

1 Internet Printing Protocol WG  
2 INTERNET-DRAFT  
3 <draft-ietf-ipp-not-spec-110.txt>  
4 Updates RFC 2910 and 2911  
5 [Target Category: standards track]  
6 Expires: ~~April 10~~August 17, 2003  
7  
8

R. Herriot  
consultant  
T. Hastings  
Xerox Corporation  
~~February 17, 2003~~October 10, 2002

9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32

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

Copyright (C) The Internet Society (2003~~2~~). All Rights Reserved.

12 Status of this Memo

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

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

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

22 **Abstract**

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

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

33 **Table of Contents**

34	1 Introduction .....	7
35	1.1 Notification Overview .....	7
36	2 Models for Notification .....	10
37	2.1 Model for Simple Notification (Normative).....	10
38	2.2 Additional Models for Notification (Informative) .....	10
39	3 Terminology .....	10
40	3.1 Conformance Terminology .....	10
41	3.2 Other Terminology .....	11
42	4 Object Relationships.....	13
43	4.1 Printer and Per-Printer Subscription Objects .....	13
44	4.2 Printer, Job and Per-Job Subscription Objects .....	13
45	5 Subscription Object .....	13
46	5.1 Rules for Support of Subscription Template Attributes.....	14
47	5.2 Rules for Processing Subscription Template Attributes.....	15
48	5.3 Subscription Template Attributes.....	18
49	5.3.1 notify-recipient-uri (uri) .....	19
50	5.3.1.1 notify-schemes-supported (1setOf uriScheme).....	20
51	5.3.2 notify-pull-method (type2 keyword) .....	20
52	5.3.2.1 notify-pull-method-supported (1setOf type2 keyword) .....	20
53	5.3.3 notify-events (1setOf type2 keyword).....	20
54	5.3.3.1 notify-events-default (1setOf type2 keyword) .....	21
55	5.3.3.2 notify-events-supported (1setOf type2 keyword).....	21
56	5.3.3.3 notify-max-events-supported (integer(2:MAX)).....	21
57	5.3.3.4 Standard Values for Subscribed Events .....	21
58	5.3.3.4.1 No Events .....	22
59	5.3.3.4.2 Subscribed Printer Events .....	22
60	5.3.3.4.3 Subscribed Job Events .....	23
61	5.3.3.5 Rules for Matching of Subscribed Events.....	24
62	5.3.3.5.1 Rules for Matching of Printer Events.....	24
63	5.3.3.5.2 Rules for Matching of Job Events .....	25
64	5.3.3.5.3 Special Cases for Matching Rules.....	25
65	5.3.4 notify-attributes (1setOf type2 keyword) .....	26
66	5.3.4.1 notify-attributes-supported (1setOf type2 keyword).....	27
67	5.3.5 notify-user-data (octetString(63)).....	27
68	5.3.6 notify-charset (charset).....	28
69	5.3.7 notify-natural-language (naturalLanguage).....	28
70	5.3.8 notify-lease-duration (integer(0:67108863)).....	28
71	5.3.8.1 notify-lease-duration-default (integer(0:67108863)) .....	29
72	5.3.8.2 notify-lease-duration-supported (1setOf (integer(0: 67108863)   rangeOfInteger(0:67108863)))	
73	29	

