

1 Internet Printing Protocol WG
2 INTERNET-DRAFT
3 <draft-ietf-ipp-not-spec-~~1009~~.txt>
4 Updates RFC 2910 and 2911
5 [Target Category: standards track]
6 Expires: ~~December 27, 2002~~ April 10, 2003

R. Herriot
consultant
T. Hastings
Xerox Corporation
October 10~~June 27~~, 2002

7
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 ~~send~~delivers an asynchronous *Event Notification*
27 to the 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

33 **Table of Contents**

34	1 Introduction.....	7
35	1.1 Notification Overview	7
36	2 Models for Notification	9
37	2.1 Model for Simple Notification (Normative).....	9
38	2.2 Additional Models for Notification (Informative).....	10
39	3 Terminology.....	11
40	3.1 Conformance Terminology.....	11
41	3.2 Other Terminology.....	11
42	4 Object Relationships	14
43	4.1 Printer and Per-Printer Subscription Objects.....	14
44	4.2 Printer, Job and Per-Job Subscription Objects.....	14
45	5 Subscription Object.....	14
46	5.1 Rules for Support of Subscription Template Attributes	15
47	5.2 Rules for Processing Subscription Template Attributes	16
48	5.3 Subscription Template Attributes	19
49	5.3.1 notify-recipient-uri (uri).....	20
50	<u>5.3.1.1 notify-schemes-supported (1setOf uriScheme).....</u>	<u>21</u>
51	5.3.2 notify-pull-method (type2 keyword).....	21
52	<u>5.3.2.1 notify-pull-method-supported (1setOf type2 keyword).....</u>	<u>21</u>
53	5.3.3 notify-events (1setOf type2 keyword).....	21
54	<u>5.3.3.1 notify-events-default (1setOf type2 keyword).....</u>	<u>22</u>
55	<u>5.3.3.2 notify-events-supported (1setOf type2 keyword).....</u>	<u>22</u>
56	<u>5.3.3.3 notify-max-events-supported (integer(2:MAX)).....</u>	<u>22</u>
57	5.3.3.4 Standard Values for Subscribed Events.....	22
58	5.3.3.4.1 No Events.....	22
59	5.3.3.4.2 Subscribed Printer Events.....	23
60	5.3.3.4.3 Subscribed Job Events	24
61	5.3.3.5 Rules for Matching of Subscribed Events	25
62	5.3.3.5.1 Rules for Matching of Printer Events	25
63	5.3.3.5.2 Rules for Matching of Job Events.....	25
64	5.3.3.5.3 Special Cases for Matching Rules	26
65	5.3.4 notify-attributes (1setOf type2 keyword).....	26
66	<u>5.3.4.1 notify-attributes-supported (1setOf type2 keyword).....</u>	<u>27</u>
67	5.3.5 notify-user-data (octetString(63))	28
68	5.3.6 notify-charset (charset)	28
69	5.3.7 notify-natural-language (naturalLanguage).....	29
70	5.3.8 notify-lease-duration (integer(0:67108863)).....	29
71	<u>5.3.8.1 notify-lease-duration-default (integer(0:67108863)).....</u>	<u>30</u>

72	5.3.8.2 notify-lease-duration-supported (1setOf (integer(0: 67108863) rangeOfInteger(0:67108863)))	
73	_____	30
74	5.3.9 notify-time-interval (integer(0:MAX))	30
75	5.4 Subscription Description Attributes.....	31
76	5.4.1 notify-subscription-id (integer (1:MAX)).....	32
77	5.4.2 notify-sequence-number (integer (0:MAX)).....	32
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)	34
81	5.4.6 notify-job-id (integer(1:MAX))	34
82	5.4.7 notify-subscriber-user-name (name(MAX))	34
83	6 Printer Description Attributes Related to Notification	35
84	6.1 printer-state-change-time (integer(1:MAX))	35
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).....	36
88	8 Attributes Only in Event Notifications	36
89	8.1 notify-subscribed-event (type2 keyword)	36
90	8.2 notify-text (text(MAX)).....	36
91	9 Event Notification Content.....	37
92	9.1 Content of Machine Consumable Event Notifications	39
93	9.1.1 Event Notification Content Common to All Events	40
94	9.1.2 Additional Event Notification Content for Job Events.....	40
95	9.1.3 Additional Event Notification Content for Printer Events.....	41
96	9.2 Content of Human Consumable Event Notification	41
97	9.2.1 Event Notification Content Common to All Events	42
98	9.2.2 Additional Event Notification Content for Job Events.....	43
99	9.2.3 Additional Event Notification Content for Printer Events.....	44
100	10 Delivery Methods.....	45
101	11 Operations for Notification.....	46
102	11.1 Subscription Creation Operations.....	46
103	11.1.1 Create-Job-Subscriptions Operation	47
104	11.1.1.1 Create-Job-Subscriptions Request	47
105	11.1.1.1.1 notify-job-id (integer(1:MAX))	48
106	11.1.1.2 Create-Job-Subscriptions Response.....	48
107	11.1.2 Create-Printer-Subscriptions operation.....	49
108	11.1.2.1 Create-Printer-Subscriptions Request.....	49
109	11.1.2.2 Create-Printer-Subscriptions Response	50
110	11.1.3 Job Creation Operations – Extensions for Notification.....	50
111	11.1.3.1 Job Creation Request	50

112	11.1.3.2 Job Creation Response.....	51
113	11.2 Other Operations.....	52
114	11.2.1 Restart-Job Operation – Extensions for Notification.....	52
115	11.2.2 Validate-Job Operation – Extensions for Notification.....	52
116	11.2.3 Get-Printer-Attributes – Extensions for Notification.....	53
117	11.2.4 Get-Subscription-Attributes operation.....	53
118	11.2.4.1 Get-Subscription-Attributes Request.....	54
119	11.2.4.1.1 “notify-subscription-id” (integer (1:MAX)).....	54
120	11.2.4.1.2 “requested-attributes” (1setOf keyword).....	54
121	11.2.4.2 Get-Subscription-Attributes Response.....	55
122	11.2.5 Get-Subscriptions operation.....	56
123	11.2.5.1 Get-Subscriptions Request.....	56
124	11.2.5.1.1 “notify-job-id” (integer(1:MAX)).....	57
125	11.2.5.1.2 “limit” (integer(1:MAX)).....	57
126	11.2.5.1.3 “requested-attributes” (1setOf type2 keyword).....	57
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.....	59
131	11.2.6.1.1 “notify-subscription-id” (integer (1:MAX)).....	59
132	11.2.6.1.2 “notify-lease-duration” (integer(0:MAX)).....	59
133	11.2.6.2 Renew-Subscription Response.....	60
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.....	61
137	11.2.7.1.1 “notify-subscription-id” (integer (1:MAX)).....	61
138	11.2.7.2 Cancel-Subscription Response.....	62
139	12 Status Codes.....	62
140	12.1 successful-ok-ignored-subscriptions (0x0003).....	62
141	12.2 client-error-ignored-all-subscriptions (0x0414).....	63
142	13 Status Codes in Subscription Attributes Groups.....	63
143	13.1 client-error-uri-scheme-not-supported (0x040C).....	63
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).....	64
147	13.5 successful-ok-ignored-or-substituted-attributes (0x0001).....	64
148	14 Encodings of Additional Attribute Tags.....	64
149	15 Conformance Requirements.....	64
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).....	69
157	Object relationships	69
158	20.2 Printer Object and Per-Printer Subscription Objects	70
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.....	71
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	74
167	24.4 Status code Registrations	74
168	24.5 Attribute Group tag Registrations.....	75
169	24.6 Registration of Events.....	75
170	24.7 Registration of Event Notification Delivery Methods	76
171	24.7.1 Requirements for Registration of Event Notification Delivery Methods	76
172	24.7.1.1 Required Characteristics	76
173	24.7.1.2 Naming Requirements	76
174	24.7.1.3 Functionality Requirements	77
175	24.7.1.4 Usage and Implementation Requirements	77
176	24.7.1.5 Publication Requirements	77
177	24.7.2 Registration Procedure.....	77
178	24.7.2.1 Present the proposal to the Community.....	77
179	24.7.2.2 Delivery Method Reviewer.....	77
180	24.7.2.3 IANA Registration	78
181	24.7.3 Delivery Method Document Registrations	78
182	24.7.4 Registration Template.....	79
183	25 Internationalization Considerations	79
184	26 Security Considerations	79
185	26.1 Client access rights	80
186	26.2 Printer security threats	81
187	26.3 Notification Recipient security threats.....	81
188	27 Contributors	81

189	28 Author's Addresses	82
190	29 Appendix G - Description of the base IPP documents (Informative)	83
191	30 Appendix H - Full Copyright Statement (Informative)	84
192		
193	Tables	
194	Table 1 – Subscription Template Attributes	20
195	Table 2 – Subscription Description Attributes	31
196	Table 3 – Printer Description Attributes Associated with Notification	35
197	Table 4 – Operation-id assignments	36
198	Table 5 – Attributes in Event Notification Content	40
199	Table 6 – Additional Event Notification Content for Job Events	41
200	Table 7 – Combinations of Events and Subscribed Events for “job-impressions-completed”	41
201	Table 8 – Additional Event Notification Content for Printer Events	41
202	Table 9 – Printer Name in Event Notification Content	43
203	Table 10 – Event Name in Event Notification Content	43
204	Table 11 – Event Time in Event Notification Content	43
205	Table 12 – Job Name in Event Notification Content	44
206	Table 13 – Job State in Event Notification Content	44
207	Table 14 – Printer State in Event Notification Content	45
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” [~~ipp-prog~~[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 [senddelivers](#) an asynchronous *Event Notification* to the specified *Notification Recipient* via the
238 specified *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 [senddeliver](#) the Event
250 Notification.
- 251 4. Some opaque data that the Printer [senddelivers](#) to the Notification Recipient in the Event
252 Notification. [For example](#), [T](#)he Notification Recipient might use this opaque data as a
253 forwarding address for the Event Notification.

- 254 5. The charset to use in text fields within an Event Notification
255 6. The natural language to use in the text fields of the Event Notification
256 7. The requested lease time in seconds for the Subscription Object

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

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

274 For each of the above operations:

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

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

- 286 - **Restart-Job operation:** When a client performs the Restart-Job operation [RFC2911], the
287 Printer re-uses the same Job and its Subscription Objects.
- 288 - **Validate-Job operation:** When a client performs this operation, a client can include zero or
289 more Subscription Template Attributes Groups in the request. The Printer determines if it

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

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

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

319 2 Models for Notification

320 2.1 Model for Simple Notification (Normative) ~~(Simple Case)~~

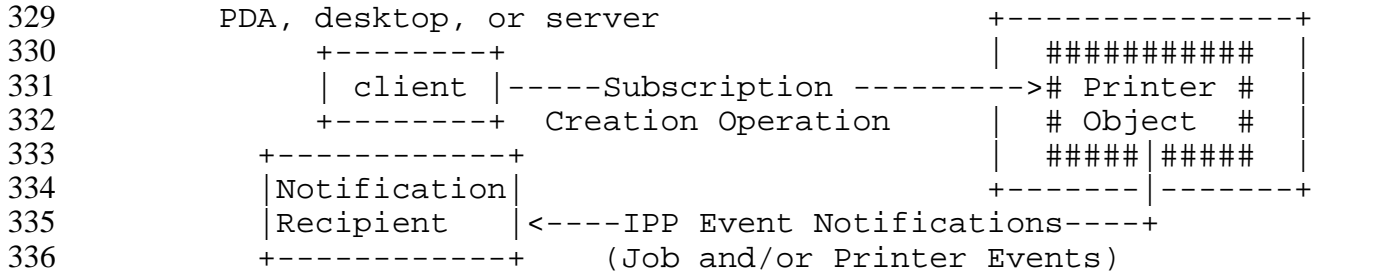
321 As part of a Subscription Creation Operation, an IPP Printer (i.e., located in an output device or a
322 server) creates one or more Subscription Objects. In a Subscription Creation Operation, the client
323 specifies the Notification Recipient to which the Printer is to deliver Event Notifications.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

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

326

327 embedded printer:

328



337 **Figure 1 – Model for Notification**

338 **2.2 Additional Models for Notification (Informative)**

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

340 **2.3 Model for Notification with Cascading Printers**

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

344 –directly to the Notification Recipient specified by the Subscribing Client or

345 –via the Print Server to the Notification Recipient specified by the Subscribing Client.

346 See Appendix 16 for more details.

347 **2.4 Distributed Model for Notification**

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

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

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

358 ~~Appendix 17 describes this example in detail.~~

359 ~~2.5 Extended Notification Recipient~~

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

364 ~~Appendix 18 describes this example in detail.~~

365 3 Terminology

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

368 3.1 Conformance Terminology

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

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

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

379 3.2 Other Terminology

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

384 ~~**Administrator**—A human user who establishes policy for and configures the print system.~~

385 **Compound Event Notification** – two or more Event Notifications that a Printer ~~send~~**delivers** together
386 as a single ~~entity~~**request or response**. The Delivery Method Document specifies whether the Delivery
387 Method supports Compound Event Notifications.

388 **Delivery Method** – the mechanism by which the Printer delivers ~~the an~~ Event Notification, e.g., via
389 ~~email or via an Event Notification Delivery Method protocol defined for delivering IPP Event~~
390 ~~Notifications.~~

391 **Delivery Method Document** – a document, separate from this document, that defines a Delivery
392 Method.

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

397 **Event Life** – For a Pull Delivery Method, the length of time in seconds after an Event occurs during
398 which the Printer will ~~return~~ retain that Event ~~in response to a request~~ for **delivery in an** Event
399 Notifications. After the Event Life expires, the Printer will no longer ~~return~~ deliver an Event
400 Notification for that Event in such a response.

401 **Event Notification** – the information about an Event that the Printer ~~send~~ delivers when an Event
402 occurs.

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

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

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

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

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

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

417 **Notification** – when not in the phrases 'Event Notification' and 'Notification Recipient' — the
418 concepts of this specification, i.e., Events, Subscription Objects, and Event Notifications.

419 **Notification Recipient** – the entity to which the Printer ~~sends~~ delivers an Event Notification. For Push
420 Delivery Methods, the IPP Printer sends the Notifications to a Notification Recipient. For Pull
421 Delivery Methods, the Notification Recipient is acting in the role of an IPP client and requests Event

422 Notifications and so the terms “client” and “Notification Recipient” are used interchangeably with such
423 Delivery Methods. For example, see [ipp-get-method].

424 ~~**Operator**—A human user who carries out the policy established by the Administrator and controls the~~
425 ~~day to day running of the print system.~~

426 **Per-Job Subscription Object** – A Subscription Object that is associated with a single Job. The
427 Create-Job-Subscriptions operation and Job Creation operations create such an object.

428 **Per-Printer Subscription Object** – A Subscription Object that is associated with the Printer as a
429 whole. The Create-Printer-Subscriptions operation creates such an object.

430 ~~**Printer**—the software that supports an output device or print server (see IPP/1.1 [RFC2911] which~~
431 ~~uses the terms Printer and Printer object interchangeably). This document extends the IPP/1.1 Printer~~
432 ~~definition to include the software that implements Subscription Creation Operations and the sending of~~
433 ~~Event Notifications, even if the software for such a Printer would be distributed across a network (see~~
434 ~~section 2.3).~~

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

437 **Pull Delivery Method** – The Printer saves Event Notifications for some event life time and expects the
438 Notification Recipient to request Event Notifications. The Printer ~~returns~~ delivers the Event
439 Notifications in a response to such a request.

440 **Push Delivery Method** –The Printer ~~send~~ delivers the Event Notification shortly after an Event occurs.
441 ~~For some Push Delivery Methods, the Notification Recipient MUST send a response; for others it~~
442 ~~MUST NOT send a response.~~

443 **Subscribed Event** – an Event that the Subscribing Client expresses interest in by making it a value of
444 the “notify-events” attribute on a Subscription Object.

445 **Subscribed Job Event** – a Subscribed Event that is a Job Event.

446 **Subscribed Printer Event** – a Subscribed Event that is a Printer Event.

447 **Subscribing Client** – The client that creates the Subscription Object.

448 **Subscription Attributes Group** – The attributes group in a response that contains Subscription Object
449 attributes.

450 **Subscription Creation Operation** – An operation that creates a Subscription Object: Job Creation
451 operations, Create-Job-Subscriptions operation, Create-Printer-Subscriptions operation. In the context
452 of a Job Creation operation, a Subscription Creation Operation is the part of the Job Creation operation
453 that creates a one or more Subscription objects. The Restart-Job operation [RFC2911] is not considered
454 a Subscription Creation Operation, since the Printer re-uses the Job’s existing Subscription Objects,
455 rather than creating any new Subscription Objects.

456 **Subscription Creation Request** – The request portion of a Subscription Creation Operation.

457 **Subscription Description Attributes** – Subscription Object attributes that a Printer supplies during a
458 Subscription Creation Operation.

459 **Subscription Object** – An object containing a set of attributes that indicate: the Notification Recipient
460 ([for Push Delivery Method only](#)), the Delivery Method, the Subscribed Events that cause the Printer to
461 [send-deliver](#) an Event Notification, and the information to [send-include](#) in an Event Notification.

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

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

467 **4 Object Relationships**

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

470 **4.1 Printer and Per-Printer Subscription Objects**

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

473 **4.2 Printer, Job and Per-Job Subscription Objects**

- 474 1. A Printer object is associated with zero or more Job objects.
- 475 2. Each Job object is associated with exactly one Printer object.
- 476 3. A Job object is associated with zero or more Per-Job Subscription Objects.
- 477 4. Each Per-Job Subscription Object is associated with exactly one Job object.

478 **5 Subscription Object**

479 A Subscribing Client creates a Subscription Object with a Subscription Creation Operation in order to
480 indicate its interest in certain Events. See section 11 for a description of these operations. When an
481 Event occurs, the Subscription Object specifies to the Printer where to [send-deliver](#) Event Notifications
482 [for Push Delivery Methods only](#), how to [send-deliver](#) them, and what to [put-include](#) in them. See
483 section 9 for details on the contents of an Event Notification.

