the Event Notification

255 7. The requested lease time in seconds for the Subscription Object

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

258 - **Job Creation operation:** When a client performs such an operation (Print-Job, Print-URI,
259 and Create-Job), a client can include zero or more Subscription Template Attributes Groups in
260 the request. The Printer creates one Subscription Object for each Subscription Template
261 Attributes Group in the request, and the Printer associates each such Subscription Object with
262 the newly created Job. This document extends these operations' definitions in [RFC2911] by
263 adding Subscription Template Attributes Groups in the request and Subscription Attributes
264 Groups in the response.

265 - **Create-Job-Subscriptions operation:** A client can include one or more Subscription
266 Template Attributes Groups in the request. The Printer creates one Subscription Object for
267 each Subscription Template Attributes Group and associates each with the job that is the
268 target of this operation.

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

273 For each of the above operations:

274 - the Printer associates a Subscription Object with the Printer or a specific Job. When a
275 Subscription Object is associated with a Job Object, it is called a *Per-Job Subscription Object*.
276 When a Subscription Object is associated with a Printer Object, it is called a *Per-Printer*
277 *Subscription Object*.

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

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

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

- 287 - **Validate-Job operation:** When a client performs this operation, a client can include zero or
288 more Subscription Template Attributes Groups in the request. The Printer determines if it
289 could create one Subscription Object for each Subscription Template Attributes Group in the
290 request. This document extends this operation's definition in [RFC2911] by adding
291 Subscription Template Attributes Groups in the request and Subscription Attributes Groups in
292 the response.
- 293 - **Get-Subscription-Attributes operation:** This operation allows a client to obtain the
294 specified attributes of a target Subscription Object.
- 295 - **Get-Subscriptions operation:** This operation allows a client to obtain the specified attributes
296 of all Subscription Objects associated with the Printer or a specified Job.
- 297 - **Renew-Subscription operation:** This operation renews the lease on the target Per-Printer
298 Subscription Object before it expires. A newly created Per-Printer Subscription Object
299 receives an initial lease. It is the duty of the client to use this operation frequently enough to
300 preserve a Per-Printer Subscription Object. The Printer deletes a Per-Printer Subscription
301 Object when its lease expires. A Per-Job Subscription Object last exactly as long as its
302 associated Job Object and thus doesn't have a lease.
- 303 - **Cancel-Subscription operation:** This operation (1) cancels the lease on the specified Per-
304 Printer Subscription Object and thereby deletes the Per-Printer Subscription Object or (2)
305 deletes the Per-Job Subscription Object.

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

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

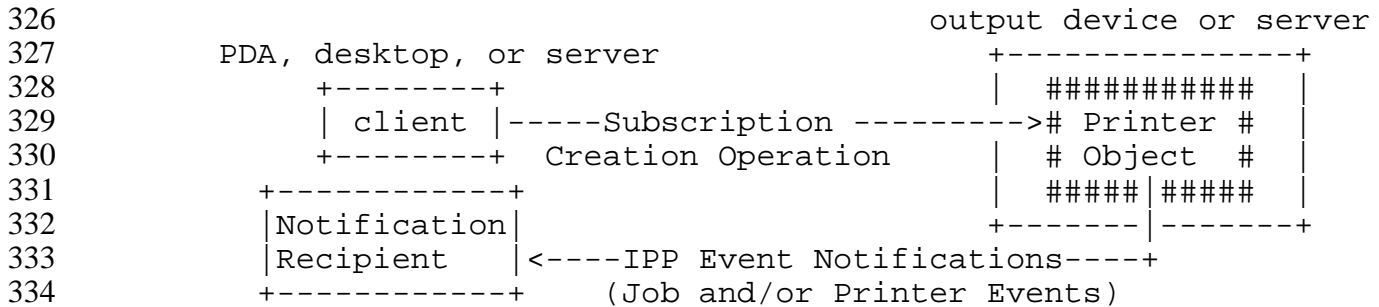
318 2 Models for Notification

319 2.1 Model for Simple Notification (Normative)

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

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

325 embedded printer:



335 **Figure 1 – Model for Notification**

336 2.2 Additional Models for Notification (Informative)

337 Additional models have been proposed (see Appendices 16, 17, and 18).

338 3 Terminology

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

341 3.1 Conformance Terminology

342 Capitalized terms, such as **MUST**, **MUST NOT**, **REQUIRED**, **SHOULD**, **SHOULD NOT**, **MAY**,
 343 **NEED NOT**, and **OPTIONAL**, have special meaning relating to conformance as defined in RFC 2119
 344 [RFC2119] and [RFC2911] section 12.1. If an implementation supports the extension defined in this
 345 document, then these terms apply; otherwise, they do not. These terms define conformance to *this*
 346 *document only*; they do not affect conformance to other documents, unless explicitly stated otherwise.
 347 See Appendix 19 for complete details.

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

350 **READ-ONLY** – an adjective used in an attribute definition to indicate that an IPP Printer **MUST NOT**
351 allow the attribute's value to be modified.

352 **3.2 Other Terminology**

353 This document uses the same terminology as [RFC2911], such as “**client**”, “**Printer**”, “**attribute**”,
354 “**attribute value**”, “**keyword**”, “**operation**”, “**request**”, “**response**”, “**administrator**”, “**operator**”,
355 and “**support**”. In addition, the following terms are defined for use in this document and the Delivery
356 Method Documents:

357 **Compound Event Notification** – two or more Event Notifications that a Printer delivers together as a
358 single request or response. The Delivery Method Document specifies whether the Delivery Method
359 supports Compound Event Notifications.

360 **Delivery Method** – the mechanism by which the Printer delivers an Event Notification.

361 **Delivery Method Document** – a document, separate from this document, that defines a Delivery
362 Method.

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

367 **Event Life** – For a Pull Delivery Method, the length of time in seconds after an Event occurs during
368 which the Printer will retain that Event for delivery in an Event Notification. After the Event Life
369 expires, the Printer will no longer deliver an Event Notification for that Event in such a response.

370 **Event Notification** – the information about an Event that the Printer delivers when an Event occurs.

371 **Event Notification Attributes Group** – The attributes group which is used to deliver an Event
372 Notification in a request (Push Delivery Methods) or a response (Pull Delivery Methods).

373 **Human Consumable Event Notification** – localized text for human consumption only. There is no
374 standardized format and thus programs should not try to parse this text.

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

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

381 **Machine Consumable Event Notification** – bytes for program consumption. The bytes are formatted
382 according to the Delivery Method document.

- 383 **Notification** – when not in the phrases ‘Event Notification’ and ‘Notification Recipient’ — the
384 concepts of this specification, i.e., Events, Subscription Objects, and Event Notifications.
- 385 **Notification Recipient** – the entity to which the Printer delivers an Event Notification. For Push
386 Delivery Methods, the IPP Printer sends the Notifications to a Notification Recipient. For Pull
387 Delivery Methods, the Notification Recipient is acting in the role of an IPP client and requests Event
388 Notifications and so the terms “client” and “Notification Recipient” are used interchangeably with such
389 Delivery Methods. For example, see [ipp-get-method].
- 390 **Per-Job Subscription Object** – A Subscription Object that is associated with a single Job. The
391 Create-Job-Subscriptions operation and Job Creation operations create such an object.
- 392 **Per-Printer Subscription Object** – A Subscription Object that is associated with the Printer as a
393 whole. The Create-Printer-Subscriptions operation creates such an object.
- 394 **Printer Event** – an Event caused by some change in the Printer that is not specific to a job, e.g.,
395 'printer-state-changed'.
- 396 **Pull Delivery Method** – The Printer saves Event Notifications for some event life time and expects the
397 Notification Recipient to request Event Notifications. The Printer delivers the Event Notifications in a
398 response to such a request.
- 399 **Push Delivery Method** – The Printer delivers the Event Notification shortly after an Event occurs.
- 400 **Subscribed Event** – an Event that the Subscribing Client expresses interest in by making it a value of
401 the “notify-events” attribute on a Subscription Object.
- 402 **Subscribed Job Event** – a Subscribed Event that is a Job Event.
- 403 **Subscribed Printer Event** – a Subscribed Event that is a Printer Event.
- 404 **Subscribing Client** – The client that creates the Subscription Object.
- 405 **Subscription Attributes Group** – The attributes group in a response that contains Subscription Object
406 attributes.
- 407 **Subscription Creation Operation** – An operation that creates a Subscription Object: Job Creation
408 operations, Create-Job-Subscriptions operation, Create-Printer-Subscriptions operation. In the context
409 of a Job Creation operation, a Subscription Creation Operation is the part of the Job Creation operation
410 that creates one or more Subscription objects. The Restart-Job operation [RFC2911] is not considered
411 a Subscription Creation Operation, since the Printer re-uses the Job’s existing Subscription Objects,
412 rather than creating any new Subscription Objects.
- 413 **Subscription Creation Request** – The request portion of a Subscription Creation Operation.
- 414 **Subscription Description Attributes** – Subscription Object attributes that a Printer supplies during a
415 Subscription Creation Operation.

416 **Subscription Object** – An object containing a set of attributes that indicate: the Notification Recipient
417 (for Push Delivery Method only), the Delivery Method, the Subscribed Events that cause the Printer to
418 deliver an Event Notification, and the information to include in an Event Notification.

419 **Subscription Template Attributes** – Subscription Object attributes that a client can supply in a
420 Subscription Creation Operation and associated Printer Object attributes that specify supported and
421 default values for the Subscription Object attributes.

422 **Subscription Template Attributes Group** – The attributes group in a request that contains
423 Subscription Object attributes that are Subscription Template Attributes.

424 **4 Object Relationships**

425 This section defines the object relationships between the Printer, Job, and Subscription Objects. It does
426 not define the implementation. For an illustration of these relationships, see Appendix 20.

427 **4.1 Printer and Per-Printer Subscription Objects**

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

430 **4.2 Printer, Job and Per-Job Subscription Objects**

- 431 1. A Printer object is associated with zero or more Job objects.
- 432 2. Each Job object is associated with exactly one Printer object.
- 433 3. A Job object is associated with zero or more Per-Job Subscription Objects.
- 434 4. Each Per-Job Subscription Object is associated with exactly one Job object.

435 **5 Subscription Object**

436 A Subscribing Client creates a Subscription Object with a Subscription Creation Operation in order to
437 indicate its interest in certain Events. See section 11 for a description of these operations. When an
438 Event occurs, the Subscription Object specifies to the Printer where to deliver Event Notifications for
439 Push Delivery Methods only, how to deliver them, and what to include in them. See section 9 for
440 details on the contents of an Event Notification.

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

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

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

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

450 **5.1 Rules for Support of Subscription Template Attributes**

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

454 Subscription Objects attributes that are Subscription Template Attributes conform to the following
455 rules:

- 456 1. Each attribute's name starts with the prefix string "notify-" and this document calls such
457 attributes "notify-xxx".
- 458 2. For each "notify-xxx" Subscription Object attribute defined in column 1 of Table 1 in section
459 5.3, Table 1 specifies corresponding Printer attributes: "notify-xxx-default", "notify-xxx-
460 supported", "yyy-supported" and "notify-max-xxx-supported" defined in column 2 of Table 1.
461 Note "xxx" stands for the same string in each case and "yyy" stands for some other string.
- 462 3. If a Printer supports "notify-xxx" in column 1 of Table 1, then the Printer MUST support all
463 associated attributes specified in column 2 of Table 1. For example, Table 1 shows that if the
464 Printer supports "notify-events", it MUST support "notify-events-default", "notify-events-
465 supported" and "notify-max-events-supported".
- 466 4. If a Printer does not support "notify-xxx" in column 1 of Table 1, then the Printer MUST NOT
467 support any associated "notify-yyy" attributes specified in column 2 of Table 1. For example,
468 Table 1 shows that if the Printer doesn't support "notify-events", it MUST NOT support "notify-
469 events-default", "notify-events-supported" and "notify-max-events-supported". Note this rule
470 does not apply to attributes whose names do not start with the string "notify-" and are thus
471 defined in another object and used by other attributes.
- 472 5. Most "notify-xxx" attributes have a corresponding "yyy-supported" attribute that specifies the
473 supported values for "notify-xxx". Column 2 of Table 1 specifies the name of each "yyy-
474 supported" attribute. The naming rules of IPP/1.1 (see [RFC2911]) are used when "yyy-
475 supported" is "notify-xxx-supported".

476 6. Some “notify-xxx” attributes have a corresponding “notify-xxx-default” attribute that specifies
477 the value for “notify-xxx” if the client does not supply it. Column 2 of Table 1 specifies the
478 name of each “notify-xxx-default” attribute. The naming rules of IPP/1.1 (see [RFC2911]) are
479 used.

480 If a client wishes to present an end user with a list of supported values from which to choose, the client
481 SHOULD query the Printer for its supported value attributes. The client SHOULD also query the
482 default value attributes. If the client then limits selectable values to only those values that are
483 supported, the client can guarantee that the values supplied by the client in the create request all fall
484 within the set of supported values at the Printer. When querying the Printer, the client MAY enumerate
485 each attribute by name in the Get-Printer-Attributes Request, or the client MAY just supply the
486 ‘subscription-template’ group name in order to get the complete set of supported attributes (both
487 supported and default attributes – see section 11.2.3).

488 5.2 Rules for Processing Subscription Template Attributes

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

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

- 496 1. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer supports it and
497 its value, the Printer MUST populate the attribute on the created Subscription Object.
- 498 2. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer doesn’t
499 support it or its value, the Printer MUST NOT populate the attribute on the created Subscription
500 Object with it. The Printer MUST do one of the following:
 - 501 a) If the value of the “notify-xxx” attribute is unsupported, the Printer MUST return the attribute
502 with its value in the Subscription Attributes Group of the response.
 - 503 b) If “notify-xxx” is an unsupported attribute, the Printer MUST return the attribute in the
504 Subscription Attributes Group of the response with the ‘unsupported’ out-of-band value.

505 Note: The rules of this step are the same as for Unsupported Attributes [RFC2911] section 3.1.7.
506 except that the unsupported attributes are returned in the Subscription Attributes Group rather than
507 the Unsupported Attributes Group because Subscription Creation Operations can create more than
508 one Subscription Object).

- 509 3. If a client is REQUIRED to supply a “notify-xxx” attribute from column 1 of Table 1 and the
510 Printer doesn’t support the supplied value, the Printer MUST NOT create a Subscription Object.
511 The rules for Unsupported Attributes in step #2 still apply.

- 512 4. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 and the attribute is
513 REQUIRED for the client to supply, the Printer MUST reject the Subscription Creation Operation
514 (including Job Creation operations) without creating a Subscription Object, and MUST return in
515 the response:
- 516 c) the status code ‘client-error-bad-request’ AND
- 517 d) no Subscription Attribute Groups.
- 518 5. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 that is OPTIONAL for
519 the client to supply, and column 2 of Table 1 either:
- 520 a) specifies a “notify-xxx-default” attribute, the Printer MUST behave as if the client had supplied
521 the “notify-xxx-default” attribute (see step #1) and populate the Subscription object with the
522 value of the “notify-xxx-default” attribute as part of the Subscription Creation operation (unlike
523 Job Template attributes where the Printer does not populate the Job object with defaults – see
524 [RFC2911]) OR
- 525 b) does not specify a “notify-xxx-default” attribute, the Printer MUST populate the “notify-xxx”
526 attribute on the Subscription Object according to the definition of the “notify-xxx” attribute in a
527 section 5.3. For some attributes, the “notify-xxx” is populated with the value of some other
528 attribute, and for others, the “notify-xxx” is NOT populated on the Subscription object at all.
- 529 6. A Printer MUST create a Subscription Object for each Subscription Template Attributes group in a
530 request unless the Printer:
- 531 a) encounters some attributes in a Subscription Template Attributes Group that require the Printer
532 not to create the Subscription Object OR
- 533 b) would create a Per-Job Subscription Object when it doesn’t have space for another Per-Job
534 Subscription Object OR
- 535 c) would create a Per-Printer Subscription Object when it doesn’t have space for another Per-
536 Printer Subscription Object.
- 537 7. A response MUST contain one Subscription Attributes Group for each Subscription Template
538 Attributes Group in the request (and in the same order) whether the Printer creates a Subscription
539 Object from the Subscription Template Attributes Group or not. However, the attributes in each
540 Subscription Attributes Group can be in any order.
- 541 8. The Printer MUST populate each Subscription Attributes Group of the response such that each
542 contains:
- 543 a) the “notify-subscription-id” attribute (see section 5.4.1), if and only if the Printer creates a
544 Subscription Object.

- 545 b) the “notify-lease-duration” attribute (see section 5.3.8), if and only if the Printer creates a Per-
546 Printer Subscription Object. The value of this attribute is the value of the Subscription Object’s
547 “notify-lease-duration” attribute. This value MAY be different from the client-supplied value
548 (see section 5.3.8). If a client supplies this attribute in the creation of a Per-Job Subscription
549 Object, it MUST appear in this group with the out-of-band value ‘unsupported’ to indicate that
550 the Printer doesn’t support it in this context.
- 551 c) all of the unsupported Subscription Template Attributes from step #2. Note, they are not
552 returned in the Unsupported Attributes Group in order to separate the unsupported attributes for
553 each Subscription Object.
- 554 d) the “notify-status-code” attribute if the Printer does not create the Subscription Object or if
555 there are unsupported attributes from step #2. The possible values of the “notify-status-code”
556 attribute are shown below (see section 13 for more details). The Printer returns the first value
557 in the list below that describes the status.
- 558 ‘client-error-uri-scheme-not-supported’: the Subscription Object was not created because
559 the scheme of the “notify-recipient-uri” attribute is not supported. See section 13.1 for
560 more details about this status code. See step #3 in this section for the case that causes
561 this error, and the resulting step #6a) that causes the Printer not to create the
562 Subscription Object.
- 563 ‘client-error-attributes-or-values-not-supported’: the Subscription Object was not created
564 because the method of the “notify-pull-method” attribute is not supported. See section
565 13.1 for more details about this status code. See step #3 in this section for the case that
566 causes this error, and the resulting step #6a) that causes the Printer not to create the
567 Subscription Object.
- 568 ‘client-error-too-many-subscriptions’: the Subscription Object was not created because the
569 Printer has no space for additional Subscription Objects. The client SHOULD try again
570 later. See section 13.3 for more details about this status code. See steps #6b) and #6c)
571 in this section for the cases that causes this error.
- 572 ‘successful-ok-too-many-events’: the Subscription Object was created without the “notify-
573 events” values included in this Subscription Attributes Group because the “notify-
574 events” attribute contains too many values. See section 13.4 for more details about this
575 status code. See step #2 in this section and section 5.3.3 for the cases that cause this
576 status code.
- 577 ‘successful-ok-ignored-or-substituted-attributes’: the Subscription Object was created but
578 some supplied Subscription Template Attributes are unsupported. These unsupported
579 attributes are also in the Subscription Attributes Group. See section 13.5 for more
580 details about this status code. See step #2 in this section for the cases that cause this
581 status code.