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

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

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

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

493 **5.1 Rules for Support of Subscription Template Attributes**

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

497 Subscription Objects attributes that are Subscription Template Attributes conform to the following
498 rules:

- 499 1. Each attribute's name starts with the prefix string "notify-" and this document calls such
500 attributes "notify-xxx".
- 501 2. For each "notify-xxx" Subscription Object attribute defined in column 1 of Table 1 in section
502 5.3, Table 1 specifies corresponding Printer attributes: "notify-xxx-default", "notify-xxx-
503 supported", "yyy-supported" and "notify-max-xxx-supported" defined in column 2 of Table 1.
504 Note "xxx" stands for the same string in each case and "yyy" stands for some other string.
- 505 3. If a Printer supports "notify-xxx" in column 1 of Table 1, then the Printer MUST support all
506 associated attributes specified in column 2 of Table 1. For example, Table 1 shows that if the
507 Printer supports "notify-events", it MUST support "notify-events-default", "notify-events-
508 supported" and "notify-max-events-supported".
- 509 4. If a Printer does not support "notify-xxx" in column 1 of Table 1, then the Printer MUST NOT
510 support any associated "notify-yyy" attributes specified in column 2 of Table 1. For example,
511 Table 1 shows that if the Printer doesn't support "notify-events", it MUST NOT support "notify-
512 events-default", "notify-events-supported" and "notify-max-events-supported". Note this rule
513 does not apply to attributes whose names do not start with the string "notify-" and are thus
514 defined in another object and used by other attributes.
- 515 5. Most "notify-xxx" attributes have a corresponding "yyy-supported" attribute that specifies the
516 supported values for "notify-xxx". Column 2 of Table 1 specifies the name of each "yyy-

517 supported” attribute. The naming rules of IPP/1.1 (see [RFC2911]) are used when “yyy-
518 supported” is “notify-xxx-supported”.

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

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

531 5.2 Rules for Processing Subscription Template Attributes

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

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

- 539 1. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer supports it and
540 its value, the Printer MUST populate the attribute on the created Subscription Object.
- 541 2. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer doesn’t
542 support it or its value, the Printer MUST NOT populate the attribute on the created Subscription
543 Object with it. The Printer MUST do one of the following:
 - 544 a) If the value of the “notify-xxx” attribute is unsupported, the Printer MUST return the attribute
545 with its value in the Subscription Attributes Group of the response.
 - 546 b) If “notify-xxx” is an unsupported attribute, the Printer MUST return the attribute in the
547 Subscription Attributes Group of the response with the ‘unsupported’ out-of-band value.

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

- 552 3. If a client is REQUIRED to supply a “notify-xxx” attribute from column 1 of Table 1 and the
553 Printer doesn’t support the supplied value, the Printer MUST NOT create a Subscription Object.
554 The rules for Unsupported Attributes in step #2 still apply.
- 555 4. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 and the attribute is
556 REQUIRED for the client to supply, the Printer MUST reject the Subscription Creation Operation
557 (including Job Creation operations) without creating a Subscription Object, and MUST return in
558 the response:
- 559 c) the status code ‘client-error-bad-request’ AND
- 560 d) no Subscription Attribute Groups.
- 561 5. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 that is OPTIONAL for
562 the client to supply, and column 2 of Table 1 either:
- 563 a) specifies a “notify-xxx-default” attribute, the Printer MUST behave as if the client had supplied
564 the “notify-xxx-default” attribute (see step #1) and populate the Subscription object with the
565 value of the “notify-xxx-default” attribute as part of the Subscription Creation operation (unlike
566 Job Template attributes where the Printer does not populate the Job object with defaults – see
567 [RFC2911]) OR
- 568 b) does not specify a “notify-xxx-default” attribute, the Printer MUST populate the “notify-xxx”
569 attribute on the Subscription Object according to the definition of the “notify-xxx” attribute in a
570 section 5.3. For some attributes, the “notify-xxx” is populated with the value of some other
571 attribute, and for others, the “notify-xxx” is NOT populated on the Subscription object at all.
- 572 6. A Printer MUST create a Subscription Object for each Subscription Template Attributes group in a
573 request unless the Printer:
- 574 a) encounters some attributes in a Subscription Template Attributes Group that require the Printer
575 not to create the Subscription Object OR
- 576 b) would create a Per-Job Subscription Object when it doesn’t have space for another Per-Job
577 Subscription Object OR
- 578 c) would create a Per-Printer Subscription Object when it doesn’t have space for another Per-
579 Printer Subscription Object.
- 580 7. A response MUST contain one Subscription Attributes Group for each Subscription Template
581 Attributes Group in the request (and in the same order) whether the Printer creates a Subscription
582 Object from the Subscription Template Attributes Group or not. However, the attributes in each
583 Subscription Attributes Group can be in any order.
- 584 8. The Printer MUST populate each Subscription Attributes Group of the response such that each
585 contains:

- 586 a) the “notify-subscription-id” attribute (see section 5.4.1), if and only if the Printer creates a
587 Subscription Object.
- 588 b) the “notify-lease-duration” attribute (see section 5.3.8), if and only if the Printer creates a Per-
589 Printer Subscription Object. The value of this attribute is the value of the Subscription Object’s
590 “notify-lease-duration” attribute. This value MAY be different from the client-supplied value
591 (see section 5.3.8). If a client supplies this attribute in the creation of a Per-Job Subscription
592 Object, it MUST appear in this group with the out-of-band value ‘unsupported’ to indicate that
593 the Printer doesn’t support it in this context.
- 594 c) all of the unsupported Subscription Template Attributes from step #2. Note, they are not
595 returned in the Unsupported Attributes Group in order to separate the unsupported attributes for
596 each Subscription Object.
- 597 d) the “notify-status-code” attribute if the Printer does not create the Subscription Object or if
598 there are unsupported attributes from step #2. The possible values of the “notify-status-code”
599 attribute are shown below (see section 13 for more details). The Printer returns the first value
600 in the list below that describes the status.
- 601 ‘client-error-uri-scheme-not-supported’: the Subscription Object was not created because
602 the scheme of the “notify-recipient-uri” attribute is not supported. See section 13.1 for
603 more details about this status code. See step #3 in this section for the case that causes
604 this error, and the resulting step #6a) that causes the Printer not to create the
605 Subscription Object.
- 606 ‘client-error-attributes-or-values-not-supported’: the Subscription Object was not created
607 because the method of the “notify-pull-method” attribute is not supported. See section
608 13.1 for more details about this status code. See step #3 in this section for the case that
609 causes this error, and the resulting step #6a) that causes the Printer not to create the
610 Subscription Object.
- 611 ‘client-error-too-many-subscriptions’: the Subscription Object was not created because the
612 Printer has no space for additional Subscription Objects. The client SHOULD try again
613 later. See section 13.3 for more details about this status code. See steps #6b) and #6c)
614 in this section for the cases that causes this error.
- 615 ‘successful-ok-too-many-events’: the Subscription Object was created without the “notify-
616 events” values included in this Subscription Attributes Group because the “notify-
617 events” attribute contains too many values. See section 13.4 for more details about this
618 status code. See step #2 in this section and section 5.3.3 for the cases that cause this
619 status code.
- 620 ‘successful-ok-ignored-or-substituted-attributes’: the Subscription Object was created but
621 some supplied Subscription Template Attributes are unsupported. These unsupported
622 attributes are also in the Subscription Attributes Group. See section 13.5 for more
623 details about this status code. See step #2 in this section for the cases that cause this
624 status code.

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

631 5.3 Subscription Template Attributes

632 This section contains the Subscription Template Attributes defined for the Subscription and Printer
633 objects.

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

- 635 - **Attribute in Subscription Object:** the name and attribute syntax of each Subscription Object
636 Attribute that is a Subscription Template Attribute
- 637 - **Default and Supported Printer Attributes:** the default attribute and supported Printer
638 attributes that are associated with the attribute in column 1.

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

641 For Push Delivery Methods, a Printer MUST support all attributes in Table 1 below except for “notify-
642 pull-method” and “notify-attributes” (and “notify-pull-method-supported” and “notify-attributes-
643 supported”). For Pull Delivery Methods, a Printer MUST support all attributes in Table 1 below
644 except for “notify-recipient-uri” and “notify-attributes” (and “notify-schemes-supported” and “notify-
645 attributes-supported”). If a Printer supports both Push and Pull Delivery Methods, then it MUST
646 support both “notify-recipient-uri” and “notify-pull-method” attributes.

647 For Pull Delivery Methods, a client MUST supply “notify-recipient-uri” and MAY omit any of the rest
648 of the attributes in column 1 of Table 1 in a Subscription Creation Request. For Push Delivery
649 Methods, a client MUST supply “notify-pull-method” and MAY omit any of the rest of the attributes in
650 column 1 of Table 1 in a Subscription Creation Request. A client MUST NOT supply both “notify-
651 recipient-uri” and “notify-pull-method” attributes in the same Subscription Creation Request.

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

658

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))	

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

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

661 **5.3.1 notify-recipient-uri (uri)**

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

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

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

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

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

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

678 **5.3.1.1 notify-schemes-supported (1setOf uriScheme)**

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

682 **5.3.2 notify-pull-method (type2 keyword)**

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

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

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

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

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

694 **5.3.2.1 notify-pull-method-supported (1setOf type2 keyword)**

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

696 **5.3.3 notify-events (1setOf type2 keyword)**

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

700 A Printer MUST support this attribute.

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

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

706 The number of values of this attribute MUST NOT exceed the value of the “notify-max-events-
707 supported” attribute. A Printer MUST support at least 2 values per Subscription Object. If the number
708 of values supplied by a client in a Subscription Creation Operation exceeds the value of this attribute,

709 the Printer MUST treat extra values as unsupported values and MUST use the value of ‘successful-ok-
710 too-many-events’ for the “notify-status-code” attribute in the Subscription Attributes Group of the
711 response.

712 **5.3.3.1 notify-events-default (1setOf type2 keyword)**

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

714 **5.3.3.2 notify-events-supported (1setOf type2 keyword)**

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

716 **5.3.3.3 notify-max-events-supported (integer(2:MAX))**

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

720 **5.3.3.4 Standard Values for Subscribed Events**

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

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

726 A Printer MUST support the Events indicated as “REQUIRED” and MAY support the Events
727 indicated as “OPTIONAL”.

728 **5.3.3.4.1 No Events**

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

730 **‘none’**: REQUIRED – no Event Notifications for any Events. As the sole value of “notify-events-
731 supported”, this value means that the Printer does not support the sending-delivery of Event
732 Notifications. As the sole value of “notify-events-default”, this value means that a client MUST
733 specify the “notify-events” attribute in order for a Subscription Creation Operation to succeed. If
734 the Printer receives this value as the sole value of a Subscription Creation Operation, it does not
735 create a Subscription Object. If a Printer receives this value with other values of a Subscription
736 Creation Operation, the Printer MUST treat this value as an unsupported value.

737 **5.3.3.4.2 Subscribed Printer Events**

738 The standard keyword values for Subscribed Printer Events are:

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

742
743 This Subscribed Event value has the following sub-values: ‘printer-restarted’ and ‘printer-
744 shutdown’. A client can listen for any of these sub-values if it doesn’t want to listen to all printer-
745 state changes:

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

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

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

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

758
759 This Event value has the following sub-values: ‘printer-media-changed’ and ‘printer-finishings-
760 changed’. A client can listen for any of these sub-values if it doesn’t want to listen to all printer-
761 configuration changes:

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

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

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

775 covers that) and does not include when a job leaves the queue (the ‘job-completed’ Event covers
776 that).

777 5.3.3.4.3 Subscribed Job Events

778 The standard keyword values for Subscribed Job Events are:

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

783
784 This Event value has the following sub-values: ‘job-created’, ‘job-completed’ and ‘job-stopped’.
785 A client can listen for any of these sub-values if it doesn’t want to listen to all ‘job-state changes’.

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

791 **‘job-completed’**: REQUIRED – the job has reached one of the completed states, i.e., the
792 value of the job’s “job-state” attribute has changed to: ‘completed’, ‘aborted’, or
793 ‘canceled’. The Job’s “time-at-completed” and “date-time-at-completed” (if supported)
794 attributes are set (see [RFC2911] section 4.3.14). When a Job completes, a Notification
795 Recipient MAY query the Job using the Get-Job-Attributes operation. To allow such a
796 query, the Printer retains the Job in the Job Retention and/or the Job History phases (see
797 [RFC2911] section 4.3.7.1) for a suitable amount of time that depends on
798 implementation and the Delivery Methods supported. The Printer also [senddelivers](#) this
799 Event when a Job is removed with the Purge-Job operation (see [RFC2911] section
800 3.2.9). In this case, the Event Notification MUST report the ‘job-state’ as ‘canceled’
801 and the Job object is no longer present for query.

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

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

811 **‘job-progress’**: OPTIONAL – when the Printer has completed Printing a sheet. See the separate [~~ipp-~~
812 [progRFC3381](#)] specification for additional attributes that a Printer MAY [senddeliver](#) in an Event

813 Notification caused by this Event. The “notify-time-interval” attribute affects this Event by
814 causing the Printer NOT to ~~send~~deliver an Event Notification every time a ‘job-progress’ Events
815 occurs. See section 5.3.9 for full details.

816 5.3.3.5 Rules for Matching of Subscribed Events

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

820 5.3.3.5.1 Rules for Matching of Printer Events

821 ~~Suppose~~Given that the Printer causes Printer Event E to occur, ~~For~~for each Per-Job or Per-Printer
822 Subscription S in the Printer, if E equals a value of this attribute in S or E is a sub-value of a value of
823 this attribute in S, the Printer MUST generate an Event Notification.

824 Consider the example. There are three Subscription Objects each with the Subscribed Printer Event
825 ‘printer-state-changed’. Subscription Object A is a Per-Printer Subscription Object. Subscription
826 Object B is a Per-Job Subscription Object for Job 1, and Subscription Object C is a Per-Job
827 Subscription Object for Job 2. When the Printer enters the ‘stopped’ state, the Printer ~~send~~delivers an
828 Event Notification to the Notification Recipients of Subscription Objects A, B, and C because this is a
829 Printer Event. Note if Job 1 has already completed, the Printer would not ~~send~~deliver an Event
830 Notification for its Subscription Object, even if Job 1 is retained in the Job Retention and/or the Job
831 History phases (see [RFC2911] section 4.3.7.1).

832 5.3.3.5.2 Rules for Matching of Job Events

833 ~~Suppose~~Given that Job J causes Job Event E to occur, ~~For~~for

- 834 1. For each Per-Printer Subscription S in the Printer, if E equals a value of this attribute in S or E is
835 a sub-value of a value of this attribute in S, the Printer MUST generate an Event Notification.
- 836 2. For each Per-Job Subscription S associated with Job J, if E equals a value of this attribute in S
837 or E is a sub-value of a value of this attribute in S, the Printer MUST generate an Event
838 Notification.
- 839 3. For each Per-Job Subscription S that is NOT associated Job J, if E equals a value of this
840 attribute in S or E is a sub-value of a value of this attribute in, the Printer MUST NOT generate
841 an Event Notification from S.

842 Consider the example: There are three Subscription Objects listening for the Job Event ‘job-
843 completed’. Subscription Object A is a Per-Printer Subscription Object. Subscription Object B is a
844 Per-Job Subscription Object for Job 1, and Subscription Object C is a Per-Job Subscription Object for
845 Job 2. In addition, Per-Printer Subscription Object D is listening for the Job Event ‘job-state-changed’.

846 When Job 1 completes, the Printer [senddelivers](#) an Event Notification to the Notification Recipient of
847 Subscription Object A (because it is Per-Printer) and Subscription Object B because it is a Per-Job
848 Subscription Object associated with the Job generating the Event. The Printer also [senddelivers](#) an
849 Event Notification to the Notification Recipient of Subscription Object D because ‘job-completed’ is a
850 sub-value of ‘job-state-changed’ – the value that Subscription Object D is listening for. The Printer
851 does not [senddeliver](#) an Event Notification to the Notification Recipients of Subscription Object C
852 because it is a Per-Job Subscription Object associated with some Job other than the Job generating the
853 Event.

854 5.3.3.5.3 Special Cases for Matching Rules

855 This section contains rule for special cases.

856 If an Event matches Subscribed Events in two different Subscription Objects and the Printer would
857 [senddeliver](#) two identical Event Notifications (except for the “notify-subscription-id” attribute) to the
858 same Notification Recipient using the same Delivery Method, the Printer MUST [senddeliver](#) both
859 Event Notifications. That is, the Printer MUST NOT try to consolidate seemingly identical Event
860 Notifications that occur in separate Subscription objects. Incidentally, the Printer MUST NOT reject
861 Subscription Creation Operations that would create this scenario.

862 If an Event matches two values of this “notify-events” attribute in a single Subscription object (e.g., a
863 value and its sub-value), a Printer MAY [senddeliver](#) one Event Notification for each matched value in
864 the Subscription Object or it MAY [senddeliver](#) only one Event Notification per Subscription Object.
865 The rules in sections 5.3.3.5.1 and 5.3.3.5.2 are purposefully [ambiguous-flexible](#) about the number of
866 Event Notifications sent when Event E matches two or more values in a Subscription Object.

867 Consider the example: There are two Per-Printer Subscription Objects when a Job completes.
868 Subscription Object A has the Subscribed Job Event ‘job-state-changed’. Subscription Object B has
869 the Subscribed Job Events ‘job-state-changed’ and ‘job-completed’. The Printer [senddelivers](#) an Event
870 Notification to the Notification Recipient of Subscription Object A with the value of ‘job-state-
871 changed’ for the “notify-subscribing-event” attribute. The Printer [senddelivers](#) either one or two Event
872 Notifications to the Notification Recipient of Subscription Object B, depending on implementation. If
873 it [senddeliver](#)s two Event Notifications, one has the value of ‘job-state-changed’ for the “notify-
874 subscribing-event” attribute, and the other has the value of ‘job-completed’ for the “notify-
875 subscribing-event” attribute. If it [senddelivers](#) one Event Notification, it has the value of either ‘job-
876 state-changed’ or ‘job-completed’ for the “notify-subscribing-event” attribute, depending on
877 implementation. The algorithm for choosing such a value is implementation dependent.

878 5.3.4 notify-attributes (1setOf type2 keyword)

879 This attribute contains a set of attribute names. When a Printer [senddelivers](#) a Machine Consumable
880 Event Notification, it includes a fixed set of attributes (see section 9.1). If this attribute is present and
881 the Event Notification is Machine Consumable, the Printer also includes the attributes specified by this
882 attribute.

883 A Printer MAY support this attribute.

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

888 Each keyword value of this attribute on a Subscription Object MUST be a value of the “notify-
889 attributes-supported (1setOf type2 keyword)” Printer attribute ([see section 5.3.4.1](#)). The “notify-
890 attributes-supported” MAY contain any Printer attribute, Job attribute or Subscription Object attribute
891 that the Printer supports in an Event Notification. It MUST NOT contain any of the attributes in
892 Section 9.1 that a Printer automatically puts in an Event Notification; it would be redundant. If a client
893 supplies an attribute in Section 9.1, the Printer MUST treat it as an unsupported attribute value of the
894 “notify-attributes” attribute.

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

- 897 a) a Subscription attribute, the Printer MUST use the attribute N in the Subscription Object that is
898 being used to generate the Event Notification.
- 899 b) a Job attribute and the Printer is generating an Event Notification from a Per-Job Subscription
900 Object S, the Printer MUST use the attribute N in the Job object associated with S.
- 901 c) a Job attribute and the Printer is generating an Event Notification from a Per-Printer Subscription
902 Object and the Event is:
- 903 • a Job Event, the Printer MUST use the attribute N in the Job object that caused the Event.
 - 904 • a Printer Event, the Printer MUST use the attribute N in the active Job.

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

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

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

911 **5.3.4.1 notify-attributes-supported (1setOf type2 keyword)**

912 [See sections 5.1 and 5.2 for the behavior of “xxx-supported” Subscription Template Printer attributes.](#)

913 5.3.5 notify-user-data (octetString(63))

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

- 916 - the identity of the Subscriber
- 917 - a path or index to some Subscriber information
- 918 - a key that identifies to the Notification Recipient the ultimate recipient of the Event
919 Notification
- 920 - the id for a Notification Recipient that had previously registered with an Instant Messaging
921 Service

922 A Printer MUST support this attribute.

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

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

930 5.3.6 notify-charset (charset)

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

933 A Printer MUST support this attribute.

934 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
935 this attribute in Subscription Creation Operation or supplies an unsupported value, the Printer MUST
936 populate this attribute in the Subscription Object with the value of the “attributes-charset” operation
937 attribute, which is a REQUIRED attribute in all IPP requests (see [RFC2911]). If the value of the
938 “attributes-charset” attribute is unsupported, the Printer MUST populate this attribute in the
939 Subscription Object with the value of the Printer’s “charset-configured” attribute. There is no “notify-
940 charset-default” Printer attribute.

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

943 **5.3.7 notify-natural-language (naturalLanguage)**

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

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

948 A client **MAY** supply this attribute in a Subscription Creation Operation. If the client does not supply
949 this attribute in Subscription Creation Operation or supplies an unsupported value, the Printer **MUST**
950 populate this attribute in the Subscription Object with the value of the “attributes-natural-language”
951 operation attribute, which is a **REQUIRED** attribute in all IPP requests (see [RFC2911] [section 3.1.4](#)).
952 If the value of the “attributes-natural-language” attribute is unsupported, the Printer **MUST** populate
953 this attribute in the Subscription Object with the value of the Printer’s “natural-language-configured”
954 attribute (see [RFC2911] [section 4.4.19](#)). There is no “notify-natural-language-default” Printer
955 attribute.

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

958 **5.3.8 notify-lease-duration (integer(0:67108863))**

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

963 This attribute is not present on a Per-Job Subscription Object because the Subscription Object lasts
964 exactly as long as the associated Job object. See discussion of the ‘job-completed’ event in section
965 5.3.3.4.3 about retention of the Job object after completion.

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

967 For a Subscription Object Creation operation of a Per-Job Subscription Object, the client **MUST NOT**
968 supply this attribute. If the client does supply this attribute, the Printer **MUST** treat it as an
969 unsupported attribute.

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

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

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

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

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

990 **5.3.8.1 notify-lease-duration-default (integer(0:67108863))**

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

992 **5.3.8.2 notify-lease-duration-supported (1setOf (integer(0: 67108863) |** 993 **rangeOfInteger(0:67108863)))**

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

995 **5.3.9 notify-time-interval (integer(0:MAX))**

996 The ‘job-progress’ Event occurs each time that a Printer completes a sheet. Some Notification
997 Recipients -do not want to receive an Event Notification every time this Event occurs. This attribute
998 allows a Subscribing Client to request how often it wants to receive Event Notifications for ‘job-
999 progress’ Events. The value of this attribute MAY be any nonnegative integer (0,MAX) indicating the
1000 minimum number of seconds between ‘job-progress’ Event Notifications.

1001 The Printer MUST support this attribute if and only if the Printer supports the ‘job-progress’ Event.

1002 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply
1003 this attribute in the Subscription Creation Operation, the Subscription Object either (1) MAY contain
1004 the “notify-time-interval” attribute with a ‘0’ value or (2) NEED NOT contain this attribute at all.
1005 There is no “notify-time-interval-default” Printer attribute.

1006 There is no “notify-time-interval-supported” Printer attribute.

1007 If the ‘job-progress’ Event occurs and a Subscription Object contains the ‘job-progress’ Event as a
1008 value of the ‘notify-events’ attribute, there are two cases to consider:

- 1009 1. This attribute is not present on the Subscription Object or has the value of 0. The Printer MUST
 1010 generate and [senddeliver](#) an Event Notification (as is the case with other Events).
- 1011 2. This attribute is present with a nonzero value of N:
- 1012 a) If the Printer has not sent an Event Notification for the ‘job-progress’ Event for the associated
 1013 Subscription Object within the past N seconds, the Printer MUST [senddeliver](#) an Event
 1014 Notification for the Event that just occurred. Note when the Printer completes the first page of a
 1015 Job, this rule implies that the Printer [senddelivers](#) an Event Notification for a Per-Job
 1016 Subscription Object.
- 1017 b) Otherwise, the Printer MUST NOT generate or [senddeliver](#) an Event Notification for the
 1018 associated Subscription Object. The Printer MUST NOT increase the value of the “notify-
 1019 sequence-number” Subscription Object attribute (i.e., the sequence of values of the “notify-
 1020 sequence-number” attribute counts the Event Notifications that the Printer sent and not the
 1021 Events that do not cause an Event Notification to be sent).
- 1022 It is RECOMMENDED that a Subscribing Client use this attribute when it subscribes to the ‘job-
 1023 progress’ Event, and that the value be sufficiently large to limit the frequency with which the Printer
 1024 [senddelivers](#) Event Notifications requests.
- 1025 This attribute MUST NOT effect any Events other than ‘job-progress’.

1026 5.4 Subscription Description Attributes

1027 Subscription Description Attributes are those attributes that a Printer adds to a Subscription Object at
 1028 the time of its creation.

1029 A Printer MUST support all attributes in this Table 2.

1030 A client MUST NOT supply the attributes in Table 2 in a Subscription Template Attributes Group of a
 1031 Subscription Creation Operation. There are no corresponding default or supported attributes.

1032 **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))

1033

1034 5.4.1 notify-subscription-id (integer (1:MAX))

1035 This attribute identifies a Subscription Object instance with a number that is unique within the context
1036 of the Printer. The Printer generates this value at the time it creates the Subscription Object.

1037 A Printer MUST support this attribute.

1038 The Printer MAY assign the value of this attribute sequentially as it creates Subscription Objects.
1039 However, if there is no security on Subscription objects, sequential assignment exposes the system to a
1040 passive traffic monitoring threat.

1041 The Printer SHOULD avoid re-using recent values of this attribute during continuous operation of the
1042 Printer as well as across power cycles. Then a Subscribing Client is unlikely to find that a stale
1043 reference accesses a new Subscription Object.

1044 The 0 value is not permitted in order to allow for compatibility with “job-id” and with SNMP index
1045 values, which also cannot be 0.

1046 5.4.2 notify-sequence-number (integer (0:MAX))

1047 The value of this attribute indicates the number of times that the Printer has generated and attempted to
1048 [send](#)[deliver](#) an Event Notification for this Subscription object. When an Event Notification contains
1049 this attribute, the Notification Recipient can determine whether it missed some Event Notifications
1050 (i.e., numbers skipped) or received duplicates (i.e., same number twice).

1051 A Printer MUST support this attribute.

1052 When the Printer creates a Subscription Object, it MUST populate this attribute with a value of 0. This
1053 value indicates that the Printer has not sent any Event Notifications for this Subscription Object.

1054 Each time the Printer [send](#)[delivers](#) a newly generated Event Notification, it MUST increase the value of
1055 this attribute by 1. For some Delivery Methods, the Printer MUST include this attribute in each Event
1056 Notification, and the value MUST be the value after it is increased by 1. That is, the value of this
1057 attribute in the first Event Notification after Subscription object creation MUST be 1, the second
1058 MUST be 2, etc. If a Delivery Method is defined such that the Notification Recipient returns a
1059 response, the Printer can re-try [send](#)[deliver](#)ing an Event Notification a certain number of times with the
1060 same sequence number when the Notification Recipient fails to return a response.

1061 If a Subscription Object lasts long enough to reach the value of MAX, its next value MUST be 0, i.e., it
1062 wraps.

1063 5.4.3 notify-lease-expiration-time (integer(0:MAX))

1064 This attribute specifies the time in the future when the lease on the Per-Printer Subscription Object will
1065 expire, i.e. the “printer-up-time” value at which the lease will expire. If the value is 0, the lease never
1066 expires.

1067 A Printer MUST support this attribute.

1068 When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present – the
1069 Subscription Object lasts exactly as long as the associated Job object. See also the discussion of the
1070 ‘job-completed’ event in section 5.3.3.4.3 about retention of the Job object after completion so that a
1071 Notification Recipient can query the Job object after receiving the ‘job-completed’ Event Notification.

1072 When the Printer creates a Per-Printer Subscription Object, it populates this attribute with a value that
1073 is the sum of the values of the Printer’s “printer-up-time” attribute and the Subscription Object’s
1074 “notify-lease-duration” attribute with the following exception. If the value of the Subscription Object’s
1075 “notify-lease-duration” attribute is 0 (i.e., no expiration time), then the value of this attribute MUST be
1076 set to 0 (i.e., no expiration time).

1077 When the Printer powers up, it MUST populate this attribute in each persistent Subscription Object
1078 with a value using the algorithm in the previous paragraph.

1079 When the “printer-up-time” equals the value of this attribute, the Printer MUST delete the Subscription
1080 Object. A client can extend a lease of a Per-Printer Subscription Object with the Renew-Subscription
1081 operation (see section 11.2.6).

1082 Note: In order to compute the number of seconds remaining in a lease for a Per-Printer Subscription
1083 Object, a client can subtract the Subscription’s “notify-printer-up-time” attribute (see section 5.4.4)
1084 from the Subscription’s “notify-lease-expiration-time” attribute.

1085 **5.4.4 notify-printer-up-time (integer(1:MAX))**

1086 This attribute is an alias for the Printer’s “printer-up-time” attribute “ (see [RFC2911] section 4.4.29).
1087 In other words, when this attribute is queried with the Get-Subscriptions or Get-Subscription-Attributes
1088 operations (see sections 11.2.4 and 11.2.5), the value returned is the current value of the Printer’s
1089 “printer-up-time” attribute, rather than the time at which the Subscription Object was created.

1090 A Printer MUST support this attribute.

1091 When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present. When
1092 the Printer creates a Per-Printer Subscription Object, this attribute MUST be present.

1093 Note: this attribute exists in a Per-Printer Subscription Object so that a client using the Get-
1094 Subscription-Attributes or Get-Subscription operations can convert the Per-Printer Subscription’s
1095 “notify-lease-expiration-time” attribute to wall clock time with one request. If the value of the “notify-
1096 lease-expiration-time” attribute is not 0 (i.e., no expiration time), then the difference between the
1097 “notify-lease-expiration-time” attribute and the “notify-printer-up-time” is the remaining number of
1098 seconds on the lease from the current time.

1099 5.4.5 notify-printer-uri (uri)

1100 This attribute identifies the Printer object that created this Subscription Object.

1101 A Printer MUST support this attribute.

1102 During a Subscription Creation Operation, the Printer MUST populate this attribute with the value of
1103 the “printer-uri” operation attribute in the request. From the Printer URI, the client can, for example,
1104 determine what security scheme was used.

1105 5.4.6 notify-job-id (integer(1:MAX))

1106 This attribute specifies whether the containing Subscription Object is a Per-Job or Per-Printer
1107 Subscription Object, and for Per-Job Subscription Objects, it specifies the associated Job.

1108 A Printer MUST support this attribute.

1109 If this attribute is not present, the Subscription Object MUST be a Per-Printer Subscription. If this
1110 attribute is present, the Subscription Object MUST be a Per-Job Subscription Object and this attribute
1111 MUST identify the Job with which the Subscription Object is associated.

1112 Note: This attribute could be useful to a Notification Recipient that receives an Event Notification
1113 generated from a Per-Job Subscription Object and caused by a Printer Event. The Event Notification
1114 gives access to the Printer and the Subscription Object. The Event Notification gives access to the
1115 associated Job only via this attribute. See discussion of the ‘job-completed’ event in section 5.3.3.4.3
1116 about retention of the Job object after completion so that a Notification Recipient can query the Job
1117 object after receiving the ‘job-completed’ Event Notification.

1118 5.4.7 notify-subscriber-user-name (name(MAX))

1119 This attribute contains the name of the user who performed the Subscription Creation Operation.

1120 A Printer MUST support this attribute.

1121 The Printer MUST populates this attribute with the most authenticated printable name that it can obtain
1122 from the authentication service over which the Subscription Creation Operation was received. The
1123 Printer uses the same mechanism for determining the value of this attribute as it does for a Job’s “job-
1124 originating-user-name” (see [RFC2911] section 4.3.6).

1125 Note: To help with authentication, a Subscription Object may have additional private attributes about
1126 the user, e.g., a credential of a principal. Such private attributes are implementation-dependent and not
1127 defined in this document.

1128 6 Printer Description Attributes Related to Notification

1129 This section defines the Printer Description attributes that are related to Notification. Table 3 lists the
 1130 Printer Description attributes, indicates the Printer support required for conformance, and whether or
 1131 not the attribute is READ-ONLY (see section 3.1):

1132 **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

1133

1134 6.1 printer-state-change-time (integer(1:MAX))

1135 This OPTIONAL attribute records the most recent time at which the ‘printer-state-changed’ Printer
 1136 Event occurred whether or not any Subscription objects were listening for this event. This attribute
 1137 helps a client or operator to determine how long the Printer has been in its current state.

1138 A Printer MAY support this attribute and if so, the attribute MUST be READ-ONLY.

1139 On power-up, the Printer MUST populate this attribute with the value of its “printer-up-time” attribute,
 1140 so that it always has a value. Whenever the ‘printer-state-changed’ Printer Event occurs, the Printer
 1141 MUST update this attribute with the value of the Printer’s “printer-up-time” attribute.

1142 6.2 printer-state-change-date-time (dateTime)

1143 This OPTIONAL attribute records the most recent time at which the ‘printer-state-changed’ Printer
 1144 Event occurred whether or not there were any Subscription Objects listening for this event. This
 1145 attribute helps a client or operator to determine how long the Printer has been in its current state.

1146 A Printer MAY support this attribute and if so, the attribute MUST be READ-ONLY.

1147 On power-up, the Printer MUST populate this attribute with the value of its “printer-current-time”
 1148 attribute, so that it always has a value (see [RFC2911] section 4.4.30 on “printer-current-time”).
 1149 Whenever the ‘printer-state-changed’ Printer Event occurs, the Printer MUST update this attribute with
 1150 the value of the Printer’s “printer-current-time” attribute.

1151 7 New Values for Existing Printer Description Attributes

1152 This section contains those attributes for which additional values are added.

1153 7.1 operations-supported (1setOf type2 enum)

1154 The following “operation-id” values are added in order to support the new operations defined in this
1155 document:

1156 **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

1157 8 Attributes Only in Event Notifications

1158 This section contains those attributes that exist only in Event Notifications and do not exist in any
1159 objects.

1160 8.1 notify-subscribed-event (type2 keyword)

1161 This attribute indicates the Subscribed Event that caused the Printer to [senddeliver](#) this Event
1162 Notification. This attribute exists only in Event Notifications.

1163 This attribute MUST contain one of the values of the “notify-events” attribute in the Subscription
1164 Object, i.e., one of the Subscribed Event values. Its value is the Subscribed Event that “matches” the
1165 Event that caused the Printer to [senddeliver](#) this Event Notification. This Subscribed Event value may
1166 be identical to the Event or the Event may be a sub-value of the Subscribed Event. For example, the
1167 ‘job-completed’ Event (which is a sub-event of the ‘job-state-changed’ event) would cause the Printer
1168 to [senddeliver](#) an Event Notification for either the ‘job-completed’ or ‘job-state-changed’ Subscribed
1169 Events and to [senddeliver](#) the ‘job-completed’ or ‘job-state-changed’ value for this attribute,
1170 respectively. See section 5.3.3.5 for the “matching” rules of Subscribed Events and for additional
1171 examples.

1172 The Delivery Method Document specifies whether the Printer includes the value of this attribute in an
1173 Event Notification.

1174 8.2 notify-text (text(MAX))

1175 This attribute contains a Human Consumable text message (see section 9.2). This message describes
1176 the Event and is encoded as plain text, i.e., ‘text/plain’ with the charset specified by Subscription
1177 Object’s “notify-charset” attribute.

1178 The Delivery Method Document specifies whether the Printer includes this attribute in an Event
1179 Notification.

1180 9 Event Notification Content

1181 This section defines the Event Notification content that the Printer [senddelivers](#) when an Event occurs.

1182 When an Event occurs, the Printer MUST find each Subscription object whose “notify-events”
1183 attribute “matches” the Event. See section 5.3.3.5 for details on “matching”. For each matched
1184 Subscription Object, the Printer MUST create an Event Notification with the content and format that
1185 the Delivery Method Document specifies. The content contains the value of attributes specified by the
1186 Delivery Method Document. The Printer obtains the values immediately after the Event occurs. For
1187 example, if the “printer-state” attribute changes from ‘idle’ to ‘processing’, the Event ‘printer-state-
1188 changed’ occurs and the Printer puts various attributes into the Event Notification, including “printer-
1189 up-time” and “printer-state” with the values that they have immediately after the Event occurs, i.e., the
1190 value of “printer-state” is ‘processing’.

1191 Event Notification Ordering:

1192 When a Printer [senddelivers](#) Event Notifications, the Event Notifications from any given Subscription
1193 Object MUST be in time stamp order, i.e., in order of increasing “printer-up-time” attribute value in
1194 the Event Notification (see Table 5). These Event Notifications MAY be interleaved with those from
1195 other Subscription Objects, as long as those others are also in time stamp order. The Printer MUST
1196 observe these ordering requirements whether [senddelivering](#) multiple pending Events as multiple
1197 separate Event Notifications or together in a single Compound Event Notification.

1198 If a Subscribing Client wants the Printer to [senddeliver](#) certain Event Notifications in time stamp order,
1199 the Subscribing Client uses a single Subscription Object. Even so, depending on the underlying
1200 transport, the actual order that a Notification Recipient receives separate Event Notifications may differ
1201 from the order sent by the Printer (e.g., email).