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

588 5.3 Subscription Template Attributes

589 This section contains the Subscription Template Attributes defined for the Subscription and Printer
590 objects.

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

- 592 - **Attribute in Subscription Object:** the name and attribute syntax of each Subscription Object
593 Attribute that is a Subscription Template Attribute
- 594 - **Default and Supported Printer Attributes:** the default attribute and supported Printer
595 attributes that are associated with the attribute in column 1.

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

598 For Push Delivery Methods, a Printer MUST support all attributes in Table 1 below except for “notify-
599 pull-method” and “notify-attributes” (and “notify-pull-method-supported” and “notify-attributes-
600 supported”). For Pull Delivery Methods, a Printer MUST support all attributes in Table 1 below
601 except for “notify-recipient-uri” and “notify-attributes” (and “notify-schemes-supported” and “notify-
602 attributes-supported”). If a Printer supports both Push and Pull Delivery Methods, then it MUST
603 support both “notify-recipient-uri” and “notify-pull-method” attributes.

604 For Pull Delivery Methods, a client MUST supply “notify-recipient-uri” and MAY omit any of the rest
605 of the attributes in column 1 of Table 1 in a Subscription Creation Request. For Push Delivery
606 Methods, a client MUST supply “notify-pull-method” and MAY omit any of the rest of the attributes in
607 column 1 of Table 1 in a Subscription Creation Request. A client MUST NOT supply both “notify-
608 recipient-uri” and “notify-pull-method” attributes in the same Subscription Creation Request.

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

615

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))	

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

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

618 **5.3.1 notify-recipient-uri (uri)**

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

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

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

627 The URI scheme of the value of this attribute on a Subscription object MUST be a value of the “notify-
 628 schemes-supported (1setOf uriScheme)” Printer attribute (see section 5.3.1.1). Note: According to
 629 [RFC2396] the “:” terminates the scheme and so is not part of the scheme. Therefore, values of the
 630 “notify-schemes-supported” Printer attribute do not include the “:” character.

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

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

635 **5.3.1.1 notify-schemes-supported (1setOf uriScheme)**

636 This attribute contains the URI schemes supported in the “notify-recipient-uri” Subscription Template
637 attribute. See sections 5.1 and 5.2 for the behavior of “xxx-supported” Subscription Template Printer
638 attributes.

639 **5.3.2 notify-pull-method (type2 keyword)**

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

641 Since a Printer MUST support the ‘ippget’ Pull Delivery Method [ipp-get-method] (see section 15), a
642 Printer MUST support this attribute and return the value as supplied by the client in any operation
643 response that includes this attribute.

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

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

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

651 **5.3.2.1 notify-pull-method-supported (1setOf type2 keyword)**

652 See sections 5.1 and 5.2 for the behavior of “xxx-supported” Subscription Template Printer attributes.

653 **5.3.3 notify-events (1setOf type2 keyword)**

654 This attribute contains a set of Subscribed Events. When an Event occurs and it “matches” a value of
655 this attribute, the Printer delivers an Event Notification using information in the Subscription Object.
656 The details of “matching” are described subsection 5.3.3.5.

657 A Printer MUST support this attribute.

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

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

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

669 **5.3.3.1 notify-events-default (1setOf type2 keyword)**

670 See sections 5.1 and 5.2 for the behavior of “xxx-default” Subscription Template Printer attributes.

671 **5.3.3.2 notify-events-supported (1setOf type2 keyword)**

672 See sections 5.1 and 5.2 for the behavior of “xxx-supported” Subscription Template Printer attributes.

673 **5.3.3.3 notify-max-events-supported (integer(2:MAX))**

674 This attribute specified the maximum number of events that the Printer supports for the “notify-events”
675 Subscription Template attribute. See sections 5.1 and 5.2 for the behavior of “xxx-supported”
676 Subscription Template Printer attributes.

677 **5.3.3.4 Standard Values for Subscribed Events**

678 Each value of this attribute is a keyword and it specifies a Subscribed Event that represents certain
679 changes. Some keywords represent a subset of changes of another keyword, e.g., ‘job-completed’ is an
680 Event value which is a sub-value of ‘job-state-change’. See section 5.3.3.5 for the case where this
681 attribute contains both a value and a sub-value.

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

683 A Printer MUST support the Events indicated as “REQUIRED” and MAY support the Events
684 indicated as “OPTIONAL”.

685 **5.3.3.4.1 No Events**

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

687 **‘none’**: REQUIRED – no Event Notifications for any Events. As the sole value of “notify-events-
688 supported”, this value means that the Printer does not support the delivery of Event Notifications.
689 As the sole value of “notify-events-default”, this value means that a client MUST specify the
690 “notify-events” attribute in order for a Subscription Creation Operation to succeed. If the Printer
691 receives this value as the sole value of a Subscription Creation Operation, it does not create a
692 Subscription Object. If a Printer receives this value with other values of a Subscription Creation
693 Operation, the Printer MUST treat this value as an unsupported value.

694 **5.3.3.4.2 Subscribed Printer Events**

695 The standard keyword values for Subscribed Printer Events are:

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

699
700 This Subscribed Event value has the following sub-values: ‘printer-restarted’ and ‘printer-
701 shutdown’. A client can listen for any of these sub-values if it doesn’t want to listen to all printer-
702 state changes:

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

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

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

707 **‘printer-config-changed’**: OPTIONAL – when the configuration of a Printer has changed, i.e., the
708 value of the “printer-message-from-operator” or any “configuration” Printer attribute has changed.
709 A “configuration” Printer attribute is an attribute which can change value because of some human
710 interaction either direct or indirect, and which is not covered by one of the other Events in this
711 section. Examples of “configuration” Printer attributes are any of the Job Template attributes,
712 such as “xxx-supported”, “xxx-ready” and “xxx-default”. The client has to perform a Get-Printer-
713 Attributes to find out the new values of these changed attributes. This Event is useful for GUI
714 clients and drivers to update the available printer capabilities to the user.

715
716 This Event value has the following sub-values: ‘printer-media-changed’ and ‘printer-finishings-
717 changed’. A client can listen for any of these sub-values if it doesn’t want to listen to all printer-
718 configuration changes:

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

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

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

734 **5.3.3.4.3 Subscribed Job Events**

735 The standard keyword values for Subscribed Job Events are:

736 **‘job-state-changed’**: REQUIRED – the job has changed from any state to any other state.
737 Specifically, the Printer delivers this Event whenever the value of the “job-state” attribute or “job-
738 state-reasons” attribute changes. When a Job is removed from the Job Retention or Job History
739 phases (see [RFC2911] section 4.3.7.1), no Event is generated.
740

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

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

748 **‘job-completed’**: REQUIRED – the job has reached one of the completed states, i.e., the
749 value of the job’s “job-state” attribute has changed to: ‘completed’, ‘aborted’, or
750 ‘canceled’. The Job’s “time-at-completed” and “date-time-at-completed” (if supported)
751 attributes are set (see [RFC2911] section 4.3.14). When a Job completes, a Notification
752 Recipient MAY query the Job using the Get-Job-Attributes operation. To allow such a
753 query, the Printer retains the Job in the Job Retention and/or the Job History phases (see
754 [RFC2911] section 4.3.7.1) for a suitable amount of time that depends on
755 implementation and the Delivery Methods supported. The Printer also delivers this
756 Event when a Job is removed with the Purge-Job operation (see [RFC2911] section
757 3.2.9). In this case, the Event Notification MUST report the ‘job-state’ as ‘canceled’
758 and the Job object is no longer present for query.

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

761 **‘job-config-changed’**: OPTIONAL – when the configuration of a job has changed, i.e., the value of
762 the “job-message-from-operator” or any of the “configuration” Job attributes have changed. A
763 “configuration” Job attribute is an attribute that can change value because of some human
764 interaction either direct or indirect. Examples of “configuration” Job attributes are any of the job
765 template attributes and the “job-name” attribute. The client performs a Get-Job-Attributes to find
766 out the new values of the changed attributes. This Event is useful for GUI clients and drivers to
767 update the job information to the user.

768 **‘job-progress’**: OPTIONAL – when the Printer has completed Printing a sheet. See the separate
769 [RFC3381] specification for additional attributes that a Printer MAY deliver in an Event
770 Notification caused by this Event. The “notify-time-interval” attribute affects this Event by
771 causing the Printer NOT to deliver an Event Notification every time a ‘job-progress’ Events
772 occurs. See section 5.3.9 for full details.

773 **5.3.3.5 Rules for Matching of Subscribed Events**

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

777 **5.3.3.5.1 Rules for Matching of Printer Events**

778 Given that the Printer causes Printer Event E to occur, for each Per-Job or Per-Printer Subscription S
779 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
780 S, the Printer MUST generate an Event Notification.

781 Consider the example. There are three Subscription Objects each with the Subscribed Printer Event
782 ‘printer-state-changed’. Subscription Object A is a Per-Printer Subscription Object. Subscription
783 Object B is a Per-Job Subscription Object for Job 1, and Subscription Object C is a Per-Job
784 Subscription Object for Job 2. When the Printer enters the ‘stopped’ state, the Printer delivers an

785 Event Notification to the Notification Recipients of Subscription Objects A, B, and C because this is a
786 Printer Event. Note if Job 1 has already completed, the Printer would not deliver an Event Notification
787 for its Subscription Object, even if Job 1 is retained in the Job Retention and/or the Job History phases
788 (see [RFC2911] section 4.3.7.1).

789 5.3.3.5.2 Rules for Matching of Job Events

790 Given that Job J causes Job Event E to occur:

- 791 1. For each Per-Printer Subscription S in the Printer, if E equals a value of this attribute in S or E is
792 a sub-value of a value of this attribute in S, the Printer MUST generate an Event Notification.
- 793 2. For each Per-Job Subscription S associated with Job J, if E equals a value of this attribute in S
794 or E is a sub-value of a value of this attribute in S, the Printer MUST generate an Event
795 Notification.
- 796 3. For each Per-Job Subscription S that is NOT associated Job J, if E equals a value of this
797 attribute in S or E is a sub-value of a value of this attribute in, the Printer MUST NOT generate
798 an Event Notification from S.

799 Consider the example: There are three Subscription Objects listening for the Job Event ‘job-
800 completed’. Subscription Object A is a Per-Printer Subscription Object. Subscription Object B is a
801 Per-Job Subscription Object for Job 1, and Subscription Object C is a Per-Job Subscription Object for
802 Job 2. In addition, Per-Printer Subscription Object D is listening for the Job Event ‘job-state-changed’.
803 When Job 1 completes, the Printer delivers an Event Notification to the Notification Recipient of
804 Subscription Object A (because it is Per-Printer) and Subscription Object B because it is a Per-Job
805 Subscription Object associated with the Job generating the Event. The Printer also delivers an Event
806 Notification to the Notification Recipient of Subscription Object D because ‘job-completed’ is a sub-
807 value of ‘job-state-changed’ – the value that Subscription Object D is listening for. The Printer does
808 not deliver an Event Notification to the Notification Recipients of Subscription Object C because it is a
809 Per-Job Subscription Object associated with some Job other than the Job generating the Event.

810 5.3.3.5.3 Special Cases for Matching Rules

811 This section contains rule for special cases.

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

818 If an Event matches two values of this “notify-events” attribute in a single Subscription object (e.g., a
819 value and its sub-value), a Printer MAY deliver one Event Notification for each matched value in the

820 Subscription Object or it MAY deliver only one Event Notification per Subscription Object. The rules
821 in sections 5.3.3.5.1 and 5.3.3.5.2 are purposefully flexible about the number of Event Notifications
822 sent when Event E matches two or more values in a Subscription Object.

823 Consider the example: There are two Per-Printer Subscription Objects when a Job completes.
824 Subscription Object A has the Subscribed Job Event 'job-state-changed'. Subscription Object B has
825 the Subscribed Job Events 'job-state-changed' and 'job-completed'. The Printer delivers an Event
826 Notification to the Notification Recipient of Subscription Object A with the value of 'job-state-
827 changed' for the "notify-subscribing-event" attribute. The Printer delivers either one or two Event
828 Notifications to the Notification Recipient of Subscription Object B, depending on implementation. If
829 it delivers two Event Notifications, one has the value of 'job-state-changed' for the "notify-
830 subscribing-event" attribute, and the other has the value of 'job-completed' for the "notify-
831 subscribing-event" attribute. If it delivers one Event Notification, it has the value of either 'job-state-
832 changed' or 'job-completed' for the "notify-subscribing-event" attribute, depending on
833 implementation. The algorithm for choosing such a value is implementation dependent.

834 **5.3.4 notify-attributes (1setOf type2 keyword)**

835 This attribute contains a set of attribute names. When a Printer delivers a Machine Consumable Event
836 Notification, it includes a fixed set of attributes (see section 9.1). If this attribute is present and the
837 Event Notification is Machine Consumable, the Printer also includes the attributes specified by this
838 attribute.

839 A Printer MAY support this attribute.

840 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
841 this attribute in Subscription Creation Operation or the Printer does not support this attribute, the
842 Subscription Object either (1) MAY contain the "notify-attributes" attribute with a 'none' value or (2)
843 NEED NOT contain the attribute at all. There is no "notify-attributes-default" Printer attribute.

844 Each keyword value of this attribute on a Subscription Object MUST be a value of the "notify-
845 attributes-supported (1setOf type2 keyword)" Printer attribute (see section 5.3.4.1). The "notify-
846 attributes-supported" MAY contain any Printer attribute, Job attribute or Subscription Object attribute
847 that the Printer supports in an Event Notification. It MUST NOT contain any of the attributes in
848 Section 9.1 that a Printer automatically puts in an Event Notification; it would be redundant. If a client
849 supplies an attribute in Section 9.1, the Printer MUST treat it as an unsupported attribute value of the
850 "notify-attributes" attribute.

851 The following rules apply to each keyword value N of the "notify-attributes" attribute: If the value N
852 names:

853 a) a Subscription attribute, the Printer MUST use the attribute N in the Subscription Object that is
854 being used to generate the Event Notification.

855 b) a Job attribute and the Printer is generating an Event Notification from a Per-Job Subscription
856 Object S, the Printer MUST use the attribute N in the Job object associated with S.

857 c) a Job attribute and the Printer is generating an Event Notification from a Per-Printer Subscription
858 Object and the Event is:

- 859 • a Job Event, the Printer MUST use the attribute N in the Job object that caused the Event.
- 860 • a Printer Event, the Printer MUST use the attribute N in the active Job.

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

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

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

867 **5.3.4.1 notify-attributes-supported (1setOf type2 keyword)**

868 See sections 5.1 and 5.2 for the behavior of “xxx-supported” Subscription Template Printer attributes.

869 **5.3.5 notify-user-data (octetString(63))**

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

- 872 - the identity of the Subscriber
- 873 - a path or index to some Subscriber information
- 874 - a key that identifies to the Notification Recipient the ultimate recipient of the Event
875 Notification
- 876 - the id for a Notification Recipient that had previously registered with an Instant Messaging
877 Service

878 A Printer MUST support this attribute.

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

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

886 **5.3.6 notify-charset (charset)**

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

889 A Printer MUST support this attribute.

890 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
891 this attribute in Subscription Creation Operation or supplies an unsupported value, the Printer MUST
892 populate this attribute in the Subscription Object with the value of the “attributes-charset” operation
893 attribute, which is a REQUIRED attribute in all IPP requests (see [RFC2911]). If the value of the
894 “attributes-charset” attribute is unsupported, the Printer MUST populate this attribute in the
895 Subscription Object with the value of the Printer’s “charset-configured” attribute. There is no “notify-
896 charset-default” Printer attribute.

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

899 **5.3.7 notify-natural-language (naturalLanguage)**

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

903 A Printer MUST support this attribute.

904 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
905 this attribute in Subscription Creation Operation or supplies an unsupported value, the Printer MUST
906 populate this attribute in the Subscription Object with the value of the “attributes-natural-language”
907 operation attribute, which is a REQUIRED attribute in all IPP requests (see [RFC2911] section 3.1.4).
908 If the value of the “attributes-natural-language” attribute is unsupported, the Printer MUST populate
909 this attribute in the Subscription Object with the value of the Printer’s “natural-language-configured”
910 attribute (see [RFC2911] section 4.4.19). There is no “notify-natural-language-default” Printer
911 attribute.

912 The value of this attribute on a Subscription Object MUST be a value of the “generated-natural-
913 language-supported (1setOf type2 naturalLanguage)” Printer attribute (see [RFC2911] section 4.4.20).

914 **5.3.8 notify-lease-duration (integer(0:67108863))**

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

919 This attribute is not present on a Per-Job Subscription Object because the Subscription Object lasts
920 exactly as long as the associated Job object. See discussion of the ‘job-completed’ event in section
921 5.3.3.4.3 about retention of the Job object after completion.

922 A Printer MUST support this attribute.

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

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

933 After the Printer has populated this attribute with a supported value, the value represents the “granted
934 duration” of the lease in seconds and the Printer updates the value of the Subscription Object’s “notify-
935 lease-expiration-time” attribute as specified in section 5.4.3.

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

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

941 Note: The maximum value 67,108,863 is 2 raised to the 26 power minus 1 and is about 2 years in
942 seconds. The value is considerably less than MAX so that there is virtually no chance of an overflow
943 when the Printer adds it to the Printer’s “printer-up-time” attribute value (see [RFC2911] section
944 4.4.29) to produce the “notify-lease-expiration-time” Subscription Description attribute value (see
945 section 5.4.3).