1202 Example: Consider two Per-Printer Subscription Objects: SO1 and SO2. SO1 requests ‘job-state-
1203 changed’ events and SO2 requests ‘printer-state-changed’ events. The number in parens is the time
1204 stamp. The following Event Notification sequences are the only ones that conform to the ordering
1205 requirements for the Printer to [senddeliver](#) the Event Notifications:

1206 (a) SO1: ‘job-created’ (1000), SO1: ‘job-stopped’ (1005), SO1: ‘job-completed’ (1009), SO2: ‘printer-
1207 stopped’ (1005)

1208 (b) SO1: ‘job-created’ (1000), SO1: ‘job-stopped’ (1005), SO2: ‘printer-stopped’ (1005), SO1: ‘job-
1209 completed’ (1009)

1210 (c) SO1: ‘job-created’ (1000), SO2: ‘printer-stopped’ (1005), SO1: ‘job-stopped’ (1005), SO1: ‘job-
1211 completed’ (1009)

- 1212 (d) SO2: ‘printer-stopped (1005), SO1: ‘job-created’ (1000), SO1: ‘job-stopped’ (1005), SO1: ‘job-
1213 completed’ (1009)
- 1214 Examples (b) and (c) are interleaved; examples (a) and (d) are not interleaved and are not appropriate
1215 for some Delivery Methods.
- 1216 If two different Events occur simultaneously, or nearly so (e.g., “printer-up-time” has the same value
1217 for both), the Printer MUST create a separate Event Notification for each Event, even if the associated
1218 Subscription Object is the same for both Events. However, the Printer MAY combine these distinct
1219 Event Notifications into a single Compound Event Notification if the Delivery Method supports
1220 Compound Event Notifications. For example, suppose that two nearly-simultaneously Events
1221 represent two successive ‘printer-state-changed’ Events, one from ‘idle’ to ‘processing’ and another
1222 from ‘processing’ to ‘stopped’. These two Events have the same name but are different instances of
1223 the Event. Then the Printer MUST create a separate Event Notification for each Event and SHOULD
1224 accurately report the “printer-state” of the first Event as ‘processing’ and the second Event as
1225 ‘stopped’.
- 1226 If a Subscription Object contains more than one Subscribed Event, and several Events occur in quick
1227 succession each matching a different Subscribed Event in the Subscription Object, the Printer MUST
1228 NOT generate a single Event Notification from several of these Events, but MAY combine distinct
1229 Event Notifications into a single Compound Event Notification if the Delivery Method supports
1230 Compound Event Notifications.
- 1231 After the Printer has created the Event Notification, the Printer delivers it via either a:
- 1232 Push Delivery Method: The Printer [senddeliver](#)s the Event Notification shortly after an Event
1233 occurs. For some Push Delivery Methods, the Notification Recipient MUST [senddeliver](#) a
1234 response; for others it MUST NOT [senddeliver](#) a response.
- 1235 Pull Delivery Method: The Printer saves Event Notifications for some Event Life and expects
1236 the Notification Recipient to request Event Notifications. The Printer returns the Event
1237 Notifications in a response to such a request.
- 1238 If an error that meets the following conditions occurs, the Printer MUST cancel the Subscription
1239 Object.
- 1240 a) the error occurs during the [senddelivering](#) of an Event Notification generated from Subscription
1241 Object S AND
- 1242 b) the error would continue to occur every time the Printer [senddelivers](#) an Event Notification
1243 generated from Subscription Object S in the future.
- 1244 For example, if the address of the “notify-recipient-uri” of Subscription Object A references a non-
1245 existent target and the Printer determines this fact, it MUST delete Subscription Object A.
- 1246 The next two sections describe the values that a Printer [senddelivers](#) in the content of Machine
1247 Consumable and Human Consumable Event Notifications, respectively.

1248 The tables in the sub-sections of this section contain the following columns:

1249 a) **Source Value:** the name of the attribute that supplies the value for the Event Notification.
1250 Asterisks in this field refer to a note below the table.

1251 b) **SendDelivers:** if the Printer supports the value (column 1) on the Source Object (column 3) the
1252 Delivery Method MUST specify:

1253 **MUST:** that the Printer MUST [senddeliver](#) the value.

1254 **SHOULD:** either that the Printer MUST [senddeliver](#) the value or that the value is
1255 incompatible with the Delivery Method.

1256 **MAY:** that the Printer MUST, SHOULD, MAY, MUST NOT, SHOULD NOT, or NEED
1257 NOT [senddeliver](#) the value. The Delivery Method specifies the level of conformance for the
1258 Printer.

1259 c) **Source Object:** the object from which the source value comes. If the object is “Event
1260 Notification”, the Printer fabricates the value when it [senddeliver](#)s the Event Notification. See
1261 section 8.

1262 9.1 Content of Machine Consumable Event Notifications

1263 This section defines the attributes that a Delivery Method MUST mention in a Delivery Method
1264 Document when specifying the Machine Consumable Event Notification’s contents.

1265 This document does not define the order of attributes in Event Notifications. However, Delivery
1266 Method Documents MAY define the order of some or all of the attributes.

1267 A Delivery Method Document MUST specify additional attributes (if any) that a Printer
1268 implementation [senddeliver](#)s in a Machine Consumable Event Notification.

1269 Notification Recipients MUST be able to accept Event Notifications containing attributes they do not
1270 recognize. What a Notification Recipient does with an unrecognized attribute is implementation-
1271 dependent. Notification Recipients MAY attempt to display unrecognized attributes anyway or MAY
1272 ignore them.

1273 The next three sections define the attributes in Event Notification Contents that are:

1274 1. for all Events

1275 2. for Job Events only

1276 3. for Printer Events only

1277 **9.1.1 Event Notification Content Common to All Events**1278 This section lists the attributes that a Delivery Method Document **MUST** specify for all Events.

1279 Table 5 lists potential values in each Event Notification.

1280 **Table 5 – Attributes in Event Notification Content**

Source Value	SendDelivers	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

1281

1282 *A Printer **MUST** [senddeliver](#) this value only if and only if it supports the Printer’s “printer-current-
1283 time” attribute.1284 ** If the Subscription Object does not contain a “notify-user-data” attribute and the Delivery Method
1285 Document **REQUIRES** the Printer to [senddeliver](#) the “notify-user-data” source value in the Event
1286 Notification, the Printer **MUST** [senddeliver](#) an octet-string of length 0.1287 *** The last three rows represent additional attributes that a client **MAY** request via the “notify-
1288 attributes” attribute. A Printer **MAY** support the “notify-attributes” attribute. The Delivery Method
1289 **MUST** say that the Printer **MUST**, **SHOULD**, **MAY**, **MUST NOT**, **SHOULD NOT**, or **NEED NOT**
1290 support the “notify-attributes” attribute and specific values of this attribute. The Delivery Method
1291 **MAY** say that support for the “notify-attributes” is conditioned on support of the attribute by the
1292 Printer or it **MAY** say that Printer **MUST** support the “notify-attributes” attribute if the Printer supports
1293 the Delivery Method.1294 **9.1.2 Additional Event Notification Content for Job Events**1295 This section lists the additional attributes that a Delivery Method Document **MUST** specify for Job
1296 Events. See Table 6.

1297

Table 6 – Additional Event Notification Content for Job Events

Source Value	SendDelivers	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

1298

1299

1300

* The Printer MUST [senddeliver](#) the “job-impressions-completed” attribute in an Event Notification only for the combinations of Events and Subscribed Events shown in Table 7.

1301

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’

1302

1303 9.1.3 Additional Event Notification Content for Printer Events

1304

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

1305

1306

Table 8 – Additional Event Notification Content for Printer Events

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

1307

1308 9.2 Content of Human Consumable Event Notification

1309

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.

1310

1311

1312

Such a Delivery Method MUST specify the following information and a Printer SHOULD [senddeliver](#) it:

1313

- 1314 a) the Printer name (see Table 9)
- 1315 b) the time of the Event (see Table 11)
- 1316 c) for Printer Events only:
- 1317 i) the Event (see Table 10) and/or Printer state information (see Table 14)
- 1318 d) for Job Events only:
- 1319 i) the job identity (see Table 12)
- 1320 ii) the Event (see Table 10) and/or Job state information (see Table 13)

1321

1322 The subsections of this section specify the attributes that a Printer MUST use to obtain this

1323 information.

1324 A Delivery Method Document MUST specify additional information (if any) that a Printer

1325 implementation [senddelivers](#) in a Human Consumable Event Notification or in the “notify-text”

1326 attribute.

1327 A client MUST NOT request additional attributes via the “notify-attributes” attribute because this

1328 attribute works only for Machine Consumable Event Notifications.

1329 Notification Recipients MUST NOT expect to be able to parse the Human Consumable Event

1330 Notification contents or the value of the “notify-text” attribute.

1331 The next three sections define the attributes in Event Notification Contents that are:

- 1332 a) for all Events
- 1333 b) for Job Events only
- 1334 c) for Printer Events only
- 1335

1336 9.2.1 Event Notification Content Common to All Events

1337 This section lists the source of the information that a Delivery Method MUST specify for all Events.

1338 There is a separate table for each piece of information. Each row in the table represents a source value

1339 for the information and the values are listed in order of preference, with the first one being the

1340 preferred one. An implementation SHOULD use the source value from the earliest row in each table.

1341 It MAY use the source value from another row instead, or it MAY combine the source values from

1342 several rows. An implementation is free to determine the best way to present this information.

1343 In all tables of this section, all rows contain a “MAY” in order to state that the Delivery Method

1344 specifies the conformance.

1345 Table 9 lists the source of the information for the Printer Name. The “printer-name” is more user-

1346 friendly unless the Notification Recipient is in a place where the Printer name is not meaningful. For

1347 example, an implementation could have the intelligence to [senddeliver](#) the value of the “printer-name”

1348 attribute to a Notification Recipient that can access the Printer via value of the “printer-name” attribute

1349 and otherwise [senddeliver](#) the value of the “notify-printer-uri” attribute.

1350

Table 9 – Printer Name in Event Notification Content

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

1351

1352

1353

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

1354

Table 10 – Event Name in Event Notification Content

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

1355

1356

1357

1358

1359

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

1360

Table 11 – Event Time in Event Notification Content

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

1361

1362 9.2.2 Additional Event Notification Content for Job Events

1363

1364

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

1365

1366

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

1367

Table 12 – Job Name in Event Notification Content

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

1368

1369 Table 13 lists the source of the information for the job state. If a Printer supports the “job-state-
 1370 message” and “job-detailed-state-message” attributes, it SHOULD use those attributes for the job state
 1371 information, otherwise, it should fabricate such information from the “job-state” and “job-state-
 1372 reasons”. For some Events, a Printer MAY combine this information with Event information.

1373

Table 13 – Job State in Event Notification Content

Source Value	SendDelivers	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

1374

1375 9.2.3 Additional Event Notification Content for Printer Events

1376 This section lists the source of the additional information that a Delivery Method MUST specify for
 1377 Printer Events.

1378 Table 14 lists the source of the information for the printer state. If a Printer supports the “printer-state-
 1379 message”, it SHOULD use that attribute for the job state information, otherwise it SHOULD fabricate
 1380 such information from the “printer-state” and “printer-state-reasons”. For some Events, a Printer MAY
 1381 combine this information with Event information.

1382

Table 14 – Printer State in Event Notification Content

Source Value	SendDeliver rs	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

1383 **10 Delivery Methods**

1384 A Delivery Method is the mechanism, i.e., protocol, by which the Printer delivers an Event Notification
 1385 to a Notification Recipient. There are several potential Delivery Methods for Event Notifications,
 1386 standardized, as well as proprietary. This specification REQUIRES that the ‘ippget’ Pull Delivery
 1387 Method [ipp-get-method] be supported. Conforming implementations MAY support additional Push
 1388 or Pull Delivery Methods as well. This document does not define any of these delivery mechanisms.
 1389 Each Delivery Method MUST be defined in a Delivery Method Document that is separate from this
 1390 document. New Delivery Methods will be created as needed using an extension to the registration
 1391 procedures defined in [RFC2911]. Such documents are registered with IANA (see section 24.7.3).

1392 The following sorts of Delivery Methods are possible:

- 1393 – The Notification Recipient polls for Event Notifications at intervals directed by the Printer
- 1394 – The Printer [senddelivers](#) Event Notifications to the Notification Recipient using http as the
 1395 transport.
- 1396 – The Printer [senddelivers](#) an email message.

1397 This section specifies how to define a Delivery Method Document and what to put in such a document.

1398 A Delivery Method Document MUST contain an exact copy of the following paragraph, caption and
 1399 table. In addition, column 2 of the table in the Delivery Method Document MUST contain answers to
 1400 questions in column 1 for the Delivery Method. Also, the Delivery Method document MUST contain a
 1401 reference to this document and call that reference [ipp-ntfy] because the table contains an [ipp-ntfy]
 1402 reference.

1403 If a Printer supports this Delivery Method, the following are its characteristics.

1404

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 send 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?	

1405

1406 11 Operations for Notification

1407 This section defines all of the operations for Notification. Section 7.1 assigns the “operation-id” for
 1408 each operation. The following two sub-sections define Subscription Creation Operations, and other
 1409 operations.

1410 11.1 Subscription Creation Operations

1411 This section defines the Subscription Creation Operations. The first section on Create-Job-
 1412 Subscriptions gives most of the information. The other Subscription Creation Operations refer to the

1413 section on Create-Job-Subscriptions, even though the Create-Job-Subscriptions operation is the only
1414 OPTIONAL operation in this document (see section 12).

1415 A Printer MUST support Create-Printer-Subscriptions and the Subscription Template Attributes Group
1416 in Job Creation operations. It MAY support Create-Job-Subscriptions operations.

1417 **11.1.1 Create-Job-Subscriptions Operation**

1418 The operation creates one or more Per-Job Subscription Objects. The client supplies one or more
1419 Subscription Template Attributes Groups each containing one or more of Subscription Template
1420 Attributes (defined in section 5.3).

1421 Except for errors, the Printer MUST create exactly one Per-Job Subscription Object from each
1422 Subscription Template Attributes Group in the request, even if the newly created Subscription Object
1423 would have identical behavior to some existing Subscription Object. The Printer MUST associate each
1424 newly created Per-Job Subscription Object with the target Job, which is specified by the “notify-job-id”
1425 operation attribute.

1426 The Printer MUST accept the request in any of the target job’s ‘not-completed’ states, i.e., ‘pending’,
1427 ‘pending-held’, ‘processing’, or ‘processing-stopped’. The Printer MUST NOT change the job’s “job-
1428 state” attribute because of this operation. If the target job is in any of the ‘completed’ states, i.e.,
1429 ‘completed’, ‘canceled’, or ‘aborted’, then the Printer MUST reject the request and return the ‘client-
1430 error-not-possible’ status code; the response MUST NOT contain any Subscription Attribute Groups.

1431 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section
1432 8.3) performing this operation MUST (1) be the job owner, (2) have Operator or Administrator access
1433 rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the
1434 Printer’s administrator-configured security policy to create Per-Job Subscription Objects for the target
1435 job. Otherwise the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-
1436 error-not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

1437 **11.1.1.1 Create-Job-Subscriptions Request**

1438 The following groups of attributes are part of the Create-Job-Subscriptions Request:

1439 Group 1: Operation Attributes

1440 Natural Language and Character Set:

1441 The “attributes-charset” and “attributes-natural-language” attributes as described in
1442 [RFC2911] section 3.1.4.1.

1443
1444 Target:

1445 The “printer-uri” attribute which defines the target for this operation as described in
1446 [RFC2911] section 3.1.5.

1447

1448 Requesting User Name:
1449 The "requesting-user-name" attribute SHOULD be supplied by the client as described in
1450 [RFC2911] section 8.3.

1451 **11.1.1.1 notify-job-id (integer(1:MAX))**

1452 The client MUST supply this attribute and it MUST specify the Job object to associate the
1453 Per-Job Subscription with. The value of "notify-job-id" MUST be the value of the "job-id" of
1454 the associated Job object. If the client does not supply this attribute, the Printer MUST reject
1455 this request with a 'client-error-bad-request' status code.
1456

1457 Group 2-N: Subscription Template Attributes

1458 For each occurrence of this group:

1459
1460 The client MUST supply one or more Subscription Template Attributes in any order. See
1461 section 5.3 for a description of each such attribute. See section 5.2 for details on processing
1462 these attributes.

1463 **11.1.1.2 Create-Job-Subscriptions Response**

1464 The Printer MUST return to the client the following sets of attributes as part of a Create-Job-
1465 Subscriptions response:

1466 Group 1: Operation Attributes

1467 Status Message:

1468 In addition to the REQUIRED status code returned in every response, the response
1469 OPTIONALLY includes a "status-message" (text(255)) and/or a "detailed-status-message"
1470 (text(MAX)) operation attribute as described in [RFC2911] sections 13 and 3.1.6.
1471

1472 In this group, the Printer can return any status codes defined in [RFC2911] and section 12.
1473 The following is a description of the important status codes:
1474

1475 **successful-ok:** the Printer created all Subscription Objects requested (see [RFC2911]).

1476 **successful-ok-ignored-subscriptions:** the Printer created some Subscription Objects
1477 requested but some failed. The Subscription Attributes Groups with a "notify-status-
1478 code" attribute are the ones that failed (see section 12.1).

1479 **client-error-ignored-all-subscriptions:** the Printer created no Subscription Objects
1480 requested and all failed. The Subscription Attributes Groups with a "notify-status-
1481 code" attribute are the ones that failed (see section 12.2).

1482 **client-error-not-possible:** For this operation and other Per-Job Subscription operations,
1483 this error can occur because the specified Job has already completed (see [RFC2911]),
1484 whether or not the Job is retained in the Job Retention and/or Job History phases (see
1485 [RFC2911] section 4.3.7.1).

1486

1487

Natural Language and Character Set:

1488

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

1489

[RFC2911] section 3.1.4.2.

1490

1491

Group 2: Unsupported Attributes

1492

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

1493

does not contain any unsupported Subscription Template Attributes; they are returned in the

1494

Subscription Attributes Group (see below).

1495

1496

Group 3-N: Subscription Attributes

1497

These groups MUST be returned unless the Printer is unable to interpret the entire request,

1498

e.g., the “status-code” parameter returned in Group 1 has the value: ‘client-error-bad-request’.

1499

1500

“notify-status-code” (type2 enum):

1501

Indicates the status of this subscription (see section 13 for the status code definitions).

1502

Section 5.2 defines when this attribute MUST be present in this group.

1503

1504

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

1505

1506

11.1.2 Create-Printer-Subscriptions operation

1507

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

1508

The operation creates Per-Printer Subscription Objects instead of Per-Job Subscription Objects, and

1509

associates each newly created Per-Printer Subscription Object with the Printer specified by the

1510

operation target rather than with a specific Job.

1511

The Printer MUST accept the request in any of its states, i.e., ‘idle’, ‘processing’, or ‘stopped’. The

1512

Printer MUST NOT change its “printer-state” attribute because of this operation.

1513

Access Rights: To create Per-Printer Subscription Objects, the authenticated user (see [RFC2911]

1514

section 8.3) performing this operation MUST have (1) Operator or Administrator access rights for this

1515

Printer (see [RFC2911] sections 1 and 8.5), or (2) be otherwise authorized by the Printer’s

1516

administrator-configured security policy to create Per-Printer Subscription Objects for this Printer.

1517

Otherwise, the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-

1518

not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

1519

11.1.2.1 Create-Printer-Subscriptions Request

1520

The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.1) except that the

1521

Operation Attributes group MUST NOT contain the “notify-job-id” attribute. If the client does supply

1522 the “notify-job-id” attribute, then the Printer MUST treat it as any other unsupported Operation
1523 attribute and MUST return it in the Unsupported Attributes group.

1524 **11.1.2.2 Create-Printer-Subscriptions Response**

1525 The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.2).

1526 **11.1.3 Job Creation Operations – Extensions for Notification**

1527 This document extends the Job Creation operations (see section 3.2) to create Subscription Objects as a
1528 part of the operation.

1529 The Job Creation operations are identical to Create-Job-Subscriptions operation with exceptions noted
1530 in this section.

1531 Unlike the Create-Job-Subscriptions operation, a Job Creation operation associates the newly created
1532 Subscription Objects with the Job object created by this operation. The operation succeeds if and only
1533 if the Job creation succeeds. If the Printer does not create some or all of the requested Subscription
1534 Objects, the Printer MUST return a ‘successful-ok-ignored-subscriptions’ status-code instead of a
1535 ‘successful-ok’ status-code, but the Printer MUST NOT reject the operation because of a failure to
1536 create Subscription Objects.

1537 If the Job Creation operation includes a Job Template group, the client MUST supply it after the
1538 Operation Attributes group and before the first Subscription Template Attributes Group.

1539 If a Printer does not support this Notification specification, then it MUST treat the Subscription
1540 Attributes Group like an unknown group and ignore it (see [RFC2911] section 5.2.2). Because the
1541 Printer ignores the Subscription Attributes Group, it doesn’t return them in the response either, thus
1542 indicating to the client that the Printer doesn’t support Notification.

1543 After completion of a successful Job Creation operation, the Printer generates a ‘job-created’ event (see
1544 section 5.3.3.4.3).

1545 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section
1546 8.3) performing this operation MUST either have permission to create Jobs on the Printer or have
1547 Operator or Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5). Otherwise
1548 the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-not-
1549 authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

1550 **11.1.3.1 Job Creation Request**

1551 The groups for this operation are sufficiently different from the Create-Job-Subscriptions operation that
1552 they are all presented here. The following groups of attributes are supplied as part of a Job Creation
1553 Request:

1554 Group 1: Operation Attributes

1555 Same as defined in [RFC2911] for Print-Job, Print-URI, and Create-Job requests.

1556

1557 Group 2: Job Template Attributes

1558 The client **OPTIONALLY** supplies a set of Job Template attributes as defined in [RFC2911]
1559 section 4.2.

1560

1561 Group 3 to N: Subscription Template Attributes

1562 The same as Group 2-N in Create-Job-Subscriptions. See section 11.1.1.1.

1563 Group N+1: Document Content (Print-Job only)

1564 The client **MUST** supply the document data to be processed.

1565

1566 11.1.3.2 Job Creation Response

1567 The Printer **MUST** return to the client the following sets of attributes as part of a Print-Job, Print-URI,
1568 and Create-Job Response:

1569 Group 1: Operation Attributes

1570 Status Message:

1571

1572 As defined in [RFC2911] for Print-Job, Print-URI, and Create-Job requests.

1573

1574 In this group, the Printer can return any status codes defined in [RFC2911] and section 12.

1575 The following is a description of the important status codes:

1576

1577 **successful-ok:** the Printer created the Job and all Subscription Objects requested (see
1578 [RFC2911]).

1579 **successful-ok-ignored-subscriptions:** the Printer created the Job and not all of the
1580 Subscription Objects requested (see section 12.1). This status-code hides
1581 ‘successful-ok-xxx’ status-codes that could reveal problems in Job creation. The
1582 Printer **MUST NOT** return the ‘client-error-ignored-all-subscriptions’ status code for
1583 Job Creation operations because the Printer returns an error status-code only when it
1584 fails to create a Job.

1585

1586 Natural Language and Character Set:

1587 The “attributes-charset” and “attributes-natural-language” attributes as described in
1588 [RFC2911] section 3.1.4.2.

1589

1590 Group 2: Unsupported Attributes

1591 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes. This group
1592 does not contain any unsupported Subscription Template Attributes; they are returned in the
1593 Subscription Attributes Group (see below).
1594

1595 Group 3: Job Object Attributes

1596 The “job-id” of the Job Object just created, etc., as defined in [RFC2911] for Print-Job, Print-
1597 URI, and Create-Job requests.
1598

1599 Group 4 to N: Subscription Attributes

1600 These groups MUST be returned if and only if the client supplied Subscription Template
1601 Attributes and the operation was accepted.

1602 See section 5.2 for details on the contents of each occurrence of this group.
1603

1604 11.2 Other Operations

1605 This section defines other operations on Subscription objects.

1606 11.2.1 Restart-Job Operation – Extensions for Notification

1607 The Restart-Job operation [RFC2911] is neither a Job Creation operation nor a Subscription Creation
1608 operation (see section 3.2). For the Restart-Job operation, the client MUST NOT supply any Job
1609 Subscription Attributes Groups. The Printer MUST treat any supplied Job Subscription Attributes as
1610 unsupported attributes.

1611 For this operation, the Printer does not return a job-id or any Subscription Attributes groups because
1612 the Printer reuses the existing Job object with the same job-id and the existing Per-Job Subscription
1613 Objects with the same subscription-ids. However, after successful completion of this operation, the
1614 Printer generates a ‘job-created’ event (see section 5.3.3.4.3).

1615 11.2.2 Validate-Job Operation – Extensions for Notification

1616 A client can test whether one or more Subscription Objects could be created using the Validate-Job
1617 operation. The client supplies one or more Subscription Template Attributes Groups (defined in
1618 section 5.3), just as in a Job Creation request.

1619 A Printer MUST support this extension to this operation.

1620 The Printer MUST accept requests that are identical to the Job Creation request defined in section
1621 11.1.3.1, except that the request MUST NOT contain document data.

1622 The Printer MUST return the same groups and attributes as the Print-Job operation (section 11.1.3.1)
1623 with the following exceptions. The Printer MUST NOT return a Job Object Attributes Group because

1624 no Job is created. The Printer MUST NOT return the “notify-subscription-id” attribute in any
1625 Subscription Attribute Group because no Subscription Object is created.

1626 If the Printer would succeed in creating a Subscription Object, the corresponding Subscription
1627 Attributes Group either has no ‘status-code’ attribute or a ‘status-code’ attribute with a value of
1628 ‘successful-ok-too-many-events’ or ‘successful-ok-ignored-or-substituted-attributes’ (see sections 5.2
1629 and 13). The status-codes have the same meaning as in Job Creation except the results state what
1630 “would happen”.

1631 The Printer MUST validate Subscription Template Attributes Groups in the same manner as the Job
1632 Creation operations.

1633 11.2.3 Get-Printer-Attributes – Extensions for Notification

1634 This operation is extended so that it returns Printer attributes defined in this document.

1635 A Printer MUST support this extension to this operation.

1636 In addition to the requirements of [RFC2911] section 3.2.5, a Printer MUST support the following
1637 additional values for the “requested-attributes” Operation attribute in this operation and return such
1638 attributes in the Printer Object Attributes group of its response.

- 1639 1. **Subscription Template Attributes:** Each supported attribute in column 2 of Table 1.
- 1640 2. **New Printer Description Attributes:** Each supported attribute in section 6.
- 1641 3. **New Group Name:** The ‘subscription-template’ group name, which names all supported
1642 Subscription Template Attribute in column 2 of Table 1. This group name is also used in the
1643 Get-Subscription-Attributes and Get-Subscriptions operation with an analogous meaning.
- 1644 4. **Extended Group Name:** The ‘all’ group name, which names all Printer attributes according to
1645 [RFC2911] section 3.2.5. In this extension ‘all’ names all attributes specified in [RFC2911]
1646 plus those named in items 1 and 2 of this list.

1647 11.2.4 Get-Subscription-Attributes operation

1648 This operation allows a client to request the values of the attributes of a Subscription Object.

1649 A Printer MUST support this operation.

1650 This operation is almost identical to the Get-Job-Attributes operation (see [RFC2911] section 3.3.4).
1651 The only differences are that the operation is directed at a Subscription Object rather than a Job object,
1652 and the returned attribute group contains Subscription Object attributes rather than Job object
1653 attributes.

1654 Access Rights: The authenticated user (see [RFC2911] section 8.3) performing this operation MUST
1655 (1) be the Subscription Object owner, (2) have Operator or Administrator access rights for this Printer
1656 (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer's administrator-
1657 configured security policy to query the Subscription Object for the target job. Otherwise the Printer
1658 MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-authenticated', or
1659 'client-error-not-authorized' status code as appropriate. Furthermore, the Printer's security policy
1660 MAY limit which attributes are returned, in a manner similar to the Get-Job-Attributes operation (see
1661 [RFC2911] end of section 3.3.4.2).

1662 **11.2.4.1 Get-Subscription-Attributes Request**

1663 The following groups of attributes are part of the Get-Subscription-Attributes request:

1664 Group 1: Operation Attributes

1665 Natural Language and Character Set:

1666 The "attributes-charset" and "attributes-natural-language" attributes as described in section
1667 [RFC2911] 3.1.4.1.

1668

1669 Target:

1670 The "printer-uri" attribute which defines the target for this operation as described in
1671 [RFC2911] section 3.1.5.

1672

1673 Requesting User Name:

1674 The "requesting-user-name" attribute SHOULD be supplied by the client as described in
1675 [RFC2911] section 8.3.

1676 **11.2.4.1.1 "notify-subscription-id" (integer (1:MAX))**

1677 The client MUST supply this attribute. The Printer MUST support this attribute. This
1678 attribute specifies the Subscription Object from which the client is requesting attributes. If the
1679 client omits this attribute, the Printer MUST reject this request with the 'client-error-bad-
1680 request' status code.

1681 **11.2.4.1.2 "requested-attributes" (1setOf keyword)**

1682 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute.
1683 This attribute specifies the attributes of the specified Subscription Object that the Printer
1684 MUST return in the response. Each value of this attribute is either an attribute name (defined
1685 in sections 5.3 and 5.4) or an attribute group name. The attribute group names are:

1686

1687 - 'subscription-template': all attributes that are both defined in section 5.3 and present on
1688 the specified Subscription Object (column 1 of Table 1).

1689 - 'subscription-description': all attributes that are both defined in section 5.4 and present
1690 on the specified Subscription Object (Table 2).

1691 - 'all': all attributes that are present on the specified Subscription Object.

1692
1693 A Printer MUST support all these group names.

1694 If the client omits this attribute, the Printer MUST respond as if this attribute had been
1695 supplied with a value of 'all'.
1696

1697 **11.2.4.2 Get-Subscription-Attributes Response**

1698 The Printer returns the following sets of attributes as part of the Get-Subscription-Attributes Response:

1699 Group 1: Operation Attributes

1700 Status Message:
1701 Same as [RFC2911].

1702
1703 Natural Language and Character Set:
1704 The "attributes-charset" and "attributes-natural-language" attributes as described in
1705 [RFC2911] section 3.1.4.2. The "attributes-natural-language" MAY be the natural language
1706 of the Subscription Object, rather than the one requested.
1707

1708 Group 2: Unsupported Attributes

1709 See [RFC2911] section 3.1.7 and section 3.2.5.2 for details on returning Unsupported
1710 Attributes.

1711
1712 The response NEED NOT contain the "requested-attributes" operation attribute with any
1713 supplied keyword values that were requested by the client but are not supported by the IPP
1714 object. If the Printer object does return unsupported attributes referenced in the "requested-
1715 attributes" operation attribute, the values of the "requested-attributes" attribute returned
1716 MUST include only the unsupported keywords that were requested by the client. If the client
1717 had requested a group name, such as 'all', the resulting unsupported attributes returned MUST
1718 NOT include attribute keyword names described in the standard but not supported by the
1719 implementation.
1720

1721 Group 3: Subscription Attributes

1722 This group contains a set of attributes with their current values. Each attribute returned in this
1723 group:

- 1724
1725 a) MUST be specified by the "requested-attributes" attribute in the request, AND
1726
1727 b) MUST be present on the specified Subscription Object AND
1728
1729 c) MUST NOT be restricted by the security policy in force. For example, a Printer MAY
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.

1730 The Printer can return the attributes of the Subscription Object in any order. The client
1731 MUST accept the attributes in any order.
1732

1733 **11.2.5 Get-Subscriptions operation**

1734 This operation allows a client to retrieve the values of attributes of all Subscription Objects belonging
1735 to a Job or Printer.

1736 A Printer MUST supported this operation.

1737 This operation is similar to the Get-Subscription-Attributes operation, except that this Get-
1738 Subscriptions operation returns attributes from possibly more than one object.

1739 This operation is similar to the Get-Jobs operation (see [RFC2911] section 3.2.6), except that the
1740 operation returns Subscription Objects rather than Job objects.

1741 Access Rights: To query Per-Job Subscription Objects of the specified job (client supplied the “notify-
1742 job-id” operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)
1743 performing this operation MUST (1) be the Subscription Object owner, (2) have Operator or
1744 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise
1745 authorized by the Printer’s administrator-configured security policy to query the Subscription Object
1746 for the target job. To query Per-Printer Subscription Objects of the Printer (client omits the “notify-
1747 job-id” operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)
1748 performing this operation MUST (1) have Operator or Administrator access rights for this Printer (see
1749 [RFC2911] sections 1 and 8.5), or (2) be otherwise authorized by the Printer’s administrator-
1750 configured security policy to query Per-Printer Subscription Objects for the target Printer. Otherwise
1751 the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-not-
1752 authenticated’, or ‘client-error-not-authorized’ status code as appropriate. Furthermore, the Printer’s
1753 security policy MAY limit which attributes are returned, in a manner similar to the Get-Jobs and Get-
1754 Printer-Attributes operations (see [RFC2911] end of sections 3.2.6.2 and 3.2.5.2).

1755 **11.2.5.1 Get-Subscriptions Request**

1756 The following groups of attributes are part of the Get-Subscriptions request:

1757 Group 1: Operation Attributes

1758 Natural Language and Character Set:

1759 The “attributes-charset” and “attributes-natural-language” attributes as described in
1760 [RFC2911] section 3.1.4.1.

1761

1762 Target:

1763 The “printer-uri” attribute which defines the target for this operation as described in
1764 [RFC2911] section 3.1.5.

1765

1766 Requesting User Name:
1767 The “requesting-user-name” attribute SHOULD be supplied by the client as described in
1768 [RFC2911] section 8.3.

1769 **11.2.5.1.1 “notify-job-id” (integer(1:MAX))**

1770 If the client specifies this attribute, the Printer returns the specified attributes of all Per-Job
1771 Subscription Objects associated with the Job whose “job-id” attribute value equals the value
1772 of this attribute. If the client does not specify this attribute, the Printer returns the specified
1773 attributes of all Per-Printer Subscription Objects. Note: there is no way to get all Per-Job
1774 Subscriptions known to the Printer in a single operation. A Get-Jobs operation followed by a
1775 Get-Subscriptions operation for each Job will return all Per-Job Subscriptions.

1776 **11.2.5.1.2 “limit” (integer(1:MAX))**

1777 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. It
1778 is an integer value that determines the maximum number of Subscription Objects that a client
1779 will receive from the Printer even if the “my-subscriptions” attribute constrains which
1780 Subscription Objects are returned. The limit is a “stateless limit” in that if the value supplied
1781 by the client is ‘N’, then only the first ‘N’ Subscription Objects are returned in the Get-
1782 Subscriptions Response. There is no mechanism to allow for the next ‘M’ Subscription
1783 Objects after the first ‘N’ Subscription Objects. If the client does not supply this attribute, the
1784 Printer responds with all applicable Subscription Objects.

1785 **11.2.5.1.3 “requested-attributes” (1setOf type2 keyword)**

1786 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute.
1787 This attribute specifies the attributes of the specified Subscription Objects that the Printer
1788 MUST return in the response. Each value of this attribute is either an attribute name (defined
1789 in sections 5.3 and 5.4) or an attribute group name (defined in section 11.2.4.1). If the client
1790 omits this attribute, the Printer MUST respond as if the client had supplied this attribute with
1791 the one value: ‘notify-subscription-id’.

1792 **11.2.5.1.4 “my-subscriptions” (boolean)**

1793 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. If
1794 the value is ‘false’, the Printer MUST consider the Subscription Objects from all users as
1795 candidates. If the value is ‘true’, the Printer MUST return the Subscription Objects created by
1796 the requesting user of this request. If the client does not supply this attribute, the Printer
1797 MUST respond as if the client had supplied the attribute with a value of ‘false’. The means
1798 for authenticating the requesting user and matching the Subscription Objects is similar to that
1799 for Jobs which is described in [RFC2911] section 8.
1800

1801 **11.2.5.2 Get-Subscriptions Response**

1802 The Printer returns the following sets of attributes as part of the Get-Subscriptions Response:

1803 Group 1: Operation Attributes

1804 Status Message:

1805 Same as [RFC2911].

1806

1807 Natural Language and Character Set:

1808 The “attributes-charset” and “attributes-natural-language” attributes as described in
1809 [RFC2911] section 3.1.4.2.

1810

1811 Group 2: Unsupported Attributes

1812 Same as for Get-Subscription-Attributes.

1813

1814 Groups 3 to N: Subscription Attributes

1815 The Printer responds with one Subscription Attributes Group for each requested Subscription
1816 Object (see the “notify-job-id” attribute in the Operation Attributes Group of this operation).

1817

1818 The Printer returns Subscription Objects in any order.

1819

1820 If the “limit” attribute is present in the Operation Attributes group of the request, the number
1821 of Subscription Attributes Groups in the response MUST NOT exceed the value of the “limit”
1822 attribute.

1823

1824 If there are no Subscription Objects associated with the specified Job or Printer, the Printer
1825 MUST return zero Subscription Attributes Groups and it MUST NOT treat this case as an
1826 error, i.e., the status-code MUST be ‘successful-ok’ unless something else causes the status
1827 code to have some other value.

1828

1829 See the Group 3 response (Subscription Attributes Group) of the Get-Subscription-Attributes
1830 operation (section 11.2.4.2) for the attributes that a Printer returns in this group.

1831

1832 11.2.6 Renew-Subscription operation

1833 This operation allows a client to request the Printer to extend the lease on a Per-Printer Subscription
1834 Object.

1835 The Printer MUST support this operation.

1836 The Printer MUST accept this request for a Per-Printer Subscription Object in any of the target
1837 Printer’s states, i.e., ‘idle’, ‘processing’, or ‘stopped’, but MUST NOT change the Printer’s “printer-
1838 state” attribute.

1839 The Printer MUST reject this request for a Per-Job Subscription Object because it has no lease (see
1840 section 5.4.3). The status code returned MUST be ‘client-error-not-possible’.

1841 *Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation MUST
1842 (1) be the owner of the Per-Printer Subscription Object, (2) have Operator or Administrator access
1843 rights for the Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the
1844 Printer’s administrator-configured security policy to renew Per-Printer Subscription Objects for the
1845 target Printer. Otherwise, the Printer MUST reject the operation and return: the ‘client-error-
1846 forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

1847 **11.2.6.1 Renew-Subscription Request**

1848 The following groups of attributes are part of the Renew-Subscription Request:

1849 Group 1: Operation Attributes

1850 Natural Language and Character Set:

1851 The “attributes-charset” and “attributes-natural-language” attributes as described in
1852 [RFC2911] section 3.1.4.1.