946 **5.3.8.1 notify-lease-duration-default (integer(0:67108863))**

947 See sections 5.1 and 5.2 for the behavior of “xxx-default” Subscription Template Printer attributes.

948 **5.3.8.2 notify-lease-duration-supported (1setOf (integer(0: 67108863) |** 949 **rangeOfInteger(0:67108863)))**

950 See sections 5.1 and 5.2 for the behavior of “xxx-supported” Subscription Template Printer attributes.

951 **5.3.9 notify-time-interval (integer(0:MAX))**

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

957 The Printer MUST support this attribute if and only if the Printer supports the 'job-progress' Event.

958 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
959 this attribute in the Subscription Creation Operation, the Subscription Object either (1) MAY contain
960 the "notify-time-interval" attribute with a '0' value or (2) NEED NOT contain this attribute at all.
961 There is no "notify-time-interval-default" Printer attribute.

962 There is no "notify-time-interval-supported" Printer attribute.

963 If the 'job-progress' Event occurs and a Subscription Object contains the 'job-progress' Event as a
964 value of the 'notify-events' attribute, there are two cases to consider:

- 965 1. This attribute is not present on the Subscription Object or has the value of 0. The Printer MUST
966 generate and deliver an Event Notification (as is the case with other Events).
- 967 2. This attribute is present with a nonzero value of N:
 - 968 a) If the Printer has not sent an Event Notification for the 'job-progress' Event for the associated
969 Subscription Object within the past N seconds, the Printer MUST deliver an Event Notification
970 for the Event that just occurred. Note when the Printer completes the first page of a Job, this
971 rule implies that the Printer delivers an Event Notification for a Per-Job Subscription Object.
 - 972 b) Otherwise, the Printer MUST NOT generate or deliver an Event Notification for the associated
973 Subscription Object. The Printer MUST NOT increase the value of the "notify-sequence-
974 number" Subscription Object attribute (i.e., the sequence of values of the "notify-sequence-
975 number" attribute counts the Event Notifications that the Printer sent and not the Events that do
976 not cause an Event Notification to be sent).

977 It is RECOMMENDED that a Subscribing Client use this attribute when it subscribes to the 'job-
978 progress' Event, and that the value be sufficiently large to limit the frequency with which the Printer
979 delivers Event Notifications requests.

980 This attribute MUST NOT effect any Events other than 'job-progress'.

981 **5.4 Subscription Description Attributes**

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

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

985 A client MUST NOT supply the attributes in Table 2 in a Subscription Template Attributes Group of a
986 Subscription Creation Operation. There are no corresponding default or supported attributes.

987 **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))

988

989 **5.4.1 notify-subscription-id (integer (1:MAX))**

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

992 A Printer MUST support this attribute.

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

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

999 The 0 value is not permitted in order to allow for compatibility with “job-id” and with SNMP index
1000 values, which also cannot be 0.

1001 **5.4.2 notify-sequence-number (integer (0:MAX))**

1002 The value of this attribute indicates the number of times that the Printer has generated and attempted to
1003 deliver an Event Notification for this Subscription object. When an Event Notification contains this
1004 attribute, the Notification Recipient can determine whether it missed some Event Notifications (i.e.,
1005 numbers skipped) or received duplicates (i.e., same number twice).

1006 A Printer MUST support this attribute.

1007 When the Printer creates a Subscription Object, it MUST populate this attribute with a value of 0. This
1008 value indicates that the Printer has not sent any Event Notifications for this Subscription Object.

1009 Each time the Printer delivers a newly generated Event Notification, it MUST increase the value of this
1010 attribute by 1. For some Delivery Methods, the Printer MUST include this attribute in each Event
1011 Notification, and the value MUST be the value after it is increased by 1. That is, the value of this
1012 attribute in the first Event Notification after Subscription object creation MUST be 1, the second
1013 MUST be 2, etc. If a Delivery Method is defined such that the Notification Recipient returns a
1014 response, the Printer can re-try delivering an Event Notification a certain number of times with the
1015 same sequence number when the Notification Recipient fails to return a response.

1016 If a Subscription Object lasts long enough to reach the value of MAX, its next value MUST be 0, i.e., it
1017 wraps.

1018 **5.4.3 notify-lease-expiration-time (integer(0:MAX))**

1019 This attribute specifies the time in the future when the lease on the Per-Printer Subscription Object will
1020 expire, i.e. the “printer-up-time” value at which the lease will expire. If the value is 0, the lease never
1021 expires.

1022 A Printer MUST support this attribute.

1023 When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present – the
1024 Subscription Object lasts exactly as long as the associated Job object. See also the discussion of the
1025 ‘job-completed’ event in section 5.3.3.4.3 about retention of the Job object after completion so that a
1026 Notification Recipient can query the Job object after receiving the ‘job-completed’ Event Notification.

1027 When the Printer creates a Per-Printer Subscription Object, it populates this attribute with a value that
1028 is the sum of the values of the Printer’s “printer-up-time” attribute and the Subscription Object’s
1029 “notify-lease-duration” attribute with the following exception. If the value of the Subscription Object’s
1030 “notify-lease-duration” attribute is 0 (i.e., no expiration time), then the value of this attribute MUST be
1031 set to 0 (i.e., no expiration time).

1032 When the Printer powers up, it MUST populate this attribute in each persistent Subscription Object
1033 with a value using the algorithm in the previous paragraph.

1034 When the “printer-up-time” equals the value of this attribute, the Printer MUST delete the Subscription
1035 Object. A client can extend a lease of a Per-Printer Subscription Object with the Renew-Subscription
1036 operation (see section 11.2.6).

1037 Note: In order to compute the number of seconds remaining in a lease for a Per-Printer Subscription
1038 Object, a client can subtract the Subscription’s “notify-printer-up-time” attribute (see section 5.4.4)
1039 from the Subscription’s “notify-lease-expiration-time” attribute.

1040 5.4.4 notify-printer-up-time (integer(1:MAX))

1041 This attribute is an alias for the Printer's "printer-up-time" attribute " (see [RFC2911] section 4.4.29).
1042 In other words, when this attribute is queried with the Get-Subscriptions or Get-Subscription-Attributes
1043 operations (see sections 11.2.4 and 11.2.5), the value returned is the current value of the Printer's
1044 "printer-up-time" attribute, rather than the time at which the Subscription Object was created.

1045 A Printer MUST support this attribute.

1046 When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present. When
1047 the Printer creates a Per-Printer Subscription Object, this attribute MUST be present.

1048 Note: this attribute exists in a Per-Printer Subscription Object so that a client using the Get-
1049 Subscription-Attributes or Get-Subscription operations can convert the Per-Printer Subscription's
1050 "notify-lease-expiration-time" attribute to wall clock time with one request. If the value of the "notify-
1051 lease-expiration-time" attribute is not 0 (i.e., no expiration time), then the difference between the
1052 "notify-lease-expiration-time" attribute and the "notify-printer-up-time" is the remaining number of
1053 seconds on the lease from the current time.

1054 5.4.5 notify-printer-uri (uri)

1055 This attribute identifies the Printer object that created this Subscription Object.

1056 A Printer MUST support this attribute.

1057 During a Subscription Creation Operation, the Printer MUST populate this attribute with the value of
1058 the "printer-uri" operation attribute in the request. From the Printer URI, the client can, for example,
1059 determine what security scheme was used.

1060 5.4.6 notify-job-id (integer(1:MAX))

1061 This attribute specifies whether the containing Subscription Object is a Per-Job or Per-Printer
1062 Subscription Object, and for Per-Job Subscription Objects, it specifies the associated Job.

1063 A Printer MUST support this attribute.

1064 If this attribute is not present, the Subscription Object MUST be a Per-Printer Subscription. If this
1065 attribute is present, the Subscription Object MUST be a Per-Job Subscription Object and this attribute
1066 MUST identify the Job with which the Subscription Object is associated.

1067 Note: This attribute could be useful to a Notification Recipient that receives an Event Notification
1068 generated from a Per-Job Subscription Object and caused by a Printer Event. The Event Notification
1069 gives access to the Printer and the Subscription Object. The Event Notification gives access to the
1070 associated Job only via this attribute. See discussion of the 'job-completed' event in section 5.3.3.4.3
1071 about retention of the Job object after completion so that a Notification Recipient can query the Job
1072 object after receiving the 'job-completed' Event Notification.

1073 **5.4.7 notify-subscriber-user-name (name(MAX))**

1074 This attribute contains the name of the user who performed the Subscription Creation Operation.

1075 A Printer **MUST** support this attribute.

1076 The Printer **MUST** populate this attribute with the most authenticated printable name that it can obtain
 1077 from the authentication service over which the Subscription Creation Operation was received. The
 1078 Printer uses the same mechanism for determining the value of this attribute as it does for a Job's "job-
 1079 originating-user-name" (see [RFC2911] section 4.3.6).

1080 Note: To help with authentication, a Subscription Object may have additional private attributes about
 1081 the user, e.g., a credential of a principal. Such private attributes are implementation-dependent and not
 1082 defined in this document.

1083 **6 Printer Description Attributes Related to Notification**

1084 This section defines the Printer Description attributes that are related to Notification. Table 3 lists the
 1085 Printer Description attributes, indicates the Printer support required for conformance, and whether or
 1086 not the attribute is READ-ONLY (see section 3.1):

1087 **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

1088

1089 **6.1 printer-state-change-time (integer(1:MAX))**

1090 This **OPTIONAL** attribute records the most recent time at which the 'printer-state-changed' Printer
 1091 Event occurred whether or not any Subscription objects were listening for this event. This attribute
 1092 helps a client or operator to determine how long the Printer has been in its current state.

1093 A Printer **MAY** support this attribute and if so, the attribute **MUST** be READ-ONLY.

1094 On power-up, the Printer **MUST** populate this attribute with the value of its "printer-up-time" attribute,
 1095 so that it always has a value. Whenever the 'printer-state-changed' Printer Event occurs, the Printer
 1096 **MUST** update this attribute with the value of the Printer's "printer-up-time" attribute.

1097 6.2 printer-state-change-date-time (dateTime)

1098 This OPTIONAL attribute records the most recent time at which the ‘printer-state-changed’ Printer
1099 Event occurred whether or not there were any Subscription Objects listening for this event. This
1100 attribute helps a client or operator to determine how long the Printer has been in its current state.

1101 A Printer MAY support this attribute and if so, the attribute MUST be READ-ONLY.

1102 On power-up, the Printer MUST populate this attribute with the value of its “printer-current-time”
1103 attribute, so that it always has a value (see [RFC2911] section 4.4.30 on “printer-current-time”).

1104 Whenever the ‘printer-state-changed’ Printer Event occurs, the Printer MUST update this attribute with
1105 the value of the Printer’s “printer-current-time” attribute.

1106 7 New Values for Existing Printer Description Attributes

1107 This section contains those attributes for which additional values are added.

1108 7.1 operations-supported (1setOf type2 enum)

1109 The following “operation-id” values are added in order to support the new operations defined in this
1110 document:

1111 **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

1112 8 Attributes Only in Event Notifications

1113 This section contains those attributes that exist only in Event Notifications and do not exist in any
1114 objects.

1115 8.1 notify-subscribed-event (type2 keyword)

1116 This attribute indicates the Subscribed Event that caused the Printer to deliver this Event Notification.
1117 This attribute exists only in Event Notifications.

1118 This attribute MUST contain one of the values of the “notify-events” attribute in the Subscription
1119 Object, i.e., one of the Subscribed Event values. Its value is the Subscribed Event that “matches” the
1120 Event that caused the Printer to deliver this Event Notification. This Subscribed Event value may be
1121 identical to the Event or the Event may be a sub-value of the Subscribed Event. For example, the ‘job-
1122 completed’ Event (which is a sub-event of the ‘job-state-changed’ event) would cause the Printer to
1123 deliver an Event Notification for either the ‘job-completed’ or ‘job-state-changed’ Subscribed Events
1124 and to deliver the ‘job-completed’ or ‘job-state-changed’ value for this attribute, respectively. See
1125 section 5.3.3.5 for the “matching” rules of Subscribed Events and for additional examples.

1126 The Delivery Method Document specifies whether the Printer includes the value of this attribute in an
1127 Event Notification.

1128 **8.2 notify-text (text(MAX))**

1129 This attribute contains a Human Consumable text message (see section 9.2). This message describes
1130 the Event and is encoded as plain text, i.e., ‘text/plain’ with the charset specified by Subscription
1131 Object’s “notify-charset” attribute.

1132 The Delivery Method Document specifies whether the Printer includes this attribute in an Event
1133 Notification.

1134 **9 Event Notification Content**

1135 This section defines the Event Notification content that the Printer delivers when an Event occurs.

1136 When an Event occurs, the Printer MUST find each Subscription object whose “notify-events”
1137 attribute “matches” the Event. See section 5.3.3.5 for details on “matching”. For each matched
1138 Subscription Object, the Printer MUST create an Event Notification with the content and format that
1139 the Delivery Method Document specifies. The content contains the value of attributes specified by the
1140 Delivery Method Document. The Printer obtains the values immediately after the Event occurs. For
1141 example, if the “printer-state” attribute changes from ‘idle’ to ‘processing’, the Event ‘printer-state-
1142 changed’ occurs and the Printer puts various attributes into the Event Notification, including “printer-
1143 up-time” and “printer-state” with the values that they have immediately after the Event occurs, i.e., the
1144 value of “printer-state” is ‘processing’.

1145 **Event Notification Ordering:**

1146 When a Printer delivers Event Notifications, the Event Notifications from any given Subscription
1147 Object MUST be in time stamp order, i.e., in order of increasing “printer-up-time” attribute value in
1148 the Event Notification (see Table 5). These Event Notifications MAY be interleaved with those from
1149 other Subscription Objects, as long as those others are also in time stamp order. The Printer MUST
1150 observe these ordering requirements whether delivering multiple pending Events as multiple separate
1151 Event Notifications or together in a single Compound Event Notification.

1152 If a Subscribing Client wants the Printer to deliver certain Event Notifications in time stamp order, the
1153 Subscribing Client uses a single Subscription Object. Even so, depending on the underlying transport,
1154 the actual order that a Notification Recipient receives separate Event Notifications may differ from the
1155 order sent by the Printer (e.g., email).

1156 Example: Consider two Per-Printer Subscription Objects: SO1 and SO2. SO1 requests 'job-state-
1157 changed' events and SO2 requests 'printer-state-changed' events. The number in parens is the time
1158 stamp. The following Event Notification sequences are the only ones that conform to the ordering
1159 requirements for the Printer to deliver the Event Notifications:

1160 (a) SO1: 'job-created' (1000), SO1: 'job-stopped' (1005), SO1: 'job-completed' (1009), SO2:
1161 'printer-stopped' (1005)

1162 (b) SO1: 'job-created' (1000), SO1: 'job-stopped' (1005), SO2: 'printer-stopped' (1005), SO1: 'job-
1163 completed' (1009)

1164 (c) SO1: 'job-created' (1000), SO2: 'printer-stopped' (1005), SO1: 'job-stopped' (1005), SO1: 'job-
1165 completed' (1009)

1166 (d) SO2: 'printer-stopped (1005), SO1: 'job-created' (1000), SO1: 'job-stopped' (1005), SO1: 'job-
1167 completed' (1009)

1168 Examples (b) and (c) are interleaved; examples (a) and (d) are not interleaved and are not appropriate
1169 for some Delivery Methods.

1170 If two different Events occur simultaneously, or nearly so (e.g., "printer-up-time" has the same value
1171 for both), the Printer MUST create a separate Event Notification for each Event, even if the associated
1172 Subscription Object is the same for both Events. However, the Printer MAY combine these distinct
1173 Event Notifications into a single Compound Event Notification if the Delivery Method supports
1174 Compound Event Notifications. For example, suppose that two nearly-simultaneously Events
1175 represent two successive 'printer-state-changed' Events, one from 'idle' to 'processing' and another
1176 from 'processing' to 'stopped'. These two Events have the same name but are different instances of
1177 the Event. Then the Printer MUST create a separate Event Notification for each Event and SHOULD
1178 accurately report the "printer-state" of the first Event as 'processing' and the second Event as
1179 'stopped'.

1180 If a Subscription Object contains more than one Subscribed Event, and several Events occur in quick
1181 succession each matching a different Subscribed Event in the Subscription Object, the Printer MUST
1182 NOT generate a single Event Notification from several of these Events, but MAY combine distinct
1183 Event Notifications into a single Compound Event Notification if the Delivery Method supports
1184 Compound Event Notifications.

1185 After the Printer has created the Event Notification, the Printer delivers it via either a:

1186 Push Delivery Method: The Printer delivers the Event Notification shortly after an Event occurs.
1187 For some Push Delivery Methods, the Notification Recipient MUST deliver a response; for
1188 others it MUST NOT deliver a response.

1189 Pull Delivery Method: The Printer saves Event Notifications for some Event Life and expects
1190 the Notification Recipient to request Event Notifications. The Printer returns the Event
1191 Notifications in a response to such a request.

1192 If an error that meets the following conditions occurs, the Printer **MUST** cancel the Subscription
1193 Object.

1194 a) the error occurs during the delivering of an Event Notification generated from Subscription Object
1195 S **AND**

1196 b) the error would continue to occur every time the Printer delivers an Event Notification generated
1197 from Subscription Object S in the future.

1198 For example, if the address of the “notify-recipient-uri” of Subscription Object A references a non-
1199 existent target and the Printer determines this fact, it **MUST** delete Subscription Object A.

1200 The next two sections describe the values that a Printer delivers in the content of Machine Consumable
1201 and Human Consumable Event Notifications, respectively.

1202 The tables in the sub-sections of this section contain the following columns:

1203 a) **Source Value:** the name of the attribute that supplies the value for the Event Notification.
1204 Asterisks in this field refer to a note below the table.

1205 b) **Delivers:** if the Printer supports the value (column 1) on the Source Object (column 3) the
1206 Delivery Method **MUST** specify:

1207 **MUST:** that the Printer **MUST** deliver the value.

1208 **SHOULD:** either that the Printer **MUST** deliver the value or that the value is incompatible
1209 with the Delivery Method.

1210 **MAY:** that the Printer **MUST**, **SHOULD**, **MAY**, **MUST NOT**, **SHOULD NOT**, or **NEED**
1211 **NOT** deliver the value. The Delivery Method specifies the level of conformance for the
1212 Printer.

1213 c) **Source Object:** the object from which the source value comes. If the object is “Event
1214 Notification”, the Printer fabricates the value when it delivers the Event Notification. See
1215 section 8.

1216 9.1 Content of Machine Consumable Event Notifications

1217 This section defines the attributes that a Delivery Method **MUST** mention in a Delivery Method
1218 Document when specifying the Machine Consumable Event Notification’s contents.

1219 This document does not define the order of attributes in Event Notifications. However, Delivery
1220 Method Documents **MAY** define the order of some or all of the attributes.

1221 A Delivery Method Document MUST specify additional attributes (if any) that a Printer
1222 implementation delivers in a Machine Consumable Event Notification.

1223 Notification Recipients MUST be able to accept Event Notifications containing attributes they do not
1224 recognize. What a Notification Recipient does with an unrecognized attribute is implementation-
1225 dependent. Notification Recipients MAY attempt to display unrecognized attributes anyway or MAY
1226 ignore them.

1227 The next three sections define the attributes in Event Notification Contents that are:

- 1228 1. for all Events
- 1229 2. for Job Events only
- 1230 3. for Printer Events only

1231 9.1.1 Event Notification Content Common to All Events

1232 This section lists the attributes that a Delivery Method Document MUST specify for all Events.

1233 Table 5 lists potential values in each Event Notification.

1234 **Table 5 – Attributes in Event Notification Content**

Source Value	Delivers	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

1235 *A Printer MUST deliver this value only if and only if it supports the Printer’s “printer-current-time”
1236 attribute.
1237

1238 ** If the Subscription Object does not contain a “notify-user-data” attribute and the Delivery Method
 1239 Document REQUIRES the Printer to deliver the “notify-user-data” source value in the Event
 1240 Notification, the Printer MUST deliver an octet-string of length 0.

1241 *** The last three rows represent additional attributes that a client MAY request via the “notify-
 1242 attributes” attribute. A Printer MAY support the “notify-attributes” attribute. The Delivery Method
 1243 MUST say that the Printer MUST, SHOULD, MAY, MUST NOT, SHOULD NOT, or NEED NOT
 1244 support the “notify-attributes” attribute and specific values of this attribute. The Delivery Method
 1245 MAY say that support for the “notify-attributes” is conditioned on support of the attribute by the
 1246 Printer or it MAY say that Printer MUST support the “notify-attributes” attribute if the Printer supports
 1247 the Delivery Method.

1248 9.1.2 Additional Event Notification Content for Job Events

1249 This section lists the additional attributes that a Delivery Method Document MUST specify for Job
 1250 Events. See Table 6.

1251 **Table 6 – Additional Event Notification Content for Job Events**

Source Value	Delivers	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

1252 * The Printer MUST deliver the “job-impressions-completed” attribute in an Event Notification only
 1253 for the combinations of Events and Subscribed Events shown in Table 7.
 1254

1255 **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’

1256

1257 9.1.3 Additional Event Notification Content for Printer Events

1258 This section lists the additional attributes that a Delivery Method Document MUST specify for Printer
 1259 Events. See Table 8.

1260

Table 8 – Additional Event Notification Content for Printer Events

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

1261

1262

9.2 Content of Human Consumable Event Notification

1263

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.

1264

1265

1266

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

1267

a) the Printer name (see Table 9)

1268

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

1269

c) for Printer Events only:

1270

i) the Event (see Table 10) and/or Printer state information (see Table 14)

1271

d) for Job Events only:

1272

i) the job identity (see Table 12)

1273

ii) the Event (see Table 10) and/or Job state information (see Table 13)

1274

1275

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

1276

1277

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

1278

1279

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

1280

1281

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

1282

1283

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

1284

a) for all Events

1285

b) for Job Events only

1286

c) for Printer Events only

1287

1288

9.2.1 Event Notification Content Common to All Events

1289

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

1290 There is a separate table for each piece of information. Each row in the table represents a source value
 1291 for the information and the values are listed in order of preference, with the first one being the
 1292 preferred one. An implementation SHOULD use the source value from the earliest row in each table.
 1293 It MAY use the source value from another row instead, or it MAY combine the source values from
 1294 several rows. An implementation is free to determine the best way to present this information.

1295 In all tables of this section, all rows contain a “MAY” in order to state that the Delivery Method
 1296 specifies the conformance.

1297 Table 9 lists the source of the information for the Printer Name. The “printer-name” is more user-
 1298 friendly unless the Notification Recipient is in a place where the Printer name is not meaningful. For
 1299 example, an implementation could have the intelligence to deliver the value of the “printer-name”
 1300 attribute to a Notification Recipient that can access the Printer via value of the “printer-name” attribute
 1301 and otherwise deliver the value of the “notify-printer-uri” attribute.

1302 **Table 9 – Printer Name in Event Notification Content**

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

1303

1304 Table 10 lists the source of the information for the Event name. A Printer MAY combine this
 1305 information with state information described for Jobs in Table 13 or for Printers in Table 14.

1306 **Table 10 – Event Name in Event Notification Content**

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

1307

1308 Table 11 lists the source of the information for the time that the Event occurred. A Printer can deliver
 1309 this value only if it supports the Printer’s “printer-current-time” attribute. If a Printer does not support
 1310 the “printer-current-time” attribute, it MUST NOT deliver the “printer-up-time” value instead, since it
 1311 is not an allowed option for human consumable information.

1312 **Table 11 – Event Time in Event Notification Content**

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

1313

1314 9.2.2 Additional Event Notification Content for Job Events

1315 This section lists the source of the additional information that a Delivery Method MUST specify for
1316 Job Events.

1317 Table 12 lists the source of the information for the job name. The “job-name” is likely more
1318 meaningful to a user than “job-id”.

1319 **Table 12 – Job Name in Event Notification Content**

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

1320

1321 Table 13 lists the source of the information for the job state. If a Printer supports the “job-state-
1322 message” and “job-detailed-state-message” attributes, it SHOULD use those attributes for the job state
1323 information, otherwise, it should fabricate such information from the “job-state” and “job-state-
1324 reasons”. For some Events, a Printer MAY combine this information with Event information.

1325 **Table 13 – Job State in Event Notification Content**

Source Value	Delivers	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

1326

1327 9.2.3 Additional Event Notification Content for Printer Events

1328 This section lists the source of the additional information that a Delivery Method MUST specify for
1329 Printer Events.

1330 Table 14 lists the source of the information for the printer state. If a Printer supports the “printer-state-
1331 message”, it SHOULD use that attribute for the job state information, otherwise it SHOULD fabricate
1332 such information from the “printer-state” and “printer-state-reasons”. For some Events, a Printer MAY
1333 combine this information with Event information.

1334

Table 14 – Printer State in Event Notification Content

Source Value	Delivers	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

1335 **10 Delivery Methods**

1336 A Delivery Method is the mechanism, i.e., protocol, by which the Printer delivers an Event Notification
 1337 to a Notification Recipient. There are several potential Delivery Methods for Event Notifications,
 1338 standardized, as well as proprietary. This specification REQUIRES that the ‘ippget’ Pull Delivery
 1339 Method [ipp-get-method] be supported. Conforming implementations MAY support additional Push
 1340 or Pull Delivery Methods as well. This document does not define any of these delivery mechanisms.
 1341 Each Delivery Method MUST be defined in a Delivery Method Document that is separate from this
 1342 document. New Delivery Methods will be created as needed using an extension to the registration
 1343 procedures defined in [RFC2911]. Such documents are registered with IANA (see section 24.7.3).

1344 The following sorts of Delivery Methods are possible:

- 1345 – The Notification Recipient polls for Event Notifications at intervals directed by the Printer
- 1346 – The Printer delivers Event Notifications to the Notification Recipient using http as the transport.
- 1347 – The Printer delivers an email message.

1348 This section specifies how to define a Delivery Method Document and what to put in such a document.

1349 A Delivery Method Document MUST contain an exact copy of the following paragraph, caption and
 1350 table. In addition, column 2 of the table in the Delivery Method Document MUST contain answers to
 1351 questions in column 1 for the Delivery Method. Also, the Delivery Method document MUST contain a
 1352 reference to this document and call that reference [ipp-ntfy] because the table contains an [ipp-ntfy]
 1353 reference.

1354 If a Printer supports this Delivery Method, the following are its characteristics.

1355 **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 delivers 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?	

1356

1357 **11 Operations for Notification**

1358 This section defines all of the operations for Notification. Section 7.1 assigns the “operation-id” for
1359 each operation. The following two sub-sections define Subscription Creation Operations, and other
1360 operations.

1361 **11.1 Subscription Creation Operations**

1362 This section defines the Subscription Creation Operations. The first section on Create-Job-
1363 Subscriptions gives most of the information. The other Subscription Creation Operations refer to the
1364 section on Create-Job-Subscriptions, even though the Create-Job-Subscriptions operation is the only
1365 OPTIONAL operation in this document (see section 12).

1366 A Printer MUST support Create-Printer-Subscriptions and the Subscription Template Attributes Group
1367 in Job Creation operations. It MAY support Create-Job-Subscriptions operations.

1368 **11.1.1 Create-Job-Subscriptions Operation**

1369 The operation creates one or more Per-Job Subscription Objects. The client supplies one or more
1370 Subscription Template Attributes Groups each containing one or more of Subscription Template
1371 Attributes (defined in section 5.3).

1372 Except for errors, the Printer MUST create exactly one Per-Job Subscription Object from each
1373 Subscription Template Attributes Group in the request, even if the newly created Subscription Object
1374 would have identical behavior to some existing Subscription Object. The Printer MUST associate each
1375 newly created Per-Job Subscription Object with the target Job, which is specified by the “notify-job-id”
1376 operation attribute.

1377 The Printer MUST accept the request in any of the target job’s ‘not-completed’ states, i.e., ‘pending’,
1378 ‘pending-held’, ‘processing’, or ‘processing-stopped’. The Printer MUST NOT change the job’s “job-
1379 state” attribute because of this operation. If the target job is in any of the ‘completed’ states, i.e.,
1380 ‘completed’, ‘canceled’, or ‘aborted’, then the Printer MUST reject the request and return the ‘client-
1381 error-not-possible’ status code; the response MUST NOT contain any Subscription Attribute Groups.

1382 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section
1383 8.3) performing this operation MUST (1) be the job owner, (2) have Operator or Administrator access
1384 rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the
1385 Printer’s administrator-configured security policy to create Per-Job Subscription Objects for the target
1386 job. Otherwise the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-
1387 error-not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

1388 **11.1.1.1 Create-Job-Subscriptions Request**

1389 The following groups of attributes are part of the Create-Job-Subscriptions Request:

1390 Group 1: Operation Attributes

1391 Natural Language and Character Set:

1392 The "attributes-charset" and "attributes-natural-language" attributes as described in
1393 [RFC2911] section 3.1.4.1.

1394

1395 Target:

1396 The "printer-uri" attribute which defines the target for this operation as described in
1397 [RFC2911] section 3.1.5.

1398

1399 Requesting User Name:

1400 The "requesting-user-name" attribute SHOULD be supplied by the client as described in
1401 [RFC2911] section 8.3.

1402 **11.1.1.1.1 notify-job-id (integer(1:MAX))**

1403 The client MUST supply this attribute and it MUST specify the Job object to associate the
1404 Per-Job Subscription with. The value of "notify-job-id" MUST be the value of the "job-id" of
1405 the associated Job object. If the client does not supply this attribute, the Printer MUST reject
1406 this request with a 'client-error-bad-request' status code.

1407

1408 Group 2-N: Subscription Template Attributes

1409 For each occurrence of this group:

1410

1411 The client MUST supply one or more Subscription Template Attributes in any order. See
1412 section 5.3 for a description of each such attribute. See section 5.2 for details on processing
1413 these attributes.

1414 **11.1.1.2 Create-Job-Subscriptions Response**

1415 The Printer MUST return to the client the following sets of attributes as part of a Create-Job-
1416 Subscriptions response:

1417 Group 1: Operation Attributes

1418 Status Message:

1419 In addition to the REQUIRED status code returned in every response, the response
1420 OPTIONALLY includes a "status-message" (text(255)) and/or a "detailed-status-message"
1421 (text(MAX)) operation attribute as described in [RFC2911] sections 13 and 3.1.6.

1422

1423 In this group, the Printer can return any status codes defined in [RFC2911] and section 12.
1424 The following is a description of the important status codes:

1425
1426 **successful-ok:** the Printer created all Subscription Objects requested (see [RFC2911]).
1427 **successful-ok-ignored-subscriptions:** the Printer created some Subscription Objects
1428 requested but some failed. The Subscription Attributes Groups with a “notify-status-
1429 code” attribute are the ones that failed (see section 12.1).
1430 **client-error-ignored-all-subscriptions:** the Printer created no Subscription Objects
1431 requested and all failed. The Subscription Attributes Groups with a “notify-status-
1432 code” attribute are the ones that failed (see section 12.2).
1433 **client-error-not-possible:** For this operation and other Per-Job Subscription operations,
1434 this error can occur because the specified Job has already completed (see [RFC2911],
1435 whether or not the Job is retained in the Job Retention and/or Job History phases (see
1436 [RFC2911] section 4.3.7.1).
1437

1438 Natural Language and Character Set:

1439 The “attributes-charset” and “attributes-natural-language” attributes as described in
1440 [RFC2911] section 3.1.4.2.
1441

1442 Group 2: Unsupported Attributes

1443 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes. This group
1444 does not contain any unsupported Subscription Template Attributes; they are returned in the
1445 Subscription Attributes Group (see below).
1446

1447 Group 3-N: Subscription Attributes

1448 These groups MUST be returned unless the Printer is unable to interpret the entire request,
1449 e.g., the “status-code” parameter returned in Group 1 has the value: ‘client-error-bad-request’.
1450
1451 “notify-status-code” (type2 enum):
1452 Indicates the status of this subscription (see section 13 for the status code definitions).
1453 Section 5.2 defines when this attribute MUST be present in this group.
1454

1455 See section 5.2 for details on the contents of each occurrence of this group.
1456

1457 11.1.2 Create-Printer-Subscriptions operation

1458 The operation is identical to Create-Job-Subscriptions with exceptions noted in this section.

1459 The operation creates Per-Printer Subscription Objects instead of Per-Job Subscription Objects, and
1460 associates each newly created Per-Printer Subscription Object with the Printer specified by the
1461 operation target rather than with a specific Job.

1462 The Printer MUST accept the request in any of its states, i.e., ‘idle’, ‘processing’, or ‘stopped’. The
1463 Printer MUST NOT change its “printer-state” attribute because of this operation.

1464 Access Rights: To create Per-Printer Subscription Objects, the authenticated user (see [RFC2911]
1465 section 8.3) performing this operation MUST have (1) Operator or Administrator access rights for this
1466 Printer (see [RFC2911] sections 1 and 8.5), or (2) be otherwise authorized by the Printer’s
1467 administrator-configured security policy to create Per-Printer Subscription Objects for this Printer.
1468 Otherwise, the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-
1469 not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

1470 **11.1.2.1 Create-Printer-Subscriptions Request**

1471 The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.1) except that the
1472 Operation Attributes group MUST NOT contain the “notify-job-id” attribute. If the client does supply
1473 the “notify-job-id” attribute, then the Printer MUST treat it as any other unsupported Operation
1474 attribute and MUST return it in the Unsupported Attributes group.

1475 **11.1.2.2 Create-Printer-Subscriptions Response**

1476 The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.2).

1477 **11.1.3 Job Creation Operations – Extensions for Notification**

1478 This document extends the Job Creation operations (see section 3.2) to create Subscription Objects as a
1479 part of the operation.

1480 The Job Creation operations are identical to Create-Job-Subscriptions operation with exceptions noted
1481 in this section.

1482 Unlike the Create-Job-Subscriptions operation, a Job Creation operation associates the newly created
1483 Subscription Objects with the Job object created by this operation. The operation succeeds if and only
1484 if the Job creation succeeds. If the Printer does not create some or all of the requested Subscription
1485 Objects, the Printer MUST return a ‘successful-ok-ignored-subscriptions’ status-code instead of a
1486 ‘successful-ok’ status-code, but the Printer MUST NOT reject the operation because of a failure to
1487 create Subscription Objects.

1488 If the Job Creation operation includes a Job Template group, the client MUST supply it after the
1489 Operation Attributes group and before the first Subscription Template Attributes Group.

1490 If a Printer does not support this Notification specification, then it MUST treat the Subscription
1491 Attributes Group like an unknown group and ignore it (see [RFC2911] section 5.2.2). Because the
1492 Printer ignores the Subscription Attributes Group, it doesn’t return them in the response either, thus
1493 indicating to the client that the Printer doesn’t support Notification.

1494 After completion of a successful Job Creation operation, the Printer generates a 'job-created' event (see
1495 section 5.3.3.4.3).

1496 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section
1497 8.3) performing this operation MUST either have permission to create Jobs on the Printer or have
1498 Operator or Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5). Otherwise
1499 the Printer MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-
1500 authenticated', or 'client-error-not-authorized' status code as appropriate.

1501 **11.1.3.1 Job Creation Request**

1502 The groups for this operation are sufficiently different from the Create-Job-Subscriptions operation that
1503 they are all presented here. The following groups of attributes are supplied as part of a Job Creation
1504 Request:

1505 Group 1: Operation Attributes

1506 Same as defined in [RFC2911] for Print-Job, Print-URI, and Create-Job requests.

1507

1508 Group 2: Job Template Attributes

1509 The client OPTIONALLY supplies a set of Job Template attributes as defined in [RFC2911]
1510 section 4.2.

1511

1512 Group 3 to N: Subscription Template Attributes

1513 The same as Group 2-N in Create-Job-Subscriptions. See section 11.1.1.1.

1514 Group N+1: Document Content (Print-Job only)

1515 The client MUST supply the document data to be processed.

1516

1517 **11.1.3.2 Job Creation Response**

1518 The Printer MUST return to the client the following sets of attributes as part of a Print-Job, Print-URI,
1519 and Create-Job Response:

1520 Group 1: Operation Attributes

1521 Status Message:

1522

1523 As defined in [RFC2911] for Print-Job, Print-URI, and Create-Job requests.

1524

1525 In this group, the Printer can return any status codes defined in [RFC2911] and section 12.

1526 The following is a description of the important status codes:

1527

1528 **successful-ok:** the Printer created the Job and all Subscription Objects requested (see
1529 [RFC2911]).
1530 **successful-ok-ignored-subscriptions:** the Printer created the Job and not all of the
1531 Subscription Objects requested (see section 12.1). This status-code hides
1532 ‘successful-ok-xxx’ status-codes that could reveal problems in Job creation. The
1533 Printer MUST NOT return the ‘client-error-ignored-all-subscriptions’ status code for
1534 Job Creation operations because the Printer returns an error status-code only when it
1535 fails to create a Job.