1853

1854 Target:

1855 The “printer-uri” attribute which defines the target for this operation as described in
1856 [RFC2911] section 3.1.5.

1857

1858 Requesting User Name:

1859 The “requesting-user-name” (name(MAX)) attribute SHOULD be supplied by the client as
1860 described in [RFC2911] section 8.3.

1861

1862 **11.2.6.1.1 “notify-subscription-id” (integer (1:MAX))**

1863 The client MUST supply this attribute. The Printer MUST support this attribute. This
1864 attribute specifies the Per-Printer Subscription Object whose lease the Printer MUST renew.
1865 If the client omits this attribute, the Printer MUST reject this request with the ‘client-error-
1866 bad-request’ status code.

1867

1868 Group 2: Subscription Template Attributes

1869 **11.2.6.1.2 “notify-lease-duration” (integer(0:MAX))**

1870 The client MAY supply this attribute. It indicates the number of seconds to renew the lease
1871 for the specified Subscription Object. A value of 0 requests an infinite lease (which MAY
1872 require Operator access rights). If the client omits this attribute, the Printer MUST use the
1873 value of the Printer’s “notify-lease-duration-default” attribute. See section 5.3.8 for more
1874 details.

1875

1876 **11.2.6.2 Renew-Subscription Response**

1877 The Printer returns the following sets of attributes as part of the Renew-Subscription Response:

1878 Group 1: Operation Attributes

1879 Status Message:

1880 Same as [RFC2911].

1881

1882 The following are some of the status codes returned (see [RFC2911]):

1883

1884 **successful-ok:** The operation successfully renewed the lease on the Subscription Object
1885 for the requested duration.1886 **successful-ok-ignored-or-substituted-attributes:** The operation successfully renewed
1887 the lease on the Subscription Object for some duration other than the amount
1888 requested.1889 **client-error-not-possible:** The operation failed because the “notify-subscription-id”
1890 Operation attribute identified a Per-Job Subscription Object.1891 **client-error-not-found:** The operation failed because the “notify-subscription-id”
1892 Operation attribute identified a non-existent Subscription Object.

1893

1894 Natural Language and Character Set:

1895 The “attributes-charset” and “attributes-natural-language” attributes as described in
1896 [RFC2911] section 3.1.4.2. The “attributes-natural-language” MAY be the natural language
1897 of the Subscription Object, rather than the one requested.

1898

1899 Group 2: Unsupported Attributes

1900 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes.

1901

1902 Group 3: Subscription Attributes

1903 The Printer MUST return the following Subscription Attribute:

1904 **11.2.6.2.1 “notify-lease-duration” (integer(0:MAX))**1905 The value of this attribute MUST be the number of seconds that the Printer has granted for the
1906 lease of the Subscription Object (see section 5.3.8 for details, such as the value of this
1907 attribute when the Printer doesn’t support the requested value).1908 **11.2.7 Cancel-Subscription operation**1909 This operation allows a client to delete a Subscription Object and stop the Printer from ~~send~~delivering
1910 more Event Notifications. Once performed, there is no way to reference the Subscription Object.

1911 A Printer MUST supported this operation.

1912 The Printer MUST accept this request in any of the target Printer's states, i.e., 'idle', 'processing', or
1913 'stopped', but MUST NOT change the Printer's "printer-state" attribute.

1914 If the specified Subscription Object is a Per-Job Subscription Object, the Printer MUST accept this
1915 request in any of the target Job's states, but MUST NOT change the Job's "job-state" attribute or affect
1916 the Job.

1917 Note: There is no way to change any attributes on a Subscription Object, except the "notify-lease-
1918 duration" attribute (using the Renew-Subscription operation). In order to change other attributes, a
1919 client performs a Subscription Creation Operation and Cancel-Subscription operation on the old
1920 Subscription Object. If the client wants to avoid missing Event Notifications, it performs the
1921 Subscription Creation Operation first. If this order would create too many Subscription Objects on the
1922 Printer, the client reverses the order.

1923 *Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation MUST
1924 (1) be the owner of the Subscription Object, (2) have Operator or Administrator access rights for the
1925 Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer's
1926 administrator-configured security policy to cancel the target Subscription Object. Otherwise, the
1927 Printer MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-
1928 authenticated', or 'client-error-not-authorized' status code as appropriate.

1929 **11.2.7.1 Cancel-Subscription Request**

1930 The following groups of attributes are part of the Cancel-Subscription Request:

1931 Group 1: Operation Attributes

1932 Natural Language and Character Set:

1933 The "attributes-charset" and "attributes-natural-language" attributes as described in
1934 [RFC2911] section 3.1.4.1.

1935

1936 Target:

1937 The "printer-uri" attribute which defines the target for this operation as described in
1938 [RFC2911] section 3.1.5.

1939

1940 Requesting User Name:

1941 The "requesting-user-name" attribute SHOULD be supplied by the client as described in
1942 [RFC2911] section 8.3.

1943 **11.2.7.1.1 "notify-subscription-id" (integer (1:MAX))**

1944 The client MUST supply this attribute. The Printer MUST support this attribute. This
1945 attribute specifies the Subscription Object that the Printer MUST cancel. If the client omits

1946 this attribute, the Printer MUST reject this request with the ‘client-error-bad-request’ status
1947 code.
1948

1949 **11.2.7.2 Cancel-Subscription Response**

1950 The Printer returns the following sets of attributes as part of the Cancel-Subscription Response:

1951 Group 1: Operation Attributes

1952 Status Message:

1953 Same as [RFC2911].
1954

1955 The following are some of the status codes returned (see [RFC2911]):
1956

1957 **successful-ok:** The operation successfully canceled (deleted) the Subscription Object.

1958 **client-error-not-found:** The operation failed because the “notify-subscription-id”

1959 Operation attribute identified a non-existent Subscription Object.
1960

1961 Natural Language and Character Set:

1962 The “attributes-charset” and “attributes-natural-language” attributes as described in

1963 [RFC2911] section 3.1.4.2. The “attributes-natural-language” MAY be the natural language
1964 of the Subscription Object, rather than the one requested.
1965

1966 Group 2: Unsupported Attributes

1967 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes.
1968

1969 **12 Status Codes**

1970 The following status codes are defined as extensions for Notification and are returned as the value of
1971 the “status-code” parameter in the Operation Attributes Group of a response (see [RFC2911] section
1972 3.1.6.1). Operations in this document can also return the status codes defined in section 13 of
1973 [RFC2911]. The ‘successful-ok’ status code is an example of such a status code.

1974 **12.1 successful-ok-ignored-subscriptions (0x0003)**

1975 The Subscription Creation Operation was unable to create all requested Subscription Objects.

1976 For a Create-Job-Subscriptions or Create-Printer-Subscriptions operation, this status code means that
1977 the Printer created one or more Subscription Objects, but not all requested Subscription Objects.

1978 For a Job Creation operation, this status code means that the Printer created the Job along with zero or
1979 more Subscription Objects. The Printer returns this status code even if other job attributes are
1980 unsupported or in conflict. That is, if an IPP Printer finds a warning that would allow it to return

1981 ‘successful-ok-ignored-subscriptions’ and either ‘successful-ok-ignored-or-substituted-attributes’
1982 and/or ‘successful-ok-conflicting-attributes’, it MUST return ‘successful-ok-ignored-subscriptions’.

1983 **12.2 client-error-ignored-all-subscriptions (0x0414)**

1984 This status code is the same as ‘successful-ok-ignored-subscriptions’ except that only the Create-Job-
1985 Subscriptions and Create-Printer-Subscriptions operation return it. They return this status code only
1986 when the Printer creates zero Subscription Objects.

1987 **13 Status Codes in Subscription Attributes Groups**

1988 This section contains values of the “notify-status-code” (type2 enum) attribute that the Printer returns
1989 in a Subscription Attributes Group in a response when the corresponding Subscription Object:

- 1990 1. is not created or
- 1991 2. is created and some of the client-supplied attributes are not supported.

1992 The following sections are ordered in decreasing order of importance of the status-codes.

1993 **13.1 client-error-uri-scheme-not-supported (0x040C)**

1994 This status code is defined in [RFC2911]. This document extends its meaning and allows it to be in a
1995 Subscription Attributes Group of a response.

1996 The scheme of the client-supplied URI in a “notify-recipient-uri” Subscription Template Attribute in a
1997 Subscription Creation Operation is not supported. See section 5.3.1.

1998 **13.2 client-error-attributes-or-values-not-supported (0x040B)**

1999 This status code is defined in [RFC2911]. This document extends its meaning and allows it to be in a
2000 Subscription Attributes Group of a response.

2001 The method of the client-supplied keyword in a “notify-pull-method” Subscription Template Attribute
2002 in a Subscription Creation Operation is not supported. See section 5.3.2.

2003 **13.3 client-error-too-many-subscriptions (0x0415)**

2004 The number of Subscription Objects supported by the Printer would be exceeded if this Subscription
2005 Object were created (see section 5.2).

2006 **13.4 successful-ok-too-many-events (0x0005)**

2007 The client supplied more Events in the “notify-events” operation attribute of a Subscription Creation
 2008 Operation than the Printer supports, as indicated in its “notify-max-events-supported” Printer attribute
 2009 (see section 5.3.3).

2010 **13.5 successful-ok-ignored-or-substituted-attributes (0x0001)**

2011 This status code is defined in [RFC2911]. This document extends its meaning to include unsupported
 2012 Subscription Template Attributes and it can appear in a Subscription Attributes Group.

2013 **14 Encodings of Additional Attribute Tags**

2014 This section assigns values to two attributes tags as extensions to the encoding defined in [RFC2910]).

2015 The “subscription-attributes-tag” delimits Subscription Template Attributes Groups in requests and
 2016 Subscription Attributes Groups in responses.

2017 The “event-notification-attributes-tag” delimits Event Notifications in Delivery Methods that use an
 2018 IPP-like encoding.

2019 The following table specifies the values for the delimiter tags:

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

2020 **15 Conformance Requirements**

2021 It is OPTIONAL for IPP clients and Printers to implement this Event Notification specification.

2022 **15.1 Conformance requirements for clients**

2023 If this Event Notification specification is implemented by a client, the client MUST support the
 2024 ‘ippget’ Pull Delivery Method and meet the conformance requirements as defined in [ipp-get-method]
 2025 for clients. A client MAY support additional Delivery Methods.

2026 **15.2 Conformance requirements for Printers**

2027 If this Event Notification specification is implemented by a Printer, the Printer MUST:

2028 - meet the Conformance Requirements detailed in section 5 of [RFC2911].

- 2029 - support the Subscription Template Attributes Group in requests and the Subscription
2030 Attributes Group in responses.
- 2031 - support all of the following attributes:
- 2032 a. REQUIRED Subscription Object attributes in section 5.
- 2033 b. REQUIRED Printer Description object attributes in section 6.
- 2034 c. REQUIRED attributes in Event Notification content in section 8.
- 2035 - support the 'ippget' Pull Delivery Method and meet the conformance requirements as defined
2036 in [ipp-get-method] for Printers. The Printer MAY support additional Push and Pull Delivery
2037 Methods.
- 2038 - [senddeliver](#) Event Notifications that conform to the requirements of section 9 and the
2039 requirements of the Delivery Method Document for each supported Delivery Method (the
2040 conformance requirements for Delivery Method Documents is specified in section 10).
- 2041 - for all of the Job Creation Operations that the Printer supports, MUST support the
2042 REQUIRED extensions for notification defined in section 11.1.3.
- 2043 - meet the conformance requirements for operations as described in Table 16 and meet the
2044 requirements for Printers as specified in the indicated sub-sections of section 11:

2045 **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

2046

2047 **16 Appendix [A](#) - Model for Notification with Cascading Printers [\(Informative\)](#)**

2048 With this model (see Figure 2 below), there is an intervening Print server between the human user and
2049 the output-device. So the system effectively has two Printer [objects](#). There are two cases to consider.

- 2050 1. When the Printer 1 (in the server) generates Events, the system behaves like the client and Printer
2051 in Figure 1. In this case, Printer 1 [senddelivers](#) Event Notifications that are shown as Event
2052 Notifications (A) of Figure 2.

2053
2054

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

2055
2056
2057

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

2058
2059
2060
2061
2062
2063
2064

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 **senddelivers** 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.

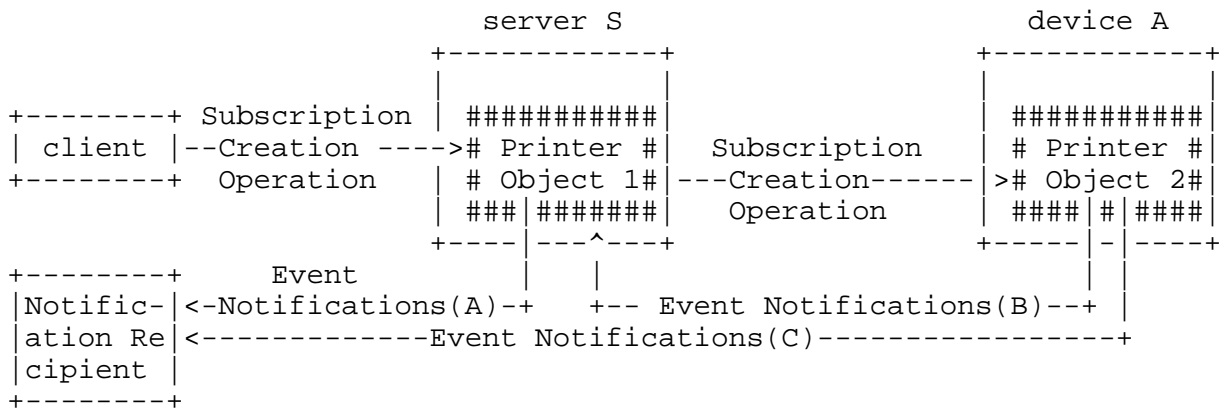
2065
2066
2067

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:

2068
2069
2070

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

2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083



2084

Figure 2 – Model for Notification with Cascading Printers

2085

17 Appendix B - Distributed Model for Notification (Informative)

2086
2087
2088
2089
2090
2091
2092

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

2093 Figure 3 shows this partitioning. The interface between the output device (or serverPrinter object) and
 2094 the remote notification serverice is outside the scope of this document and is intended to be transparent
 2095 to the client and this document. ~~The combination of the output device (or server) and the notification~~
 2096 ~~service together constitute an IPP Printer conforming to this Notification document.~~

2097

2098

2099

2100

2101

2102

2103

2104

2105

2106

2107

2108

2109

2110

2111

2112

2113

2114

2115

2116

2117

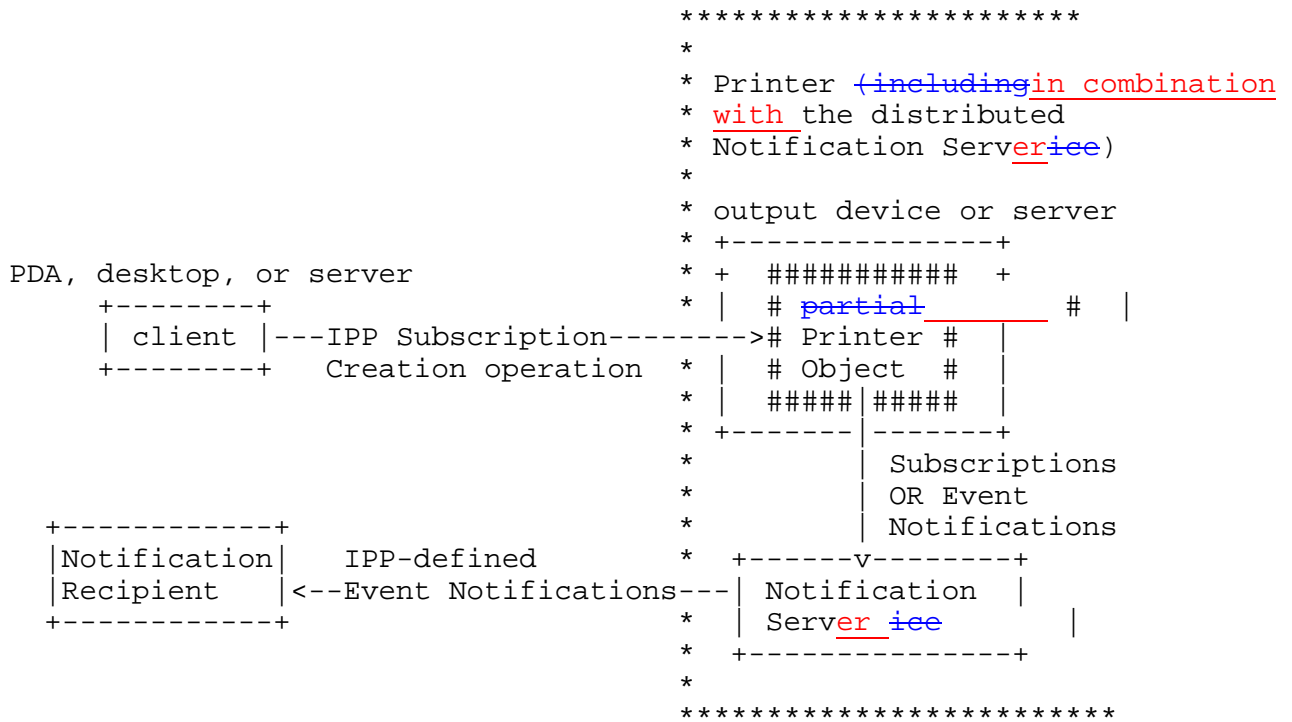
2118

2119

2120

2121

2122



*** = Implementation configuration opaque boundary

2123

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

2124

18 Appendix C - Extended Notification Recipient (Informative)

2125

2126

2127

2128

The model allows for an extended Notification Recipient that is itself a notification serverice 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.

2129

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

2130

2131

2132

2133

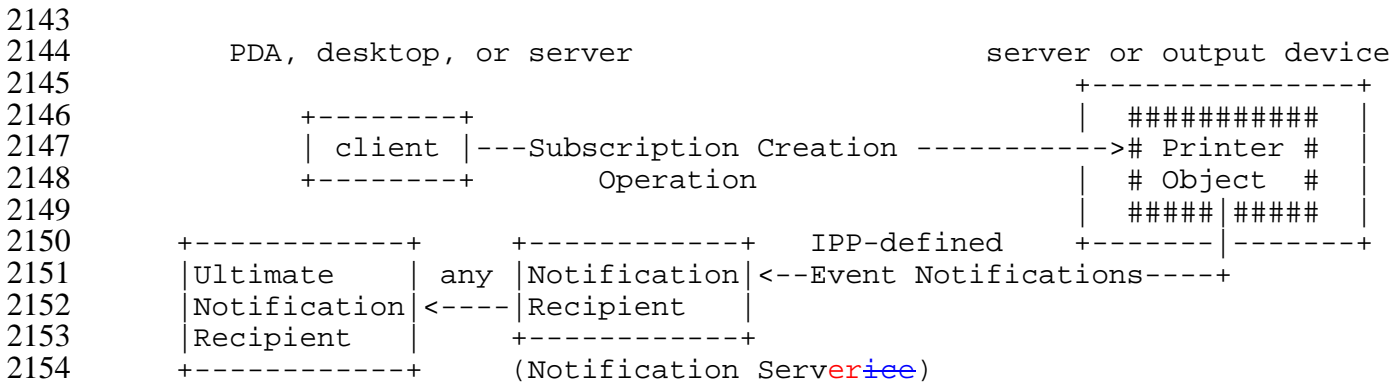
2134

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

2135 extended Notification Recipient (by means outside this document), since it is a notification service in
 2136 its own right.