1536

1537 Natural Language and Character Set:

1538 The “attributes-charset” and “attributes-natural-language” attributes as described in
1539 [RFC2911] section 3.1.4.2.

1540

1541 Group 2: Unsupported Attributes

1542 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes. This group
1543 does not contain any unsupported Subscription Template Attributes; they are returned in the
1544 Subscription Attributes Group (see below).

1545

1546 Group 3: Job Object Attributes

1547 The “job-id” of the Job Object just created, etc., as defined in [RFC2911] for Print-Job, Print-
1548 URI, and Create-Job requests.

1549

1550 Group 4 to N: Subscription Attributes

1551 These groups MUST be returned if and only if the client supplied Subscription Template
1552 Attributes and the operation was accepted.

1553

1554 See section 5.2 for details on the contents of each occurrence of this group.

1554

1555 **11.2 Other Operations**

1556 This section defines other operations on Subscription objects.

1557 **11.2.1 Restart-Job Operation – Extensions for Notification**

1558 The Restart-Job operation [RFC2911] is neither a Job Creation operation nor a Subscription Creation
1559 operation (see section 3.2). For the Restart-Job operation, the client MUST NOT supply any Job
1560 Subscription Attributes Groups. The Printer MUST treat any supplied Job Subscription Attributes as
1561 unsupported attributes.

1562 For this operation, the Printer does not return a job-id or any Subscription Attributes groups because
1563 the Printer reuses the existing Job object with the same job-id and the existing Per-Job Subscription
1564 Objects with the same subscription-ids. However, after successful completion of this operation, the
1565 Printer generates a ‘job-created’ event (see section 5.3.3.4.3).

1566 **11.2.2 Validate-Job Operation – Extensions for Notification**

1567 A client can test whether one or more Subscription Objects could be created using the Validate-Job
1568 operation. The client supplies one or more Subscription Template Attributes Groups (defined in
1569 section 5.3), just as in a Job Creation request.

1570 A Printer **MUST** support this extension to this operation.

1571 The Printer **MUST** accept requests that are identical to the Job Creation request defined in section
1572 11.1.3.1, except that the request **MUST NOT** contain document data.

1573 The Printer **MUST** return the same groups and attributes as the Print-Job operation (section 11.1.3.1)
1574 with the following exceptions. The Printer **MUST NOT** return a Job Object Attributes Group because
1575 no Job is created. The Printer **MUST NOT** return the “notify-subscription-id” attribute in any
1576 Subscription Attribute Group because no Subscription Object is created.

1577 If the Printer would succeed in creating a Subscription Object, the corresponding Subscription
1578 Attributes Group either has no ‘status-code’ attribute or a ‘status-code’ attribute with a value of
1579 ‘successful-ok-too-many-events’ or ‘successful-ok-ignored-or-substituted-attributes’ (see sections 5.2
1580 and 13). The status-codes have the same meaning as in Job Creation except the results state what
1581 “would happen”.

1582 The Printer **MUST** validate Subscription Template Attributes Groups in the same manner as the Job
1583 Creation operations.

1584 **11.2.3 Get-Printer-Attributes – Extensions for Notification**

1585 This operation is extended so that it returns Printer attributes defined in this document.

1586 A Printer **MUST** support this extension to this operation.

1587 In addition to the requirements of [RFC2911] section 3.2.5, a Printer **MUST** support the following
1588 additional values for the “requested-attributes” Operation attribute in this operation and return such
1589 attributes in the Printer Object Attributes group of its response.

- 1590 1. **Subscription Template Attributes:** Each supported attribute in column 2 of Table 1.
- 1591 2. **New Printer Description Attributes:** Each supported attribute in section 6.
- 1592 3. **New Group Name:** The ‘subscription-template’ group name, which names all supported
1593 Subscription Template Attribute in column 2 of Table 1. This group name is also used in the
1594 Get-Subscription-Attributes and Get-Subscriptions operation with an analogous meaning.
- 1595 4. **Extended Group Name:** The ‘all’ group name, which names all Printer attributes according to
1596 [RFC2911] section 3.2.5. In this extension ‘all’ names all attributes specified in [RFC2911]
1597 plus those named in items 1 and 2 of this list.

1598 **11.2.4 Get-Subscription-Attributes operation**

1599 This operation allows a client to request the values of the attributes of a Subscription Object.

1600 A Printer **MUST** support this operation.

1601 This operation is almost identical to the Get-Job-Attributes operation (see [RFC2911] section 3.3.4).
1602 The only differences are that the operation is directed at a Subscription Object rather than a Job object,
1603 and the returned attribute group contains Subscription Object attributes rather than Job object
1604 attributes.

1605 Access Rights: The authenticated user (see [RFC2911] section 8.3) performing this operation **MUST**
1606 (1) be the Subscription Object owner, (2) have Operator or Administrator access rights for this Printer
1607 (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer's administrator-
1608 configured security policy to query the Subscription Object for the target job. Otherwise the Printer
1609 **MUST** reject the operation and return: the 'client-error-forbidden', 'client-error-not-authenticated', or
1610 'client-error-not-authorized' status code as appropriate. Furthermore, the Printer's security policy
1611 **MAY** limit which attributes are returned, in a manner similar to the Get-Job-Attributes operation (see
1612 [RFC2911] end of section 3.3.4.2).

1613 **11.2.4.1 Get-Subscription-Attributes Request**

1614 The following groups of attributes are part of the Get-Subscription-Attributes request:

1615 Group 1: Operation Attributes

1616 Natural Language and Character Set:

1617 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1618 [RFC2911] 3.1.4.1.

1619

1620 Target:

1621 The "printer-uri" attribute which defines the target for this operation as described in
1622 [RFC2911] section 3.1.5.

1623

1624 Requesting User Name:

1625 The "requesting-user-name" attribute **SHOULD** be supplied by the client as described in
1626 [RFC2911] section 8.3.

1627 **11.2.4.1.1 "notify-subscription-id" (integer (1:MAX))**

1628 The client **MUST** supply this attribute. The Printer **MUST** support this attribute. This
1629 attribute specifies the Subscription Object from which the client is requesting attributes. If the
1630 client omits this attribute, the Printer **MUST** reject this request with the 'client-error-bad-
1631 request' status code.

1632 **11.2.4.1.2 “requested-attributes” (1setOf keyword)**

1633 The client **OPTIONALLY** supplies this attribute. The Printer **MUST** support this attribute.
1634 This attribute specifies the attributes of the specified Subscription Object that the Printer
1635 **MUST** return in the response. Each value of this attribute is either an attribute name (defined
1636 in sections 5.3 and 5.4) or an attribute group name. The attribute group names are:

- 1637
- 1638 - ‘subscription-template’: all attributes that are both defined in section 5.3 and present on
1639 the specified Subscription Object (column 1 of Table 1).
 - 1640 - ‘subscription-description’: all attributes that are both defined in section 5.4 and present
1641 on the specified Subscription Object (Table 2).
 - 1642 - ‘all’: all attributes that are present on the specified Subscription Object.
- 1643

1644 A Printer **MUST** support all these group names.

1645 If the client omits this attribute, the Printer **MUST** respond as if this attribute had been
1646 supplied with a value of ‘all’.

1647

1648 **11.2.4.2 Get-Subscription-Attributes Response**

1649 The Printer returns the following sets of attributes as part of the Get-Subscription-Attributes Response:

1650 Group 1: Operation Attributes

1651 Status Message:
1652 Same as [RFC2911].

1653

1654 Natural Language and Character Set:
1655 The “attributes-charset” and “attributes-natural-language” attributes as described in
1656 [RFC2911] section 3.1.4.2. The “attributes-natural-language” **MAY** be the natural language
1657 of the Subscription Object, rather than the one requested.

1658

1659 Group 2: Unsupported Attributes

1660 See [RFC2911] section 3.1.7 and section 3.2.5.2 for details on returning Unsupported
1661 Attributes.

1662

1663 The response **NEED NOT** contain the "requested-attributes" operation attribute with any
1664 supplied keyword values that were requested by the client but are not supported by the IPP
1665 object. If the Printer object does return unsupported attributes referenced in the "requested-
1666 attributes" operation attribute, the values of the “requested-attributes” attribute returned
1667 **MUST** include only the unsupported keywords that were requested by the client. If the client
1668 had requested a group name, such as 'all', the resulting unsupported attributes returned **MUST**
1669 **NOT** include attribute keyword names described in the standard but not supported by the
1670 implementation.

1671

1672 Group 3: Subscription Attributes

1673 This group contains a set of attributes with their current values. Each attribute returned in this
1674 group:

1675

1676 a) MUST be specified by the “requested-attributes” attribute in the request, AND

1677

b) MUST be present on the specified Subscription Object AND

1678

1679 c) MUST NOT be restricted by the security policy in force. For example, a Printer MAY
1680 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.

1681

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

1682

1683

1684 11.2.5 Get-Subscriptions operation

1685 This operation allows a client to retrieve the values of attributes of all Subscription Objects belonging
1686 to a Job or Printer.

1687

A Printer MUST supported this operation.

1688

This operation is similar to the Get-Subscription-Attributes operation, except that this Get-
1689 Subscriptions operation returns attributes from possibly more than one object.

1690

This operation is similar to the Get-Jobs operation (see [RFC2911] section 3.2.6), except that the
1691 operation returns Subscription Objects rather than Job objects.

1692

Access Rights: To query Per-Job Subscription Objects of the specified job (client supplied the “notify-
1693 job-id” operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)
1694 performing this operation MUST (1) be the Subscription Object owner, (2) have Operator or
1695 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise
1696 authorized by the Printer’s administrator-configured security policy to query the Subscription Object
1697 for the target job. To query Per-Printer Subscription Objects of the Printer (client omits the “notify-
1698 job-id” operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)
1699 performing this operation MUST (1) have Operator or Administrator access rights for this Printer (see
1700 [RFC2911] sections 1 and 8.5), or (2) be otherwise authorized by the Printer’s administrator-
1701 configured security policy to query Per-Printer Subscription Objects for the target Printer. Otherwise
1702 the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-not-
1703 authenticated’, or ‘client-error-not-authorized’ status code as appropriate. Furthermore, the Printer’s
1704 security policy MAY limit which attributes are returned, in a manner similar to the Get-Jobs and Get-
1705 Printer-Attributes operations (see [RFC2911] end of sections 3.2.6.2 and 3.2.5.2).

1706 **11.2.5.1 Get-Subscriptions Request**

1707 The following groups of attributes are part of the Get-Subscriptions request:

1708 Group 1: Operation Attributes

1709 Natural Language and Character Set:

1710 The “attributes-charset” and “attributes-natural-language” attributes as described in
1711 [RFC2911] section 3.1.4.1.

1712

1713 Target:

1714 The “printer-uri” attribute which defines the target for this operation as described in
1715 [RFC2911] section 3.1.5.

1716

1717 Requesting User Name:

1718 The “requesting-user-name” attribute SHOULD be supplied by the client as described in
1719 [RFC2911] section 8.3.

1720 **11.2.5.1.1 “notify-job-id” (integer(1:MAX))**

1721 If the client specifies this attribute, the Printer returns the specified attributes of all Per-Job
1722 Subscription Objects associated with the Job whose “job-id” attribute value equals the value
1723 of this attribute. If the client does not specify this attribute, the Printer returns the specified
1724 attributes of all Per-Printer Subscription Objects. Note: there is no way to get all Per-Job
1725 Subscriptions known to the Printer in a single operation. A Get-Jobs operation followed by a
1726 Get-Subscriptions operation for each Job will return all Per-Job Subscriptions.

1727 **11.2.5.1.2 “limit” (integer(1:MAX))**

1728 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. It
1729 is an integer value that determines the maximum number of Subscription Objects that a client
1730 will receive from the Printer even if the “my-subscriptions” attribute constrains which
1731 Subscription Objects are returned. The limit is a “stateless limit” in that if the value supplied
1732 by the client is ‘N’, then only the first ‘N’ Subscription Objects are returned in the Get-
1733 Subscriptions Response. There is no mechanism to allow for the next ‘M’ Subscription
1734 Objects after the first ‘N’ Subscription Objects. If the client does not supply this attribute, the
1735 Printer responds with all applicable Subscription Objects.

1736 **11.2.5.1.3 “requested-attributes” (1setOf type2 keyword)**

1737 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute.
1738 This attribute specifies the attributes of the specified Subscription Objects that the Printer
1739 MUST return in the response. Each value of this attribute is either an attribute name (defined
1740 in sections 5.3 and 5.4) or an attribute group name (defined in section 11.2.4.1). If the client
1741 omits this attribute, the Printer MUST respond as if the client had supplied this attribute with
1742 the one value: ‘notify-subscription-id’.

1743 **11.2.5.1.4 “my-subscriptions” (boolean)**

1744 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. If
1745 the value is ‘false’, the Printer MUST consider the Subscription Objects from all users as
1746 candidates. If the value is ‘true’, the Printer MUST return the Subscription Objects created by
1747 the requesting user of this request. If the client does not supply this attribute, the Printer
1748 MUST respond as if the client had supplied the attribute with a value of ‘false’. The means
1749 for authenticating the requesting user and matching the Subscription Objects is similar to that
1750 for Jobs which is described in [RFC2911] section 8.
1751

1752 **11.2.5.2 Get-Subscriptions Response**

1753 The Printer returns the following sets of attributes as part of the Get-Subscriptions Response:

1754 Group 1: Operation Attributes

1755 Status Message:

1756 Same as [RFC2911].

1757

1758 Natural Language and Character Set:

1759 The “attributes-charset” and “attributes-natural-language” attributes as described in
1760 [RFC2911] section 3.1.4.2.

1761

1762 Group 2: Unsupported Attributes

1763 Same as for Get-Subscription-Attributes.

1764

1765 Groups 3 to N: Subscription Attributes

1766 The Printer responds with one Subscription Attributes Group for each requested Subscription
1767 Object (see the “notify-job-id” attribute in the Operation Attributes Group of this operation).

1768

1769 The Printer returns Subscription Objects in any order.

1770

1771 If the “limit” attribute is present in the Operation Attributes group of the request, the number
1772 of Subscription Attributes Groups in the response MUST NOT exceed the value of the “limit”
1773 attribute.

1774

1775 If there are no Subscription Objects associated with the specified Job or Printer, the Printer
1776 MUST return zero Subscription Attributes Groups and it MUST NOT treat this case as an
1777 error, i.e., the status-code MUST be ‘successful-ok’ unless something else causes the status
1778 code to have some other value.

1779

1780 See the Group 3 response (Subscription Attributes Group) of the Get-Subscription-Attributes
1781 operation (section 11.2.4.2) for the attributes that a Printer returns in this group.

1782

1783 **11.2.6 Renew-Subscription operation**

1784 This operation allows a client to request the Printer to extend the lease on a Per-Printer Subscription
1785 Object.

1786 The Printer **MUST** support this operation.

1787 The Printer **MUST** accept this request for a Per-Printer Subscription Object in any of the target
1788 Printer's states, i.e., 'idle', 'processing', or 'stopped', but **MUST NOT** change the Printer's "printer-
1789 state" attribute.

1790 The Printer **MUST** reject this request for a Per-Job Subscription Object because it has no lease (see
1791 section 5.4.3). The status code returned **MUST** be 'client-error-not-possible'.

1792 *Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation **MUST**
1793 (1) be the owner of the Per-Printer Subscription Object, (2) have Operator or Administrator access
1794 rights for the Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the
1795 Printer's administrator-configured security policy to renew Per-Printer Subscription Objects for the
1796 target Printer. Otherwise, the Printer **MUST** reject the operation and return: the 'client-error-
1797 forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' status code as appropriate.

1798 **11.2.6.1 Renew-Subscription Request**

1799 The following groups of attributes are part of the Renew-Subscription Request:

1800 Group 1: Operation Attributes

1801 Natural Language and Character Set:

1802 The "attributes-charset" and "attributes-natural-language" attributes as described in
1803 [RFC2911] section 3.1.4.1.

1804

1805 Target:

1806 The "printer-uri" attribute which defines the target for this operation as described in
1807 [RFC2911] section 3.1.5.

1808

1809 Requesting User Name:

1810 The "requesting-user-name" (name(MAX)) attribute **SHOULD** be supplied by the client as
1811 described in [RFC2911] section 8.3.

1812

1813 **11.2.6.1.1 "notify-subscription-id" (integer (1:MAX))**

1814 The client **MUST** supply this attribute. The Printer **MUST** support this attribute. This
1815 attribute specifies the Per-Printer Subscription Object whose lease the Printer **MUST** renew.

1816 If the client omits this attribute, the Printer MUST reject this request with the ‘client-error-
1817 bad-request’ status code.

1818

1819 Group 2: Subscription Template Attributes

1820 11.2.6.1.2 “notify-lease-duration” (integer(0:MAX))

1821 The client MAY supply this attribute. It indicates the number of seconds to renew the lease
1822 for the specified Subscription Object. A value of 0 requests an infinite lease (which MAY
1823 require Operator access rights). If the client omits this attribute, the Printer MUST use the
1824 value of the Printer’s “notify-lease-duration-default” attribute. See section 5.3.8 for more
1825 details.

1826

1827 11.2.6.2 Renew-Subscription Response

1828 The Printer returns the following sets of attributes as part of the Renew-Subscription Response:

1829 Group 1: Operation Attributes

1830 Status Message:

1831 Same as [RFC2911].

1832

1833 The following are some of the status codes returned (see [RFC2911]):

1834

1835 **successful-ok:** The operation successfully renewed the lease on the Subscription Object
1836 for the requested duration.

1837 **successful-ok-ignored-or-substituted-attributes:** The operation successfully renewed
1838 the lease on the Subscription Object for some duration other than the amount
1839 requested.

1840 **client-error-not-possible:** The operation failed because the “notify-subscription-id”
1841 Operation attribute identified a Per-Job Subscription Object.

1842 **client-error-not-found:** The operation failed because the “notify-subscription-id”
1843 Operation attribute identified a non-existent Subscription Object.

1844

1845 Natural Language and Character Set:

1846 The “attributes-charset” and “attributes-natural-language” attributes as described in
1847 [RFC2911] section 3.1.4.2. The “attributes-natural-language” MAY be the natural language
1848 of the Subscription Object, rather than the one requested.

1849

1850 Group 2: Unsupported Attributes

1851 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes.

1852

1853 Group 3: Subscription Attributes

1854 The Printer MUST return the following Subscription Attribute:

1855 **11.2.6.2.1 “notify-lease-duration” (integer(0:MAX))**

1856 The value of this attribute MUST be the number of seconds that the Printer has granted for the
1857 lease of the Subscription Object (see section 5.3.8 for details, such as the value of this
1858 attribute when the Printer doesn't support the requested value).

1859 **11.2.7 Cancel-Subscription operation**

1860 This operation allows a client to delete a Subscription Object and stop the Printer from delivering more
1861 Event Notifications. Once performed, there is no way to reference the Subscription Object.

1862 A Printer MUST supported this operation.

1863 The Printer MUST accept this request in any of the target Printer's states, i.e., 'idle', 'processing', or
1864 'stopped', but MUST NOT change the Printer's "printer-state" attribute.

1865 If the specified Subscription Object is a Per-Job Subscription Object, the Printer MUST accept this
1866 request in any of the target Job's states, but MUST NOT change the Job's "job-state" attribute or affect
1867 the Job.

1868 Note: There is no way to change any attributes on a Subscription Object, except the "notify-lease-
1869 duration" attribute (using the Renew-Subscription operation). In order to change other attributes, a
1870 client performs a Subscription Creation Operation and Cancel-Subscription operation on the old
1871 Subscription Object. If the client wants to avoid missing Event Notifications, it performs the
1872 Subscription Creation Operation first. If this order would create too many Subscription Objects on the
1873 Printer, the client reverses the order.

1874 *Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation MUST
1875 (1) be the owner of the Subscription Object, (2) have Operator or Administrator access rights for the
1876 Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer's
1877 administrator-configured security policy to cancel the target Subscription Object. Otherwise, the
1878 Printer MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-
1879 authenticated', or 'client-error-not-authorized' status code as appropriate.

1880 **11.2.7.1 Cancel-Subscription Request**

1881 The following groups of attributes are part of the Cancel-Subscription Request:

1882 Group 1: Operation Attributes

1883 Natural Language and Character Set:
1884 The “attributes-charset” and “attributes-natural-language” attributes as described in
1885 [RFC2911] section 3.1.4.1.
1886

1887 Target:
1888 The “printer-uri” attribute which defines the target for this operation as described in
1889 [RFC2911] section 3.1.5.
1890

1891 Requesting User Name:
1892 The “requesting-user-name” attribute SHOULD be supplied by the client as described in
1893 [RFC2911] section 8.3.

1894 11.2.7.1.1 “notify-subscription-id” (integer (1:MAX))

1895 The client MUST supply this attribute. The Printer MUST support this attribute. This
1896 attribute specifies the Subscription Object that the Printer MUST cancel. If the client omits
1897 this attribute, the Printer MUST reject this request with the ‘client-error-bad-request’ status
1898 code.
1899

1900 11.2.7.2 Cancel-Subscription Response

1901 The Printer returns the following sets of attributes as part of the Cancel-Subscription Response:

1902 Group 1: Operation Attributes

1903 Status Message:
1904 Same as [RFC2911].
1905

1906 The following are some of the status codes returned (see [RFC2911]):
1907

1908 **successful-ok:** The operation successfully canceled (deleted) the Subscription Object.
1909 **client-error-not-found:** The operation failed because the “notify-subscription-id”
1910 Operation attribute identified a non-existent Subscription Object.
1911

1912 Natural Language and Character Set:
1913 The “attributes-charset” and “attributes-natural-language” attributes as described in
1914 [RFC2911] section 3.1.4.2. The “attributes-natural-language” MAY be the natural language
1915 of the Subscription Object, rather than the one requested.
1916

1917 Group 2: Unsupported Attributes

1918 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes.
1919

1920 **12 Status Codes**

1921 The following status codes are defined as extensions for Notification and are returned as the value of
1922 the “status-code” parameter in the Operation Attributes Group of a response (see [RFC2911] section
1923 3.1.6.1). Operations in this document can also return the status codes defined in section 13 of
1924 [RFC2911]. The ‘successful-ok’ status code is an example of such a status code.

1925 **12.1 successful-ok-ignored-subscriptions (0x0003)**

1926 The Subscription Creation Operation was unable to create all requested Subscription Objects.

1927 For a Create-Job-Subscriptions or Create-Printer-Subscriptions operation, this status code means that
1928 the Printer created one or more Subscription Objects, but not all requested Subscription Objects.

1929 For a Job Creation operation, this status code means that the Printer created the Job along with zero or
1930 more Subscription Objects. The Printer returns this status code even if other job attributes are
1931 unsupported or in conflict. That is, if an IPP Printer finds a warning that would allow it to return
1932 ‘successful-ok-ignored-subscriptions’ and either ‘successful-ok-ignored-or-substituted-attributes’
1933 and/or ‘successful-ok-conflicting-attributes’, it MUST return ‘successful-ok-ignored-subscriptions’.

1934 **12.2 client-error-ignored-all-subscriptions (0x0414)**

1935 This status code is the same as ‘successful-ok-ignored-subscriptions’ except that only the Create-Job-
1936 Subscriptions and Create-Printer-Subscriptions operation return it. They return this status code only
1937 when the Printer creates zero Subscription Objects.

1938 **13 Status Codes in Subscription Attributes Groups**

1939 This section contains values of the “notify-status-code” (type2 enum) attribute that the Printer returns
1940 in a Subscription Attributes Group in a response when the corresponding Subscription Object:

- 1941 1. is not created or
- 1942 2. is created and some of the client-supplied attributes are not supported.

1943 The following sections are ordered in decreasing order of importance of the status-codes.

1944 **13.1 client-error-uri-scheme-not-supported (0x040C)**

1945 This status code is defined in [RFC2911]. This document extends its meaning and allows it to be in a
1946 Subscription Attributes Group of a response.

1947 The scheme of the client-supplied URI in a “notify-recipient-uri” Subscription Template Attribute in a
1948 Subscription Creation Operation is not supported. See section 5.3.1.

1949 **13.2 client-error-attributes-or-values-not-supported (0x040B)**

1950 This status code is defined in [RFC2911]. This document extends its meaning and allows it to be in a
1951 Subscription Attributes Group of a response.

1952 The method of the client-supplied keyword in a “notify-pull-method” Subscription Template Attribute
1953 in a Subscription Creation Operation is not supported. See section 5.3.2.

1954 **13.3 client-error-too-many-subscriptions (0x0415)**

1955 The number of Subscription Objects supported by the Printer would be exceeded if this Subscription
1956 Object were created (see section 5.2).

1957 **13.4 successful-ok-too-many-events (0x0005)**

1958 The client supplied more Events in the “notify-events” operation attribute of a Subscription Creation
1959 Operation than the Printer supports, as indicated in its “notify-max-events-supported” Printer attribute
1960 (see section 5.3.3).

1961 **13.5 successful-ok-ignored-or-substituted-attributes (0x0001)**

1962 This status code is defined in [RFC2911]. This document extends its meaning to include unsupported
1963 Subscription Template Attributes and it can appear in a Subscription Attributes Group.

1964 **14 Encodings of Additional Attribute Tags**

1965 This section assigns values to two attributes tags as extensions to the encoding defined in [RFC2910]).

1966 The “subscription-attributes-tag” delimits Subscription Template Attributes Groups in requests and
1967 Subscription Attributes Groups in responses.

1968 The “event-notification-attributes-tag” delimits Event Notifications in Delivery Methods that use an
1969 IPP-like encoding.

1970 The following table specifies the values for the delimiter tags:

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

1971 **15 Conformance Requirements**

1972 It is OPTIONAL for IPP clients and Printers to implement this Event Notification specification.

1973 **15.1 Conformance requirements for clients**

1974 If this Event Notification specification is implemented by a client, the client **MUST** support the
1975 'ippget' Pull Delivery Method and meet the conformance requirements as defined in [ipp-get-method]
1976 for clients. A client **MAY** support additional Delivery Methods.

1977 **15.2 Conformance requirements for Printers**

1978 If this Event Notification specification is implemented by a Printer, the Printer **MUST**:

- 1979 - meet the Conformance Requirements detailed in section 5 of [RFC2911].
- 1980 - support the Subscription Template Attributes Group in requests and the Subscription
1981 Attributes Group in responses.
- 1982 - support all of the following attributes:
 - 1983 a. **REQUIRED** Subscription Object attributes in section 5.
 - 1984 b. **REQUIRED** Printer Description object attributes in section 6.
 - 1985 c. **REQUIRED** attributes in Event Notification content in section 8.
- 1986 - support the 'ippget' Pull Delivery Method and meet the conformance requirements as defined
1987 in [ipp-get-method] for Printers. The Printer **MAY** support additional Push and Pull Delivery
1988 Methods.
- 1989 - deliver Event Notifications that conform to the requirements of section 9 and the requirements
1990 of the Delivery Method Document for each supported Delivery Method (the conformance
1991 requirements for Delivery Method Documents is specified in section 10).
- 1992 - for all of the Job Creation Operations that the Printer supports, **MUST** support the
1993 **REQUIRED** extensions for notification defined in section 11.1.3.
- 1994 - meet the conformance requirements for operations as described in Table 16 and meet the
1995 requirements for Printers as specified in the indicated sub-sections of section 11:

1996

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

1997

1998

16 Appendix A - Model for Notification with Cascading Printers (Informative)

1999

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

2000

2001

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 delivers Event Notifications that are shown as Event Notifications (A) of Figure 2.

2002

2003

2004

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

2005

2006

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

2007

2008

2009

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 delivers 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 client performs a Subscription Creation Operation, Printer 1 need not forward the Subscription Creation Operation to Printer 2 if it would create a duplicate Subscription Object on Printer 2.

2010

2011

2012

2013

2014

2015

2016

Note: when Printer 1 is forwarding Subscription Creation Operations to Printer 2, it may request Printer 2 to create additional Subscription Objects (called “piggy-backing”). Piggy-backing is useful when:

2017

2018

2019

- Device A is configured to accept (IPP or non-IPP) requests from other servers.

2020

- Server S wants to receive Job Events that the client didn't request and Server S wants these Events for jobs it submits and not for other jobs.

2021

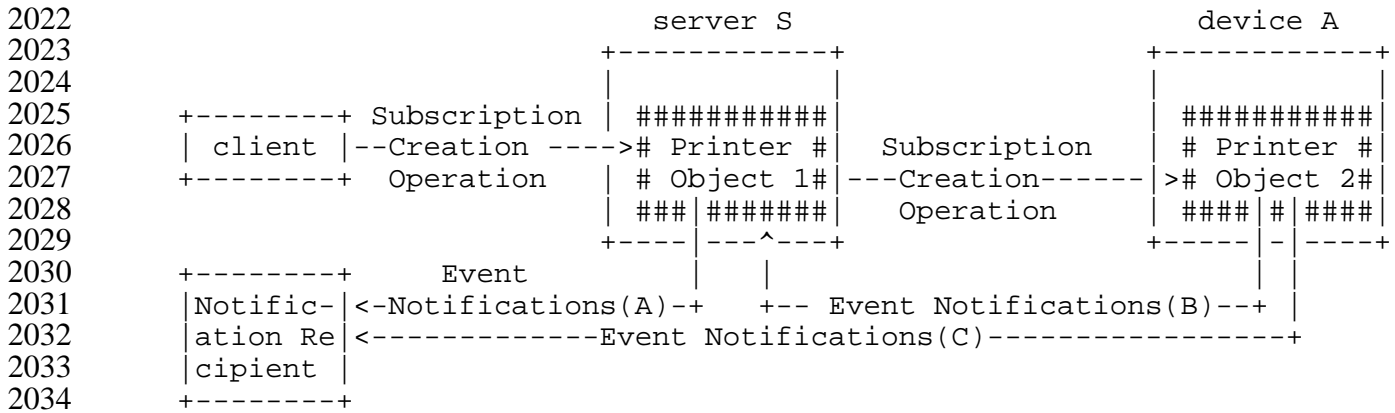


Figure 2 – Model for Notification with Cascading Printers

2036 **17 Appendix B - Distributed Model for Notification (Informative)**

2037 A Printer implementation could use some other remote notification server to provide some or most of
2038 the service. For example, the remote notification server could deliver Event Notifications using
2039 Delivery Methods that are not directly supported by the output device or Printer object. Or, the remote
2040 notification server could store Subscription Objects (passed to it from the output device in response to
2041 Subscription Creation requests), accept Events, format the Event Notification in the natural language of
2042 the Notification Recipient, and deliver the Event Notifications to the Notification Recipient(s).

2043 Figure 3 shows this partitioning. The interface between the output device (or Printer object) and the
2044 remote notification server is outside the scope of this document and is intended to be transparent to the
2045 client and this document.

2046

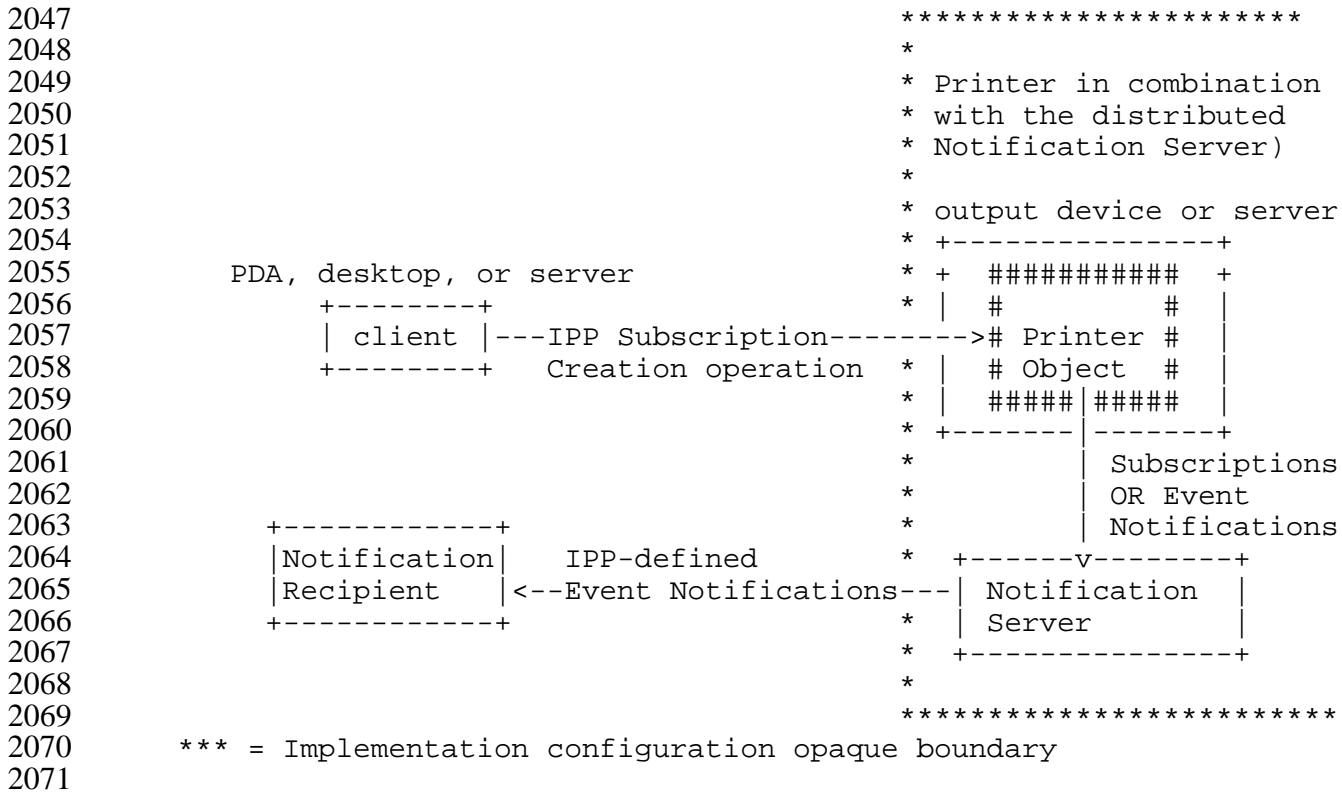


Figure 3 – Opaque Use of a Notification Server Transparent to the Client

18 Appendix C - Extended Notification Recipient (Informative)

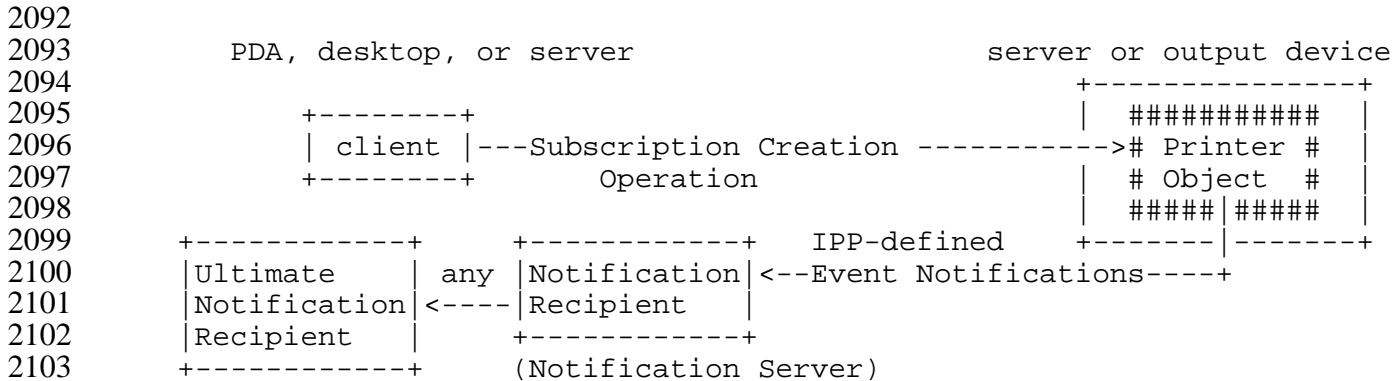
The model allows for an extended Notification Recipient that is itself a notification server that forwards each Event Notification to another recipient (called the Ultimate Notification Recipient in this section). The Delivery Method to the Ultimate Recipient is probably different from the Delivery Method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

2090 Examples of this extended Notification Recipient are paging, immediate messaging services, general
 2091 notification services, and NOS vendors' infrastructure. Figure 4 shows this approach.