2137 The IPP Printer treats the extended Notification Recipient like any other Notification Recipient and the
 2138 IPP Printer is not aware of the forwarding. The Delivery Method that the extended Notification
 2139 Recipient uses for delivering the Event Notification to the Ultimate Notification Recipient is beyond
 2140 the scope of this document and is transparent to the IPP Printer.

2141 Examples of this extended Notification Recipient are paging, immediate messaging services, general
 2142 notification services, and NOS vendors' infrastructure. Figure 4 shows this approach.



2155 **Figure 4 – Use of an Extended Notification Recipient transparent to the Printer**

2156 **19 Appendix D - Details about Conformance Terminology (Normative)**

2157 The following paragraphs provide more details about conformance terminology.

2158 **REQUIRED** - an adjective used to indicate that a conforming IPP Printer implementation **MUST**
 2159 support the indicated operation, object, attribute, attribute value, status code, or out-of-band value
 2160 in requests and responses. See [RFC2911] “Appendix A - Terminology for a definition of
 2161 “support”. *Since support of this entire Notification specification is OPTIONAL for*
 2162 *conformance to IPP/1.1, the use of the term REQUIRED in this document means*
 2163 *“REQUIRED if this OPTIONAL Notification specification is implemented”.*

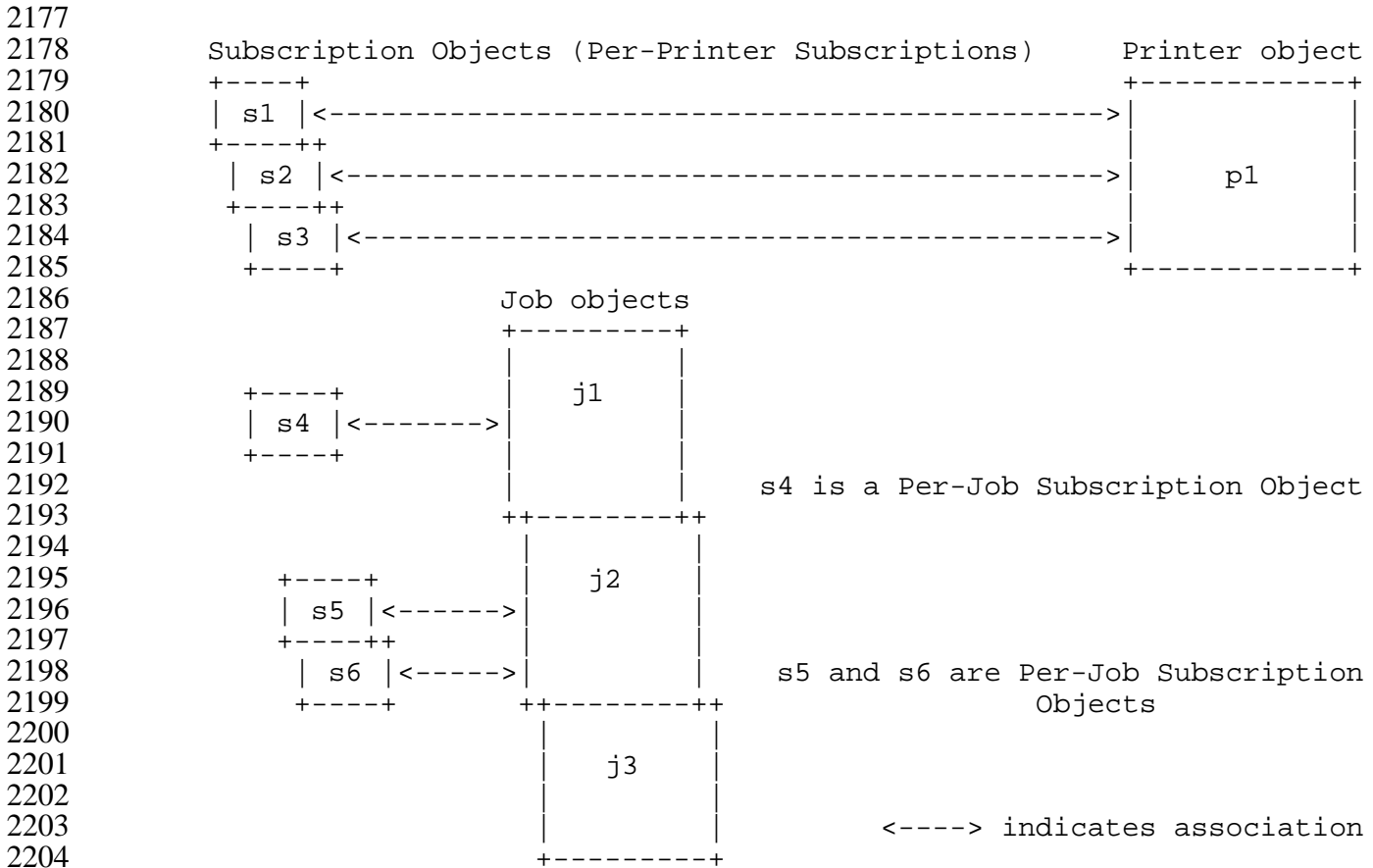
2164 **RECOMMENDED** - an adjective used to indicate that a conforming IPP Printer implementation is
 2165 recommended to support the indicated operation, object, attribute, attribute value, status code, or
 2166 out-of-band value in requests and responses. *Since support of this entire Notification*
 2167 *specification is OPTIONAL for conformance to IPP/1.1, the use of the term RECOMMENDED*
 2168 *in this document means “RECOMMENDED if this OPTIONAL Notification specification is*
 2169 *implemented”.*

2170 **OPTIONAL** - an adjective used to indicate that a conforming IPP Printer implementation **MAY**, but is
 2171 **NOT REQUIRED** to, support the indicated operation, object, attribute, attribute value, status code,
 2172 or out-of-band value in requests and responses.

2173 **20 Appendix E - Object Model for Notification (Normative)**

2174 This section describes the Notification object model that adds a Subscription Object which together
2175 with the Job and Printer object provide the complete Notification semantics.

2176 The object relationships can be seen pictorially as:



2205 **Figure 5 – Object Model for Notification**

2206 s1, s2, and s3 are Per-Printer Subscription Objects and can identify Printer and/or Job Events.
2207 s4, s5, and s6 are Per-Job Subscription Objects and can identify Printer and/or Job Events.

2208 **20.1 Appendix--Object relationships**

2209 This sub-section defines the object relationships between the Printer, Job, and Subscription Objects by
2210 example. Whether Per-Printer Subscription Objects are actually contained in a Printer object or are
2211 just bi-directionally associated with them in some way is IMPLEMENTATION DEPENDENT and is
2212 transparent to the client. Similarly, whether Per-Job Subscription Objects are actually contained in a
2213 Job object or are just bi-directionally associated with them in some way is IMPLEMENTATION
2214 DEPENDENT and is transparent to the client. The object relationships are defined as follows:

20.2 Printer Object and Per-Printer Subscription Objects

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

20.3 Job Object and Per-Job Subscription Objects

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

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

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

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

2248 **22 Normative References**

- 2249 [ipp-get-method]
2250 Herriot, R., and T. Hastings, "Internet Printing Protocol (IPP): The 'ippget' Delivery Method for
2251 Event Notifications", <draft-ietf-ipp-notify-get-087.txt>, [September 10](#)~~June 27~~, 2002.
- 2252 [RFC2119]
2253 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997
- 2254 [RFC2396]
2255 Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic
2256 Syntax", RFC 2396, August 1998.
- 2257 [RFC2717]
2258 R. Petke and I. King, "Registration Procedures for URL Scheme Names", RFC 2717, November
2259 1999.
- 2260 [RFC2910]
2261 Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.1: Encoding and
2262 Transport", RFC 2910, September 2000.
- 2263 [RFC2911]
2264 deBry, R., Hastings, T., Herriot, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.1:
2265 Model and Semantics", RFC 2911, September 2000.
- 2266 [[RFC3381ipp-prog](#)]
2267 Hastings, T., ~~Bergman, R., and H. Lewis, H., and R. Bergman~~, "IPP: Job Progress Attributes",
2268 <draft-ietf-ipp-job-prog-03.txt> ~~work in progress~~[RFC 3381](#), ~~July 17, 2001~~[September 2002](#).

2269 **23 Informative References**

- 2270 [IANA-CON]
2271 Narte, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs",
2272 BCP 26, RFC 2434, October 1998.
- 2273 [ipp-not-req]
2274 deBry, R., Lewis, H., and T. Hastings, "Internet Printing Protocol/1.1: Requirements for IPP
2275 Notifications", <draft-ietf-ipp-not-06.txt>, work in progress, July 17, 2001.
- 2276 [RFC2565]
2277 Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.0: Encoding and
2278 Transport", RFC 2565, April 1999.
- 2279 [RFC2566]
2280 deBry, R., , Hastings, T., Herriot, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.0:
2281 Model and Semantics", RFC 2566, April 1999.

- 2282 [RFC2567]
 2283 Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.
- 2284 [RFC2568]
 2285 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
 2286 RFC 2568, April 1999.
- 2287 [RFC2569]
 2288 Herriot, R., Hastings, T., Jacobs, N., and J. Martin, "Mapping between LPD and IPP Protocols",
 2289 RFC 2569, April 1999.
- 2290 [RFC2616]
 2291 Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee,
 2292 "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
- 2293 [RFC3196]
 2294 Hastings, T., Manros, C., Zehler, P., Kugler, C., and H. Holst, "Internet Printing Protocol/1.1:
 2295 Implementer's Guide", RFC3196, November 2001.

2296 24 IANA Considerations

2297 This section contains the registration information for IANA to add to the various IPP Registries
 2298 according to the procedures defined in RFC 2911 [RFC2911] section 6 to cover the definitions in this
 2299 document. In addition, this section defines how Events and Delivery Methods will be registered when
 2300 they are defined in other documents. [The resulting registrations will be published in the](http://www.iana.org/assignments/ipp-registrations-registry)
 2301 <http://www.iana.org/assignments/ipp-registrations-registry>.

2302 *Note to RFC Editors: Replace RFC NNNN below with the RFC number for this document, so that it*
 2303 *accurately reflects the content of the information for the IANA Registry.*

2304 24.1 Attribute Registrations

2305 The following table lists all the attributes defined in this document. These are to be registered
 2306 according to the procedures in RFC 2911 [RFC2911] section 6.2.

2307	Subscription Template attributes:	Ref.	Section:
2308	notify-attributes (1setOf type2 keyword)	RFC NNNN	5.3.4
2309	notify-attributes-supported (1setOf type2 keyword)		
2310		RFC NNNN	5.3.4.1
2311	notify-charset (charset)	RFC NNNN	5.3.6
2312	notify-events (1setOf type2 keyword)	RFC NNNN	5.3.3
2313	notify-events-default (1setOf type2 keyword)	RFC NNNN	5.3.3.1
2314	notify-events-supported (1setOf type2 keyword)	RFC NNNN	5.3.3.2
2315	notify-lease-duration (integer(0:67108863))	RFC NNNN	5.3.8
2316	notify-lease-duration-default (integer(0:67108863))		
2317		RFC NNNN	5.3.8.1

2318	notify-lease-duration-supported (1setOf (integer(0: 67108863)			
2319	rangeOfInteger(0:67108863)))	RFC	NNNN	5.3.8.2
2320	notify-max-events-supported (integer(2:MAX))	RFC	NNNN	5.3.3.3
2321	notify-natural-language (naturalLanguage)	RFC	NNNN	5.3.7
2322	notify-pull-method (type2 keyword)	RFC	NNNN	5.3.2
2323	notify-pull-method-supported (1setOf type2 keyword)			
2324		RFC	NNNN	5.3.2.1
2325	notify-recipient-uri (uri)	RFC	NNNN	5.3.1
2326	notify-schemes-supported (1setOf uriScheme)	RFC	NNNN	5.3.1.1
2327	notify-time-interval (integer(0:MAX))	RFC	NNNN	5.3.9
2328	notify-user-data (octetString(63))	RFC	NNNN	5.3.5
2329				
2330	Subscription Description Attributes:			
2331	notify-job-id (integer(1:MAX)))	RFC	NNNN	5.4.6
2332	notify-lease-expiration-time (integer(0:MAX)))	RFC	NNNN	5.4.3
2333	notify-printer-up-time (integer(1:MAX)))	RFC	NNNN	5.4.4
2334	notify-printer-uri (uri)	RFC	NNNN	5.4.5
2335	notify-sequence-number (integer (0:MAX)))	RFC	NNNN	5.4.2
2336	notify-subscriber-user-name (name(MAX))	RFC	NNNN	5.4.7
2337	notify-subscription-id (integer (1:MAX))	RFC	NNNN	5.4.1
2338				
2339	Printer Description Attributes:			
2340	printer-state-change-date-time (dateTime)	RFC	NNNN	6.2
2341	printer-state-change-time (integer(1:MAX))	RFC	NNNN	6.1
2342				
2343	Attributes Only in Event Notifications			
2344	notify-subscribed-event (type2 keyword)	RFC	NNNN	8.1
2345	notify-text (text(MAX))	RFC	NNNN	8.2
2346				

The resulting attribute registrations will be published in the <ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attributes/> area.

24.2 Additional Enum Attribute Value Registrations for the “operations-supported” Printer Attribute

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

Attribute	Value	Name	Reference	Section
	-----	-----	-----	-----
operations-supported (type2 enum)			RFC2911	4.4.15
0x0016		Create-Printer-Subscriptions	RFC NNNN	7.1
0x0017		Create-Job-Subscriptions	RFC NNNN	7.1
0x0018		Get-Subscription-Attributes	RFC NNNN	7.1
0x0019		Get-Subscriptions	RFC NNNN	7.1

2364	<u>0x001A</u>	Renew-Subscription	RFC NNNN	7.1
2365	<u>0x001B</u>	Cancel-Subscription	RFC NNNN	7.1

2366

2367 ~~The resulting enum attribute value registrations will be published in the~~2368 ~~<ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/operations-supported/>~~2369 ~~area.~~

2370

2371

24.3 Operation Registrations

2372 The following table lists all of the operations defined in this document. These are to be registered
2373 according to the procedures in RFC 2911 [RFC2911] section 6.4.

2374	Operations:	Ref.	Section:
2375	Cancel-Subscription Operation	RFC NNNN	11.2.7
2376	Create-Job-Subscriptions Operation	RFC NNNN	11.1.1
2377	Create-Printer-Subscriptions Operation	RFC NNNN	11.1.2
2378	Get-Printer-Attributes - Extensions	RFC NNNN	11.2.3
2379	Get-Subscription-Attributes Operation	RFC NNNN	11.2.4
2380	Get-Subscriptions Operation	RFC NNNN	11.2.5
2381	Job Creation Operations - Extensions	RFC NNNN	11.1.3
2382	Renew-Subscription Operation	RFC NNNN	11.2.6
2383	Validate-Job Operation - Extensions	RFC NNNN	11.2.2

2384

2385 ~~The resulting operation registrations will be published in the~~2386 ~~<ftp://ftp.iana.org/in-notes/iana/assignments/ipp/operations/>~~2387 ~~area.~~

2388

2389

24.4 Status code Registrations

2390 The following table lists all the status codes defined in this document. These are to be registered
2391 according to the procedures in RFC 2911 [RFC2911] section 6.6.

2392	Value	Name	Ref.	Section:
2393	-----	-----	-----	-----
2394	<u>0x0000:0x00FF</u>	<u>- "successful"</u>		
2395	0x0003	successful-ok-ignored-subscriptions	RFC NNNN	12.1
2396				
2397	<u>0x0400:0x04FF</u>	<u>- "client-error"</u>		
2398	0x0414	client-error-ignored-all-subscriptions	RFC NNNN	12.2
2399				
2400	Status Codes in Subscription Attributes Groups:			
2401	0x040C	client error uri scheme not supported	RFC NNNN	13.1
2402	0x040B	client error attributes or values not supported		
2403			RFC NNNN	13.2
2404	0x0415	client error too many subscriptions	RFC NNNN	13.3
2405	0x0005	successful ok too many events	RFC NNNN	13.4
2406	0x0001	successful ok ignored or substituted attributes		

2407 ~~_____ RFC NNNN 13.5~~

2408
2409 ~~The resulting status code registrations will be published in the~~
2410 ~~ftp://ftp.iana.org/in-notes/iana/assignments/ipp/status-codes/~~
2411 ~~area.~~
2412

2413 **24.5 Attribute Group tag Registrations**

2414 The following table lists all the attribute group tags defined in this document. These are to be
2415 registered according to the procedures in RFC 2911 [RFC2911] section 6.5.

<u>Value</u>	<u>Name</u>	<u>Ref.</u>	<u>Section</u>
<u>-----</u>	<u>-----</u>	<u>-----</u>	<u>-----</u>
0x06	subscription-attributes-tag	RFC NNNN	14
0x07	event-notification-attributes-tag	RFC NNNN	14

2420
2421 ~~The resulting attribute group tag registrations will be published in the~~
2422 ~~ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-group-tags/~~
2423 ~~area.~~
2424

2425 **24.6 Registration of Events**

2426 When other document define additional type2 keywords to be used with the “notify-events”
2427 Subscription Template attribute (see section 5.3.3)), these event keywords will be registered according
2428 to the procedures of [RFC2911] section 7.1 as additional attribute values for use with the “notify-
2429 events” Subscription Template attribute, i.e., the "notify-events", "notify-events-default", and "notify-
2430 events-supported" attributes.

2431 Therefore, the IPP Registry entry for an Event will be of the form:

<u>type2 keyword Attribute Values:</u>	<u>Ref.</u>	<u>Section:</u>
<u>Attribute</u>		
<u>Value</u>	<u>Ref.</u>	<u>Section</u>
<u>-----</u>	<u>-----</u>	<u>-----</u>
<u>notify-events (1setOf type2 keyword)</u>		
<u>notify-events-default (1setOf type2 keyword)</u>		
<u>notify-events-supported (1setOf type2 keyword)</u>		
<u><event keyword name></u>	RFC xxxx	m.n

2441 ~~The resulting type2 keyword attribute values will be published in the~~
2442 ~~ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/notify-events/~~
2443 ~~area.~~
2444

2445 **24.7 Registration of Event Notification Delivery Methods**

2446 This section describes the requirements and procedures for registration and publication of Event
2447 Notification Delivery Methods and for the submission of such proposals.