2104 **Figure 4 – Use of an Extended Notification Recipient transparent to the Printer**

2105 **19 Appendix D - Details about Conformance Terminology (Normative)**

2106 The following paragraphs provide more details about conformance terminology.

2107 **REQUIRED** - an adjective used to indicate that a conforming IPP Printer implementation **MUST**
 2108 support the indicated operation, object, attribute, attribute value, status code, or out-of-band value
 2109 in requests and responses. See [RFC2911] "Appendix A - Terminology for a definition of
 2110 "support". *Since support of this entire Notification specification is OPTIONAL for*
 2111 *conformance to IPP/1.1, the use of the term REQUIRED in this document means*
 2112 *"REQUIRED if this OPTIONAL Notification specification is implemented".*

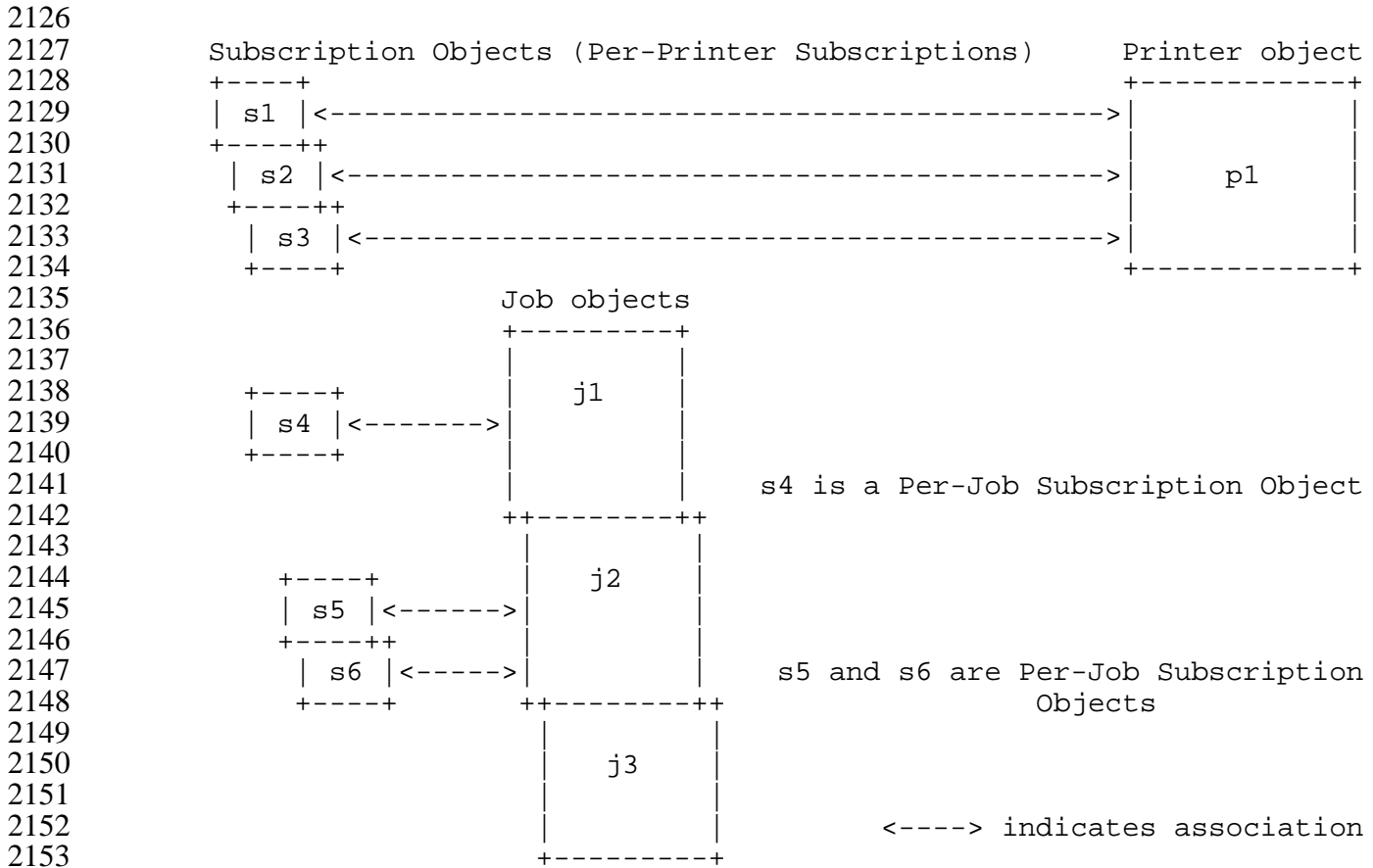
2113 **RECOMMENDED** - an adjective used to indicate that a conforming IPP Printer implementation is
 2114 recommended to support the indicated operation, object, attribute, attribute value, status code, or
 2115 out-of-band value in requests and responses. *Since support of this entire Notification*
 2116 *specification is OPTIONAL for conformance to IPP/1.1, the use of the term RECOMMENDED*
 2117 *in this document means "RECOMMENDED if this OPTIONAL Notification specification is*
 2118 *implemented".*

2119 **OPTIONAL** - an adjective used to indicate that a conforming IPP Printer implementation **MAY**, but is
 2120 **NOT REQUIRED** to, support the indicated operation, object, attribute, attribute value, status code,
 2121 or out-of-band value in requests and responses.

2122 **20 Appendix E - Object Model for Notification (Normative)**

2123 This section describes the Notification object model that adds a Subscription Object which together
 2124 with the Job and Printer object provide the complete Notification semantics.

2125 The object relationships can be seen pictorially as:



2154 **Figure 5 – Object Model for Notification**

2155 s1, s2, and s3 are Per-Printer Subscription Objects and can identify Printer and/or Job Events.
 2156 s4, s5, and s6 are Per-Job Subscription Objects and can identify Printer and/or Job Events.

2157 **20.1 Object relationships**

2158 This sub-section defines the object relationships between the Printer, Job, and Subscription Objects by
 2159 example. Whether Per-Printer Subscription Objects are actually contained in a Printer object or are
 2160 just bi-directionally associated with them in some way is IMPLEMENTATION DEPENDENT and is
 2161 transparent to the client. Similarly, whether Per-Job Subscription Objects are actually contained in a
 2162 Job object or are just bi-directionally associated with them in some way is IMPLEMENTATION
 2163 DEPENDENT and is transparent to the client. The object relationships are defined as follows:

2164 **20.2 Printer Object and Per-Printer Subscription Objects**

- 2165 1. The Printer object contains (is associated with) zero or more Per-Printer Subscription Objects
 2166 (p1 contains s1-s3 Per-Printer Subscription Objects).

- 2167 2. Each Per-Printer Subscription Object (s1, s2, and s3) is contained in (or is associated with)
2168 exactly one Printer object (p1).

2169 **20.3 Job Object and Per-Job Subscription Objects**

- 2170 1. A Job object (j1, j2, j3) is associated with zero or more Per-Job Subscription Objects (s4-s6).
2171 Job j1 is associated with Per-Job Subscription Object s4, Job j2 is associated with Per-Job
2172 Subscription Objects s5 and s6, and Job j3 is not associated with any Per-Job Subscription
2173 Object.

- 2174 2. Each Per-Job Subscription Object is associated with exactly one Job object.

2175 **21 Appendix F - Per-Job versus Per-Printer Subscription Objects (Normative)**

2176 Per-Job and Per-Printer Subscription Objects are quite similar. Either type of Subscription Object can
2177 subscribe to Job Events, Printer Events, or both. Both types of Subscription Objects can be queried
2178 using the Get-Subscriptions and Get-Subscription-Attributes operations and canceled using the Cancel-
2179 Subscription operation. Both types of Subscription Objects create Subscription Objects which have the
2180 same Subscription Object attributes defined. However, there are some semantic differences between
2181 Per-Job Subscription Objects and Per-Printer Subscription Objects. A Per-Job Subscription Object is
2182 established by the client when submitting a job and after creating the job using the Create-Job-
2183 Subscriptions operation by specifying the “job-id” of the Job with the “notify-job-id” attribute. A Per-
2184 Printer Subscription Object is established between a client and a Printer using the Create-Printer-
2185 Subscriptions operation. Some specific differences are:

- 2186 1. A client usually creates one or more Per-Job Subscription Objects as part of the Job Creation
2187 operations (Create-Job, Print-Job, and Print-URI), rather than using the OPTIONAL Create-Job-
2188 Subscriptions operation, especially since Printer implementations NEED NOT support the
2189 Create-Job-Subscriptions operation, since it is OPTIONAL.
- 2190 2. For Per-Job Subscription Objects, the Subscription Object is only valid while the job is “not-
2191 complete” (see sections 5.4.3) while for the Per-Printer Subscription Objects, the Subscription
2192 Object is valid until the time (in seconds) that the Printer returned in the “notify-lease-
2193 expiration-time” operation attribute.
- 2194 3. Job Events in a Per-Job Subscription Object apply only to “one job” (the Job created by the Job
2195 Creation operation or references by the Create-Job-Subscriptions operation) while Job Events in
2196 a Per-Printer Subscription Object apply to ALL jobs contained in the IPP Printer.

2197 **22 Normative References**

- 2198 [ipp-get-method]
2199 Herriot, R., and T. Hastings, "Internet Printing Protocol (IPP): The 'ippget' Delivery Method for
2200 Event Notifications", <draft-ietf-ipp-notify-get-08.txt>, September 10, 2002.

- 2201 [RFC2119]
2202 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March 1997
- 2203 [RFC2396]
2204 Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic
2205 Syntax", RFC 2396, August 1998.
- 2206 [RFC2717]
2207 R. Petke and I. King, "Registration Procedures for URL Scheme Names", RFC 2717, November
2208 1999.
- 2209 [RFC2910]
2210 Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.1: Encoding and
2211 Transport", RFC 2910, September 2000.
- 2212 [RFC2911]
2213 deBry, R., Hastings, T., Herriot, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.1:
2214 Model and Semantics", RFC 2911, September 2000.
- 2215 [RFC3381]
2216 Hastings, T., Lewis, H., and R. Bergman, "IPP: Job Progress Attributes", RFC 3381, September
2217 2002.

2218 **23 Informative References**

- 2219 [IANA-CON]
2220 Narte, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs",
2221 BCP 26, RFC 2434, October 1998.
- 2222 [ipp-not-req]
2223 deBry, R., Lewis, H., and T. Hastings, "Internet Printing Protocol/1.1: Requirements for IPP
2224 Notifications", <draft-ietf-ipp-not-06.txt>, work in progress, July 17, 2001.
- 2225 [RFC2565]
2226 Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.0: Encoding and
2227 Transport", RFC 2565, April 1999.
- 2228 [RFC2566]
2229 deBry, R., , Hastings, T., Herriot, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.0:
2230 Model and Semantics", RFC 2566, April 1999.
- 2231 [RFC2567]
2232 Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.

- 2233 [RFC2568]
 2234 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
 2235 RFC 2568, April 1999.
- 2236 [RFC2569]
 2237 Herriot, R., Hastings, T., Jacobs, N., and J. Martin, "Mapping between LPD and IPP Protocols",
 2238 RFC 2569, April 1999.
- 2239 [RFC2616]
 2240 Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee,
 2241 "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
- 2242 [RFC3196]
 2243 Hastings, T., Manros, C., Zehler, P., Kugler, C., and H. Holst, "Internet Printing Protocol/1.1:
 2244 Implementer's Guide", RFC3196, November 2001.

2245 24 IANA Considerations

2246 This section contains the registration information for IANA to add to the various IPP Registries
 2247 according to the procedures defined in RFC 2911 [RFC2911] section 6 to cover the definitions in this
 2248 document. In addition, this section defines how Events and Delivery Methods will be registered when
 2249 they are defined in other documents. The resulting registrations will be published in the
 2250 <http://www.iana.org/assignments/ipp-registrations> registry.

2251 *Note to RFC Editors: Replace RFC NNNN below with the RFC number for this document, so that it*
 2252 *accurately reflects the content of the information for the IANA Registry.*

2253 24.1 Attribute Registrations

2254 The following table lists all the attributes defined in this document. These are to be registered
 2255 according to the procedures in RFC 2911 [RFC2911] section 6.2.

2256	Subscription Template attributes:	Ref.	Section:
2257	notify-attributes (1setOf type2 keyword)	RFC NNNN	5.3.4
2258	notify-attributes-supported (1setOf type2 keyword)		
2259		RFC NNNN	5.3.4.1
2260	notify-charset (charset)	RFC NNNN	5.3.6
2261	notify-events (1setOf type2 keyword)	RFC NNNN	5.3.3
2262	notify-events-default (1setOf type2 keyword)	RFC NNNN	5.3.3.1
2263	notify-events-supported (1setOf type2 keyword)	RFC NNNN	5.3.3.2
2264	notify-lease-duration (integer(0:67108863))	RFC NNNN	5.3.8
2265	notify-lease-duration-default (integer(0:67108863))		
2266		RFC NNNN	5.3.8.1
2267	notify-lease-duration-supported (1setOf (integer(0: 67108863)		
2268	rangeOfInteger(0:67108863)))	RFC NNNN	5.3.8.2
2269	notify-max-events-supported (integer(2:MAX))	RFC NNNN	5.3.3.3

2270	notify-natural-language (naturalLanguage)	RFC NNNN	5.3.7
2271	notify-pull-method (type2 keyword)	RFC NNNN	5.3.2
2272	notify-pull-method-supported (1setOf type2 keyword)		
2273		RFC NNNN	5.3.2.1
2274	notify-recipient-uri (uri)	RFC NNNN	5.3.1
2275	notify-schemes-supported (1setOf uriScheme)	RFC NNNN	5.3.1.1
2276	notify-time-interval (integer(0:MAX))	RFC NNNN	5.3.9
2277	notify-user-data (octetString(63))	RFC NNNN	5.3.5
2278			
2279	Subscription Description Attributes:		
2280	notify-job-id (integer(1:MAX))	RFC NNNN	5.4.6
2281	notify-lease-expiration-time (integer(0:MAX))	RFC NNNN	5.4.3
2282	notify-printer-up-time (integer(1:MAX))	RFC NNNN	5.4.4
2283	notify-printer-uri (uri)	RFC NNNN	5.4.5
2284	notify-sequence-number (integer (0:MAX))	RFC NNNN	5.4.2
2285	notify-subscriber-user-name (name(MAX))	RFC NNNN	5.4.7
2286	notify-subscription-id (integer (1:MAX))	RFC NNNN	5.4.1
2287			
2288	Printer Description Attributes:		
2289	printer-state-change-date-time (dateTime)	RFC NNNN	6.2
2290	printer-state-change-time (integer(1:MAX))	RFC NNNN	6.1
2291			
2292	Attributes Only in Event Notifications		
2293	notify-subscribed-event (type2 keyword)	RFC NNNN	8.1
2294	notify-text (text(MAX))	RFC NNNN	8.2
2295			
2296			

2297 24.2 Additional Enum Attribute Value Registrations

2298 The following table lists all the new enum attribute values defined in this document. These are to be
 2299 registered according to the procedures in RFC 2911 [RFC2911] section 6.1.

2300	Attribute			
2301	Value	Name	Reference	Section
2302	-----	-----	-----	-----
2303	operations-supported (type2 enum)		RFC2911	4.4.15
2304	0x0016	Create-Printer-Subscriptions	RFC NNNN	7.1
2305	0x0017	Create-Job-Subscriptions	RFC NNNN	7.1
2306	0x0018	Get-Subscription-Attributes	RFC NNNN	7.1
2307	0x0019	Get-Subscriptions	RFC NNNN	7.1
2308	0x001A	Renew-Subscription	RFC NNNN	7.1
2309	0x001B	Cancel-Subscription	RFC NNNN	7.1
2310				

2311 24.3 Operation Registrations

2312 The following table lists all of the operations defined in this document. These are to be registered
 2313 according to the procedures in RFC 2911 [RFC2911] section 6.4.

	Operations:	Ref.	Section:
2314	Cancel-Subscription Operation	RFC NNNN	11.2.7
2315	Create-Job-Subscriptions Operation	RFC NNNN	11.1.1
2316	Create-Printer-Subscriptions Operation	RFC NNNN	11.1.2
2317	Get-Printer-Attributes - Extensions	RFC NNNN	11.2.3
2318	Get-Subscription-Attributes Operation	RFC NNNN	11.2.4
2319	Get-Subscriptions Operation	RFC NNNN	11.2.5
2320	Job Creation Operations - Extensions	RFC NNNN	11.1.3
2321	Renew-Subscription Operation	RFC NNNN	11.2.6
2322	Validate-Job Operation - Extensions	RFC NNNN	11.2.2
2323			
2324			

2325 24.4 Status code Registrations

2326 The following table lists all the status codes defined in this document. These are to be registered
 2327 according to the procedures in RFC 2911 [RFC2911] section 6.6.

	Value	Name	Ref.	Section:
2328	-----	-----	-----	-----
2329				
2330	0x0000:0x00FF	- "successful"		
2331	0x0003	successful-ok-ignored-subscriptions	RFC NNNN	12.1
2332				
2333	0x0400:0x04FF	- "client-error"		
2334	0x0414	client-error-ignored-all-subscriptions	RFC NNNN	12.2
2335				

2336 24.5 Attribute Group tag Registrations

2337 The following table lists all the attribute group tags defined in this document. These are to be
 2338 registered according to the procedures in RFC 2911 [RFC2911] section 6.5.

	Value	Name	Ref.	Section
2339	-----	-----	-----	-----
2340				
2341	0x06	subscription-attributes-tag	RFC NNNN	14
2342	0x07	event-notification-attributes-tag	RFC NNNN	14
2343				

2344 24.6 Registration of Events

2345 When other document define additional type2 keywords to be used with the "notify-events"
 2346 Subscription Template attribute (see section 5.3.3), these event keywords will be registered according
 2347 to the procedures of [RFC2911] section 7.1 as additional attribute values for use with the "notify-
 2348 events" Subscription Template attribute, i.e., the "notify-events", "notify-events-default", and "notify-
 2349 events-supported" attributes.

2350 Therefore, the IPP Registry entry for an Event will be of the form:

2351 Attribute		Ref.	Section
2352 Value	-----	-----	-----
2353			
2354	notify-events (1setOf type2 keyword)		
2355	notify-events-default (1setOf type2 keyword)		
2356	notify-events-supported (1setOf type2 keyword)		
2357	<event keyword name>	RFC xxxxx	m.n
2358			

2359 **24.7 Registration of Event Notification Delivery Methods**

2360 This section describes the requirements and procedures for registration and publication of Event
2361 Notification Delivery Methods and for the submission of such proposals.

2362 **24.7.1 Requirements for Registration of Event Notification Delivery Methods**

2363 Registered IPP Event Notification Delivery Methods are expected to follow a number of requirements
2364 described below.

2365 **24.7.1.1 Required Characteristics**

2366 A Delivery Method Document MUST either (1) contain all of the semantics of the Delivery Method or
2367 (2) contain the IPP Delivery Method registration requirements and a profile of some other protocol that
2368 in combination is the Delivery Method (e.g., mailto). The Delivery Method Document (and any
2369 documents it requires) MUST define either (1) a URL for a Push Delivery Method that the meets the
2370 requirements of [RFC2717]. or (2) a keyword for a Pull Delivery method.

2371
2372 IPP Event Notification Delivery Method Documents MUST meet the requirements of this document
2373 (see sections 9 and 10).

2374 In addition, a Delivery Method Document MUST contain the following information:

2375
2376 Type of registration: IPP Event Notification Delivery Method
2377 Name of this delivery method:
2378 Proposed URL scheme name of this Push Delivery Method or the keyword name of this Pull
2379 Delivery Method:
2380 Name of proposer:
2381 Address of proposer:
2382 Email address of proposer:
2383 Is this delivery method REQUIRED or OPTIONAL for conformance to the IPP Event Notification
2384 and Subscriptions document:
2385 Is this delivery method defining Machine Consumable and/or Human Consumable content:
2386

2387 24.7.1.2 Naming Requirements

2388 Exactly one (URL scheme or keyword) name **MUST** be assigned to each Delivery Method.

2389 Each assigned name **MUST** uniquely identify a single Delivery Method. All Push Delivery Method
2390 names **MUST** conform to the rules for URL scheme names, according to [RFC2396] and [RFC2717]
2391 for schemes in the IETF tree. All Pull Delivery Method names **MUST** conform to the rules for
2392 keywords according to [RFC2911].

2393 24.7.1.3 Functionality Requirements

2394 Delivery Methods **MUST** function as a protocol that is capable of delivering (push or pull) IPP Event
2395 Notifications to Notification Recipients.

2396 24.7.1.4 Usage and Implementation Requirements

2397 Use of a large number of Delivery Methods may hamper interoperability. However, the use of a large
2398 number of undocumented and/or unlabelled Delivery Methods hampers interoperability even more.

2399 A Delivery Method should therefore be registered **ONLY** if it adds significant functionality that is
2400 valuable to a large community, **OR** if it documents existing practice in a large community. Note that
2401 Delivery Methods registered for the second reason should be explicitly marked as being of limited or
2402 specialized use and should only be used with prior bilateral agreement.

2403 24.7.1.5 Publication Requirements

2404 Delivery Method Documents **MUST** be published in a standards track, informational, or experimental
2405 RFCs.

2406 24.7.2 Registration Procedure

2407 The IPP WG is developing a small number of Delivery Methods which are intended to be published as
2408 standards track RFCs. However, some parties may wish to register additional Delivery Methods in the
2409 future. This section describes the procedures for these additional Delivery Methods.

2410 24.7.2.1 Present the proposal to the Community

2411 First the Delivery Method Document **MUST** be an Internet-Draft with a target category of standards
2412 track, informational, or experimental. The same **MUST** be true for any documents that it references.

2413 Deliver the proposed Delivery Method Document proposal to the "ipp@pwg.org" mailing list. This
2414 mailing list has been established by [RFC2911] for reviewing proposed registrations and discussing

2415 other IPP matters. Proposed Delivery Method Documents are not formally registered and MUST NOT
2416 be used until approved.

2417 The intent of the public posting is to solicit comments and feedback on the definition and suitability of
2418 the Delivery Method and the name chosen for it over a four week period.

2419 **24.7.2.2 Delivery Method Reviewer**

2420 The Delivery Method Reviewer is the same person who has been appointed by the IETF Application
2421 Area Director(s) as the IPP Designated Expert according to [RFC2911] and [IANA-CON]. When the
2422 four week period is over and the IPP Designated Expert is convinced that consensus has been achieved,
2423 the IPP Designated Expert either approves the request for registration or rejects it. Rejection may
2424 occur because of significant objections raised on the list or objections raised externally.

2425 Decisions made by the Reviewer must be posted to the `ipp@pwg.org` mailing list within 14 days.
2426 Decisions made by the Reviewer may be appealed to the IESG.

2427 **24.7.2.3 IANA Registration**

2428 Provided that the Delivery Method registration proposal has either passed review or has been
2429 successfully appealed to the IESG, the IANA will register the Delivery Method and make it available to
2430 the community.

2431 **24.7.3 Delivery Method Document Registrations**

2432 Each Push Delivery Method Document defines a URI scheme which is registered as an additional value
2433 of the “notify-schemes-supported” Printer attribute. These uriScheme values will be registered
2434 according to the procedures of [RFC2911] section 7.1 for additional attribute values. Therefore, the
2435 IPP Registry entry for a Push Delivery Method will be of the form:

2436	Attribute		
2437	Value	Ref.	Section
2438	-----	-----	-----
2439	notify-schemes-supported (type2 keyword)	RFC xxxx	5.3.1.1
2440	<scheme name>	RFC xxxx	m.n

2441
2442 Each Pull Delivery Method Document defines a keyword method which is registered as an additional
2443 value of the “notify-pull-method” and “notify-pull-method-supported” Printer attributes. These
2444 keyword values will be registered according to the procedures of [RFC2911] section 7.1 for additional
2445 attribute values. Therefore, the IPP Registry entry for a Pull Delivery Method will be of the form:

2446	Attribute		
2447	Value	Ref.	Section
2448	-----	-----	-----
2449	notify-pull-method (type2 keyword)	[ipp-ntfy]	5.3.2
2450	notify-pull-method-supported (1setOf type2 keyword)		
2451		[ipp-ntfy]	5.3.2.1
2452	<method keyword name>	RFC xxxx	m.n
2453			

2454 **24.7.4 Registration Template**

2455 To: ipp@pwg.org
 2456 Subject: Registration of a new Delivery Method

2457

2458 Delivery Method name:

2459

2460 (All Push Delivery Method names must be suitable for use as the value of a URL scheme in the IETF
 2461 tree and all Pull Delivery Method names must be suitable IPP keywords according to [RFC2911])

2462

2463 Published specification(s):

2464

2465 (A specification for the Delivery Method must be openly available that accurately describes what is
 2466 being registered.)

2467

2468 Person & email address to contact for further information:

2469 **25 Internationalization Considerations**

2470 This IPP Notification specification continues support for the internationalization of [RFC2911] of
 2471 attributes containing text strings and names. Allowing a Subscribing Client to specify a different
 2472 natural language and charset for each Subscription Object increases the internationalization support.

2473 The Printer MUST be able to localize the content of Human Consumable Event Notifications and to
 2474 localize the value of “notify-text” attribute in Machine Consumable Event Notifications that it delivers
 2475 to Notification Recipients. For localization, the Printer MUST use the value of the “notify-charset”
 2476 attribute and the “notify-natural-language” attribute in the Subscription Object supplied by the
 2477 Subscribing Client.

2478 **26 Security Considerations**

2479 Clients submitting Notification requests to the IPP Printer have the same security issues as submitting
 2480 an IPP/1.1 print job request (see [RFC2911] section 3.2.1 and section 8). The same mechanisms used
 2481 by IPP/1.1 can therefore be used by the client Notification submission. Operations that require
 2482 authentication can use the HTTP authentication. Operations that require privacy can use the
 2483 HTTP/TLS privacy. As with IPP/1.1 Print Job Objects, if there is no security on Subscription Objects,
 2484 sequential assignment of subscription-ids exposes the system to a passive traffic monitoring threat.

2485 **26.1 Client access rights**

2486 The Subscription Object access control model is the same as the access control model for Job objects.
2487 The client **MUST** have the following access rights for the indicated Subscription operations:

- 2488 1. Create-Job-Subscriptions (see section 11.1.1): A Per-Job Subscription object is associated with
2489 a Job. To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section
2490 8.3) performing this operation **MUST** (1) be the job owner, (2) have Operator or Administrator
2491 access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized
2492 by the Printer's administrator-configured security policy to create Per-Job Subscription Objects
2493 for the target job.
- 2494 2. Create-Printer-Subscriptions (see section 11.1.2): A Per-Printer Subscription object is
2495 associated with the Printer. To create Per-Printer Subscription Objects, the authenticated user
2496 (see [RFC2911] section 8.3) performing this operation **MUST** (1) have Operator or
2497 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5) or (2) be
2498 otherwise authorized by the Printer's administrator-configured security policy to create Per-
2499 Printer Subscription Objects for this Printer.
- 2500 3. Get-Subscription-Attributes (see section 11.2.4): The access control model for this operation is
2501 the same as that of the Get-Job-Attributes operation (see [RFC2911] section 3.3.4). The
2502 primary difference is that a Get-Subscription-Attributes operation is directed at a Subscription
2503 Object rather than at a Job object, and a returned attribute group contains Subscription Object
2504 attributes rather than Job object attributes. To query the specified Subscription Object, the
2505 authenticated user (see [RFC2911] section 8.3) performing this operation **MUST** (1) be the
2506 Subscription Object owner, (2) have Operator or Administrator access rights for this Printer
2507 (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer's
2508 administrator-configured security policy to query the Subscription Object for the target job.
2509 Furthermore, the Printer's security policy **MAY** limit which attributes are returned, in a manner
2510 similar to the Get-Job-Attributes operation (see [RFC2911] end of section 3.3.4.2).
- 2511 4. Get-Subscriptions (see section 11.2.5): The access control model for this operation is the same
2512 as that of the Get-Jobs operation (see [RFC2911] section 3.2.6). The primary difference is that
2513 the operation is directed at Subscription Objects rather than at Job objects, and the returned
2514 attribute groups contain Subscription Object attributes rather than Job object attributes. To
2515 query Per-Job Subscription Objects of the specified job (client supplied the "notify-job-id"
2516 operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)
2517 performing this operation **MUST** (1) be the Subscription Object owner, (2) have Operator or
2518 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be
2519 otherwise authorized by the Printer's administrator-configured security policy to query the
2520 Subscription Object for the target job. To query Per-Printer Subscription Objects of the Printer
2521 (client omits the "notify-job-id" operation attribute - see section 11.2.5.1.1), the authenticated
2522 user (see [RFC2911] section 8.3) performing this operation **MUST** (1) have Operator or
2523 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (2) be
2524 otherwise authorized by the Printer's administrator-configured security policy to query Per-
2525 Printer Subscription Objects for the target Printer. Furthermore, the Printer's security policy

2526 MAY limit which attributes are returned, in a manner similar to the Get-Job-Attributes
2527 operation (see [RFC2911] end of section 3.2.6.2).

2528 5. Renew-Subscriptions (see section 11.2.6): The authenticated user (see [RFC2911] section 8.3)
2529 performing this operation MUST (1) be the owner of the Per-Printer Subscription Object, (2)
2530 have Operator or Administrator access rights for the Printer (see [RFC2911] sections 1 and
2531 8.5), or (3) be otherwise authorized by the Printer's administrator-configured security policy to
2532 renew Per-Printer Subscription Objects for the target Printer

2533 6. Cancel-Subscription (see section 11.2.7): The authenticated user (see [RFC2911] section 8.3)
2534 performing this operation MUST (1) be the owner of the Subscription Object, (2) have Operator
2535 or Administrator access rights for the Printer (see [RFC2911] sections 1 and 8.5), or (3) be
2536 otherwise authorized by the Printer's administrator-configured security policy to cancel the
2537 target Subscription Object.

2538 The standard security concerns (delivery to the right user, privacy of content, tamper proof content)
2539 apply to each Delivery Method. Some Delivery Methods are more secure than others. Each Delivery
2540 Method Document MUST discuss its Security Considerations.

2541 **26.2 Printer security threats**

2542 Notification trap door: If a Printer supports the OPTIONAL "notify-attributes" Subscription Template
2543 attribute (see section 5.3.4) where the client can request that the Printer return any specified Job,
2544 Printer, and Subscription object attributes, the Printer MUST apply the same security policy to these
2545 requested attributes in the Get-Notifications request as it does for the Get-Jobs, Get-Job-Attributes,
2546 Get-Printer-Attributes, and Get-Subscription-Attributes requests.

2547 **26.3 Notification Recipient security threats**

2548 Unwanted Events Notifications (spam): For any Push Delivery Method, by far the biggest security
2549 concern is the abuse of notification: delivering unwanted Event Notifications to third parties (i.e.,
2550 spam). The problem is made worse by notification addresses that may be redistributed to multiple
2551 parties. There exist scenarios where third party notification is used (see Scenario #2 and #3 in [ipp-
2552 not-req]). Any fully secure solution would require active agreement of all recipients before delivering
2553 anything.

2554 **27 Contributors**

2555 The following people made significant contributions to the design and review of this specification:

2556 Scott A. Isaacson
2557 Novell, Inc.
2558 122 E 1700 S
2559 Provo, UT 84606

2560

2561

Phone: 801-861-7366

2562

Fax: 801-861-2517

2563

e-mail: sisaacson@novell.com

2564

2565

Roger deBry

2566

Utah Valley State College

2567

Orem, UT 84058

2568

2569

Phone: (801) 222-8000

2570

EMail: debryro@uvsc.edu

2571

2572

Jay Martin

2573

Underscore Inc.

2574

9 Jacqueline St.

2575

Hudson, NH 03051-5308

2576

603-889-7000

2577

fax: 775-414-0245

2578

e-mail: jkm@underscore.com

2579

2580

Michael Shepherd

2581

Xerox Corporation

2582

800 Phillips Road MS 128-51E

2583

Webster, NY 14450

2584

2585

Phone: 716-422-2338

2586

Fax: 716-265-8871

2587

e-mail: mshepherd@usa.xerox.com

2588

2589

Ron Bergman

2590

Hitachi Koki Imaging Solutions

2591

1757 Tapo Canyon Road

2592

Simi Valley, CA 93063-3394

2593

2594

Phone: 805-578-4421

2595

Fax: 805-578-4001

2596

Email: rbergma@hitachi-hkis.com

2597

28 Author's Addresses

2598

Robert Herriot

2599

706 Colorado Ave.

2600

Palo Alto, CA 94303

2601

2602 Phone: 650-327-4466
2603 Fax: 650-327-4466
2604 Email: bob@herriot.com

2605
2606 Tom Hastings
2607 Xerox Corporation
2608 737 Hawaii St. ESAE 231
2609 El Segundo, CA 90245

2610
2611 Phone: 310-333-6413
2612 Fax: 310-333-5514
2613 e-mail: hastings@cp10.es.xerox.com

2614
2615 IPP Web Page: <http://www.pwg.org/ipp/>
2616 IPP Mailing List: ipp@pwg.org

2617
2618 To subscribe to the ipp mailing list, send the following email:

- 2619 1) send it to majordomo@pwg.org
2620 2) leave the subject line blank
2621 3) put the following two lines in the message body:
2622 subscribe ipp
2623 end

2624
2625 Implementers of this specification document are encouraged to join the IPP Mailing List in order to
2626 participate in any discussions of clarification issues and review of registration proposals for additional
2627 attributes and values. In order to reduce spam the mailing list rejects mail from non-subscribers, so
2628 you must subscribe to the mailing list in order to send a question or comment to the mailing list.

2629 **29 Appendix G - Description of the base IPP documents (Informative)**

2630 The base set of IPP documents includes:

- 2631 Design Goals for an Internet Printing Protocol [RFC2567]
2632 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
2633 Internet Printing Protocol/1.1: Model and Semantics [RFC2911]
2634 Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
2635 Internet Printing Protocol/1.1: Implementer's Guide [RFC3196]
2636 Mapping between LPD and IPP Protocols [RFC2569]

2637
2638 The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed
2639 printing functionality, and it enumerates real-life scenarios that help to clarify the features that need to
2640 be included in a printing protocol for the Internet. It identifies requirements for three types of users:
2641 end users, operators, and administrators. It calls out a subset of end user requirements that are satisfied
2642 in IPP/1.0 [RFC2566, RFC2565]. A few OPTIONAL operator operations have been added to IPP/1.1
2643 [RFC2911, RFC2910].

2644 The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document
2645 describes IPP from a high level view, defines a roadmap for the various documents that form the suite
2646 of IPP specification documents, and gives background and rationale for the IETF IPP working group's
2647 major decisions.

2648 The "Internet Printing Protocol/1.1: Model and Semantics" document describes a simplified model
2649 with abstract objects, their attributes, and their operations. The model introduces a Printer and a Job.
2650 The Job supports multiple documents per Job. The model document also addresses how security,
2651 internationalization, and directory issues are addressed.

2652 The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the
2653 abstract operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It also
2654 defines the encoding rules for a new Internet MIME media type called "application/ipp". This
2655 document also defines the rules for transporting over HTTP a message body whose Content-Type is
2656 "application/ipp". This document defines the 'ipp' scheme for identifying IPP printers and jobs.

2657 The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to
2658 implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some
2659 of the considerations that may assist them in the design of their client and/or IPP object
2660 implementations. For example, a typical order of processing requests is given, including error
2661 checking. Motivation for some of the specification decisions is also included.

2662 The "Mapping between LPD and IPP Protocols" document gives some advice to implementers of
2663 gateways between IPP and LPD (Line Printer Daemon) implementations.

2664 **30 Appendix H - Full Copyright Statement (Informative)**

2665 Copyright (C) The Internet Society (1998,1999,2000,2001,2002). All Rights Reserved

2666 This document and translations of it may be copied and furnished to others, and derivative works that
2667 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published
2668 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright
2669 notice and this paragraph are included on all such copies and derivative works. However, this
2670 document itself may not be modified in any way, such as by removing the copyright notice or
2671 references to the Internet Society or other Internet organizations, except as needed for the purpose of
2672 developing Internet standards in which case the procedures for copyrights defined in the Internet
2673 Standards process must be followed, or as required to translate it into languages other than English.

2674 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or
2675 its successors or assigns.

2676 This document and the information contained herein is provided on an “AS IS” basis and THE
2677 INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL
2678 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
2679 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
2680 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
2681 PARTICULAR PURPOSE.

2682 **Acknowledgement**

2683

2684 Funding for the RFC Editor function is currently provided by the Internet Society.