2448 **24.7.1 Requirements for Registration of Event Notification Delivery Methods**

2449 Registered IPP Event Notification Delivery Methods are expected to follow a number of requirements
2450 described below.

2451 **24.7.1.1 Required Characteristics**

2452 A Delivery Method Document MUST either (1) contain all of the semantics of the Delivery Method or
2453 (2) contain the IPP Delivery Method registration requirements and a profile of some other protocol that
2454 in combination is the Delivery Method (e.g., mailto). The Delivery Method Document (and any
2455 documents it requires) MUST define either (1) a URL for a Push Delivery Method that meets the
2456 requirements of [RFC2717]. or (2) a keyword for a Pull Delivery method.

2457
2458 IPP Event Notification Delivery Method Documents MUST meet the requirements of this document
2459 (see sections 9 and 10).

2460 In addition, a Delivery Method Document MUST contain the following information:

2461
2462 Type of registration: IPP Event Notification Delivery Method

2463 Name of this delivery method:

2464 Proposed URL scheme name of this Push Delivery Method or the keyword name of this Pull
2465 Delivery Method:

2466 Name of proposer:

2467 Address of proposer:

2468 Email address of proposer:

2469 Is this delivery method REQUIRED or OPTIONAL for conformance to the IPP Event Notification
2470 and Subscriptions document:

2471 Is this delivery method defining Machine Consumable and/or Human Consumable content:
2472

2473 **24.7.1.2 Naming Requirements**

2474 Exactly one (URL scheme or keyword) name MUST be assigned to each Delivery Method.

2475 Each assigned name MUST uniquely identify a single Delivery Method. All Push Delivery Method
2476 names MUST conform to the rules for URL scheme names, according to [RFC2396] and [RFC2717]
2477 for schemes in the IETF tree. All Pull Delivery Method names MUST conform to the rules for
2478 keywords according to [RFC2911].

2479 **24.7.1.3 Functionality Requirements**

2480 Delivery Methods MUST function as a protocol that is capable of delivering (push or pull) IPP Event
2481 Notifications to Notification Recipients.

2482 **24.7.1.4 Usage and Implementation Requirements**

2483 Use of a large number of Delivery Methods may hamper interoperability. However, the use of a large
2484 number of undocumented and/or unlabelled Delivery Methods hampers interoperability even more.

2485 A Delivery Method should therefore be registered ONLY if it adds significant functionality that is
2486 valuable to a large community, OR if it documents existing practice in a large community. Note that
2487 Delivery Methods registered for the second reason should be explicitly marked as being of limited or
2488 specialized use and should only be used with prior bilateral agreement.

2489 **24.7.1.5 Publication Requirements**

2490 Delivery Method Documents MUST be published in a standards track, informational, or experimental
2491 RFCs.

2492 **24.7.2 Registration Procedure**

2493 The IPP WG is developing a small number of Delivery Methods which are intended to be published as
2494 standards track RFCs. However, some parties may wish to register additional Delivery Methods in the
2495 future. This section describes the procedures for these additional Delivery Methods.

2496 **24.7.2.1 Present the proposal to the Community**

2497 First the Delivery Method Document MUST be an Internet-Draft with a target category of standards
2498 track, informational, or experimental. The same MUST be true for any documents that it references.

2499 [SendDeliver](#) the proposed Delivery Method Document proposal to the “ipp@pwg.org” mailing list.
2500 This mailing list has been established by [RFC2911] for reviewing proposed registrations and
2501 discussing other IPP matters. Proposed Delivery Method Documents are not formally registered and
2502 MUST NOT be used until approved.

2503 The intent of the public posting is to solicit comments and feedback on the definition and suitability of
2504 the Delivery Method and the name chosen for it over a four week period.

2505 **24.7.2.2 Delivery Method Reviewer**

2506 The Delivery Method Reviewer is the same person who has been appointed by the IETF Application
2507 Area Director(s) as the IPP Designated Expert according to [RFC2911] and [IANA-CON]. When the

2508 four week period is over and the IPP Designated Expert is convinced that consensus has been achieved,
 2509 the IPP Designated Expert either approves the request for registration or rejects it. Rejection may
 2510 occur because of significant objections raised on the list or objections raised externally.

2511 Decisions made by the Reviewer must be posted to the ipp@pwg.org mailing list within 14 days.
 2512 Decisions made by the Reviewer may be appealed to the IESG.

2513 **24.7.2.3 IANA Registration**

2514 Provided that the Delivery Method registration proposal has either passed review or has been
 2515 successfully appealed to the IESG, the IANA will register the Delivery Method and make it available to
 2516 the community.

2517 **24.7.3 Delivery Method Document Registrations**

2518 Each Push Delivery Method Document defines a URI scheme which is registered as an additional value
 2519 of the “notify-schemes-supported” Printer attribute. These uriScheme values will be registered
 2520 according to the procedures of [RFC2911] section 7.1 for additional attribute values. Therefore, the
 2521 IPP Registry entry for a Push Delivery Method will be of the form:

uriScheme Attribute Values:	Ref.	Section:
<u>Attribute</u>	<u>Ref.</u>	<u>Section</u>
<u>Value</u>	<u>Ref.</u>	<u>Section</u>
-----	-----	-----
<u>notify-schemes-supported (type2 keyword)</u>	<u>RFC xxxx</u>	<u>5.3.1.1</u>
<u><scheme name></u>	<u>RFC xxxx</u>	<u>m.n</u>

2522 ~~The resulting Delivery Method URI schemes will be published in the~~
 2523 ~~<ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/notify-schemes-supported/>~~
 2524 ~~area.~~

2525 Each Pull Delivery Method Document defines a keyword method which is registered as an additional
 2526 value of the “notify-pull-method” and “notify-pull-method-supported” Printer attributes. These
 2527 keyword values will be registered according to the procedures of [RFC2911] section 7.1 for additional
 2528 attribute values. Therefore, the IPP Registry entry for a Pull Delivery Method will be of the form:

keyword Attribute Values:	Ref.	Section:
<u>Attribute</u>	<u>Ref.</u>	<u>Section</u>
<u>Value</u>	<u>Ref.</u>	<u>Section</u>
-----	-----	-----
<u>notify-pull-method (type2 keyword)</u>	<u>[ipp-ntfy]</u>	<u>5.3.2</u>
<u>notify-pull-method-supported (1setOf type2 keyword)</u>	<u>[ipp-ntfy]</u>	<u>5.3.2.1</u>
<u><method <u>keyword</u> name></u>	<u>RFC xxxx</u>	<u>m.n</u>

2546 ~~The resulting Delivery Method URI schemes will be published in the~~
2547 ~~ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/notify-pull-method-supported/~~
2548 ~~area.~~
2549

2550 **24.7.4 Registration Template**

2551 To: ipp@pwg.org
2552 Subject: Registration of a new Delivery Method

2553
2554 Delivery Method name:

2555
2556 (All Push Delivery Method names must be suitable for use as the value of a URL scheme in the IETF
2557 tree and all Pull Delivery Method names must be suitable IPP keywords according to [RFC2911])

2558
2559 Published specification(s):

2560
2561 (A specification for the Delivery Method must be openly available that accurately describes what is
2562 being registered.)

2563
2564 Person & email address to contact for further information:

2565 **25 Internationalization Considerations**

2566 This IPP Notification specification continues support for the internationalization of [RFC2911] of
2567 attributes containing text strings and names. Allowing a Subscribing Client to specify a different
2568 natural language and charset for each Subscription Object increases the internationalization support.

2569 The Printer MUST be able to localize the content of Human Consumable Event Notifications and to
2570 localize the value of “notify-text” attribute in Machine Consumable Event Notifications that it
2571 ~~send~~deliver to Notification Recipients. For localization, the Printer MUST use the value of the
2572 “notify-charset” attribute and the “notify-natural-language” attribute in the Subscription Object
2573 supplied by the Subscribing Client.

2574 **26 Security Considerations**

2575 Clients submitting Notification requests to the IPP Printer have the same security issues as submitting
2576 an IPP/1.1 print job request (see [RFC2911] section 3.2.1 and section 8). The same mechanisms used
2577 by IPP/1.1 can therefore be used by the client Notification submission. Operations that require
2578 authentication can use the HTTP authentication. Operations that require privacy can use the
2579 HTTP/TLS privacy. As with IPP/1.1 Print Job Objects, if there is no security on Subscription Objects,
2580 sequential assignment of subscription-ids exposes the system to a passive traffic monitoring threat.

2581 **26.1 Client access rights**

2582 The Subscription Object access control model is the same as the access control model for Job objects.
2583 The client MUST have the following access rights for the indicated Subscription operations:

- 2584 1. Create-Job-Subscriptions (see section 11.1.1): A Per-Job Subscription object is associated with
2585 a Job. To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section
2586 8.3) performing this operation MUST (1) be the job owner, (2) have Operator or Administrator
2587 access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized
2588 by the Printer's administrator-configured security policy to create Per-Job Subscription Objects
2589 for the target job.
- 2590 2. Create-Printer-Subscriptions (see section 11.1.2): A Per-Printer Subscription object is
2591 associated with the Printer. To create Per-Printer Subscription Objects, the authenticated user
2592 (see [RFC2911] section 8.3) performing this operation MUST (1) have Operator or
2593 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5) or (2) be
2594 otherwise authorized by the Printer's administrator-configured security policy to create Per-
2595 Printer Subscription Objects for this Printer.
- 2596 3. Get-Subscription-Attributes (see section 11.2.4): The access control model for this operation is
2597 the same as that of the Get-Job-Attributes operation (see [RFC2911] section 3.3.4). The
2598 primary difference is that a Get-Subscription-Attributes operation is directed at a Subscription
2599 Object rather than at a Job object, and a returned attribute group contains Subscription Object
2600 attributes rather than Job object attributes. To query the specified Subscription Object, the
2601 authenticated user (see [RFC2911] section 8.3) performing this operation MUST (1) be the
2602 Subscription Object owner, (2) have Operator or Administrator access rights for this Printer
2603 (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer's
2604 administrator-configured security policy to query the Subscription Object for the target job.
2605 Furthermore, the Printer's security policy MAY limit which attributes are returned, in a manner
2606 similar to the Get-Job-Attributes operation (see [RFC2911] end of section 3.3.4.2).
- 2607 4. Get-Subscriptions (see section 11.2.5): The access control model for this operation is the same
2608 as that of the Get-Jobs operation (see [RFC2911] section 3.2.6). The primary difference is that
2609 the operation is directed at Subscription Objects rather than at Job objects, and the returned
2610 attribute groups contain Subscription Object attributes rather than Job object attributes. To
2611 query Per-Job Subscription Objects of the specified job (client supplied the "notify-job-id"
2612 operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)
2613 performing this operation MUST (1) be the Subscription Object owner, (2) have Operator or
2614 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be
2615 otherwise authorized by the Printer's administrator-configured security policy to query the
2616 Subscription Object for the target job. To query Per-Printer Subscription Objects of the Printer
2617 (client omits the "notify-job-id" operation attribute - see section 11.2.5.1.1), the authenticated
2618 user (see [RFC2911] section 8.3) performing this operation MUST (1) have Operator or
2619 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (2) be
2620 otherwise authorized by the Printer's administrator-configured security policy to query Per-
2621 Printer Subscription Objects for the target Printer. Furthermore, the Printer's security policy

2622 MAY limit which attributes are returned, in a manner similar to the Get-Job-Attributes
2623 operation (see [RFC2911] end of section 3.2.6.2).

2624 5. Renew-Subscriptions (see section 11.2.6): The authenticated user (see [RFC2911] section 8.3)
2625 performing this operation MUST (1) be the owner of the Per-Printer Subscription Object, (2)
2626 have Operator or Administrator access rights for the Printer (see [RFC2911] sections 1 and
2627 8.5), or (3) be otherwise authorized by the Printer's administrator-configured security policy to
2628 renew Per-Printer Subscription Objects for the target Printer

2629 6. Cancel-Subscription (see section 11.2.7): The authenticated user (see [RFC2911] section 8.3)
2630 performing this operation MUST (1) be the owner of the Subscription Object, (2) have Operator
2631 or Administrator access rights for the Printer (see [RFC2911] sections 1 and 8.5), or (3) be
2632 otherwise authorized by the Printer's administrator-configured security policy to cancel the
2633 target Subscription Object.

2634 The standard security concerns (delivery to the right user, privacy of content, tamper proof content)
2635 apply to each Delivery Method. Some Delivery Methods are more secure than others. Each Delivery
2636 Method Document MUST discuss its Security Considerations.

2637 26.2 Printer security threats

2638 Notification trap door: If a Printer supports the OPTIONAL "notify-attributes" Subscription Template
2639 attribute (see section 5.3.4) where the client can request that the Printer return any specified Job,
2640 Printer, and Subscription object attributes, the Printer MUST apply the same security policy to these
2641 requested attributes in the Get-Notifications request as it does for the Get-Jobs, Get-Job-Attributes,
2642 Get-Printer-Attributes, and Get-Subscription-Attributes requests.

2643 26.3 Notification Recipient security threats

2644 Unwanted Events Notifications (spam): For any Push Delivery Method, by far the biggest security
2645 concern is the abuse of notification: ~~send~~delivering unwanted Event Notifications to third parties (i.e.,
2646 spam). The problem is made worse by notification addresses that may be redistributed to multiple
2647 parties. There exist scenarios where third party notification is used (see Scenario #2 and #3 in [ipp-
2648 not-req]). Any fully secure solution would require active agreement of all recipients before ~~send~~
2649 ~~out~~delivering anything.

2650 27 Contributors

2651 The following people made significant contributions to the design and review of this specification:

2652 Scott A. Isaacson
2653 Novell, Inc.
2654 122 E 1700 S
2655 Provo, UT 84606

2656

2657 Phone: 801-861-7366

2658 Fax: 801-861-2517

2659 e-mail: sisaacson@novell.com

2660

2661 Roger deBry

2662 Utah Valley State College

2663 Orem, UT 84058

2664

2665 Phone: (801) 222-8000

2666 EMail: debryro@uvsc.edu

2667

2668 Jay Martin

2669 Underscore Inc.

2670 9 Jacqueline St.

2671 Hudson, NH 03051-5308

2672 603-889-7000

2673 fax: 775-414-0245

2674 e-mail: jkm@underscore.com

2675

2676 Michael Shepherd

2677 Xerox Corporation

2678 800 Phillips Road MS 128-51E

2679 Webster, NY 14450

2680

2681 Phone: 716-422-2338

2682 Fax: 716-265-8871

2683 e-mail: mshepherd@crt.xerox.com

2684

2685 Ron Bergman

2686 Hitachi Koki Imaging Solutions

2687 1757 Tapo Canyon Road

2688 Simi Valley, CA 93063-3394

2689

2690 Phone: 805-578-4421

2691 Fax: 805-578-4001

2692 Email: rbergma@hitachi-hkis.com

2693 **28 Author's Addresses**

2694 Robert Herriot

2695 706 Colorado Ave.

2696 Palo Alto, CA 94303

2697

2698 Phone: 650-327-4466

2699 Fax: 650-327-4466
2700 Email: bob@herriot.com
2701
2702 Tom Hastings
2703 Xerox Corporation
2704 737 Hawaii St. ESAE 231
2705 El Segundo, CA 90245
2706
2707 Phone: 310-333-6413
2708 Fax: 310-333-5514
2709 e-mail: hastings@cp10.es.xerox.com
2710
2711 IPP Web Page: <http://www.pwg.org/ipp/>
2712 IPP Mailing List: ipp@pwg.org
2713
2714 To subscribe to the ipp mailing list, send the following email:
2715 1) send it to majordomo@pwg.org
2716 2) leave the subject line blank
2717 3) put the following two lines in the message body:
2718 subscribe ipp
2719 end

2720
2721 Implementers of this specification document are encouraged to join the IPP Mailing List in order to
2722 participate in any discussions of clarification issues and review of registration proposals for additional
2723 attributes and values. In order to reduce spam the mailing list rejects mail from non-subscribers, so
2724 you must subscribe to the mailing list in order to send a question or comment to the mailing list.

2725 **29 Appendix G - Description of the base IPP documents (Informative)**

2726 The base set of IPP documents includes:

2727 Design Goals for an Internet Printing Protocol [RFC2567]
2728 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
2729 Internet Printing Protocol/1.1: Model and Semantics [RFC2911]
2730 Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
2731 Internet Printing Protocol/1.1: Implementer's Guide [RFC3196]
2732 Mapping between LPD and IPP Protocols [RFC2569]
2733

2734 The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed
2735 printing functionality, and it enumerates real-life scenarios that help to clarify the features that need to
2736 be included in a printing protocol for the Internet. It identifies requirements for three types of users:
2737 end users, operators, and administrators. It calls out a subset of end user requirements that are satisfied
2738 in IPP/1.0 [RFC2566, RFC2565]. A few OPTIONAL operator operations have been added to IPP/1.1
2739 [RFC2911, RFC2910].

2740 The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document
2741 describes IPP from a high level view, defines a roadmap for the various documents that form the suite
2742 of IPP specification documents, and gives background and rationale for the IETF IPP working group's
2743 major decisions.

2744 The "Internet Printing Protocol/1.1: Model and Semantics" document describes a simplified model
2745 with abstract objects, their attributes, and their operations. The model introduces a Printer and a Job.
2746 The Job supports multiple documents per Job. The model document also addresses how security,
2747 internationalization, and directory issues are addressed.

2748 The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the
2749 abstract operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It also
2750 defines the encoding rules for a new Internet MIME media type called "application/ipp". This
2751 document also defines the rules for transporting over HTTP a message body whose Content-Type is
2752 "application/ipp". This document defines the 'ipp' scheme for identifying IPP printers and jobs.

2753 The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to
2754 implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some
2755 of the considerations that may assist them in the design of their client and/or IPP object
2756 implementations. For example, a typical order of processing requests is given, including error
2757 checking. Motivation for some of the specification decisions is also included.

2758 The "Mapping between LPD and IPP Protocols" document gives some advice to implementers of
2759 gateways between IPP and LPD (Line Printer Daemon) implementations.

2760 **30 Appendix H - Full Copyright Statement (Informative)**

2761 Copyright (C) The Internet Society (1998,1999,2000,2001,2002). All Rights Reserved

2762 This document and translations of it may be copied and furnished to others, and derivative works that
2763 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published
2764 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright
2765 notice and this paragraph are included on all such copies and derivative works. However, this
2766 document itself may not be modified in any way, such as by removing the copyright notice or
2767 references to the Internet Society or other Internet organizations, except as needed for the purpose of
2768 developing Internet standards in which case the procedures for copyrights defined in the Internet
2769 Standards process must be followed, or as required to translate it into languages other than English.

2770 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or
2771 its successors or assigns.

2772 This document and the information contained herein is provided on an "AS IS" basis and THE
2773 INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL
2774 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
2775 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY

2776 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
2777 PARTICULAR PURPOSE.

2778 **Acknowledgement**

2779
2780 Funding for the RFC Editor function is currently provided by the Internet Society.