74	5.3.9 notify-time-interval (integer(0:MAX)).....	30
75	5.4 Subscription Description Attributes .....	30
76	5.4.1 notify-subscription-id (integer (1:MAX)) .....	31
77	5.4.2 notify-sequence-number (integer (0:MAX)) .....	31
78	5.4.3 notify-lease-expiration-time (integer(0:MAX)) .....	32
79	5.4.4 notify-printer-up-time (integer(1:MAX)).....	33
80	5.4.5 notify-printer-uri (uri).....	33
81	5.4.6 notify-job-id (integer(1:MAX)).....	33
82	5.4.7 notify-subscriber-user-name (name(MAX)) .....	34
83	6 Printer Description Attributes Related to Notification.....	34
84	6.1 printer-state-change-time (integer(1:MAX)).....	34
85	6.2 printer-state-change-date-time (dateTime).....	35
86	7 New Values for Existing Printer Description Attributes.....	35
87	7.1 operations-supported (1setOf type2 enum) .....	35
88	8 Attributes Only in Event Notifications.....	35
89	8.1 notify-subscribed-event (type2 keyword) .....	35
90	8.2 notify-text (text(MAX)) .....	36
91	9 Event Notification Content.....	36
92	9.1 Content of Machine Consumable Event Notifications.....	38
93	9.1.1 Event Notification Content Common to All Events.....	39
94	9.1.2 Additional Event Notification Content for Job Events .....	40
95	9.1.3 Additional Event Notification Content for Printer Events .....	40
96	9.2 Content of Human Consumable Event Notification.....	41
97	9.2.1 Event Notification Content Common to All Events.....	41
98	9.2.2 Additional Event Notification Content for Job Events .....	43
99	9.2.3 Additional Event Notification Content for Printer Events .....	43
100	10 Delivery Methods .....	44
101	11 Operations for Notification.....	46
102	11.1 Subscription Creation Operations .....	46
103	11.1.1 Create-Job-Subscriptions Operation.....	46
104	11.1.1.1 Create-Job-Subscriptions Request.....	47
105	11.1.1.1.1 notify-job-id (integer(1:MAX)).....	47
106	11.1.1.2 Create-Job-Subscriptions Response .....	47
107	11.1.2 Create-Printer-Subscriptions operation .....	48
108	11.1.2.1 Create-Printer-Subscriptions Request .....	49
109	11.1.2.2 Create-Printer-Subscriptions Response.....	49
110	11.1.3 Job Creation Operations – Extensions for Notification.....	49
111	11.1.3.1 Job Creation Request.....	50
112	11.1.3.2 Job Creation Response .....	50
113	11.2 Other Operations .....	51
114	11.2.1 Restart-Job Operation – Extensions for Notification .....	51

115	11.2.2 Validate-Job Operation – Extensions for Notification .....	52
116	11.2.3 Get-Printer-Attributes – Extensions for Notification .....	52
117	11.2.4 Get-Subscription-Attributes operation .....	53
118	11.2.4.1 Get-Subscription-Attributes Request .....	53
119	11.2.4.1.1 “notify-subscription-id” (integer (1:MAX)).....	53
120	11.2.4.1.2 “requested-attributes” (1setOf keyword).....	54
121	11.2.4.2 Get-Subscription-Attributes Response .....	54
122	11.2.5 Get-Subscriptions operation .....	55
123	11.2.5.1 Get-Subscriptions Request .....	56
124	11.2.5.1.1 “notify-job-id” (integer(1:MAX)) .....	56
125	11.2.5.1.2 “limit” (integer(1:MAX)) .....	56
126	11.2.5.1.3 “requested-attributes” (1setOf type2 keyword).....	56
127	11.2.5.1.4 “my-subscriptions” (boolean).....	57
128	11.2.5.2 Get-Subscriptions Response .....	57
129	11.2.6 Renew-Subscription operation .....	58
130	11.2.6.1 Renew-Subscription Request.....	58
131	11.2.6.1.1 “notify-subscription-id” (integer (1:MAX)).....	58
132	11.2.6.1.2 “notify-lease-duration” (integer(0:MAX)) .....	59
133	11.2.6.2 Renew-Subscription Response .....	59
134	11.2.6.2.1 “notify-lease-duration” (integer(0:MAX)) .....	60
135	11.2.7 Cancel-Subscription operation .....	60
136	11.2.7.1 Cancel-Subscription Request .....	60
137	11.2.7.1.1 “notify-subscription-id” (integer (1:MAX)).....	61
138	11.2.7.2 Cancel-Subscription Response .....	61
139	12 Status Codes .....	61
140	12.1 successful-ok-ignored-subscriptions (0x0003).....	62
141	12.2 client-error-ignored-all-subscriptions (0x0414).....	62
142	13 Status Codes in Subscription Attributes Groups .....	62
143	13.1 client-error-uri-scheme-not-supported (0x040C) .....	62
144	13.2 client-error-attributes-or-values-not-supported (0x040B).....	62
145	13.3 client-error-too-many-subscriptions (0x0415) .....	63
146	13.4 successful-ok-too-many-events (0x0005) .....	63
147	13.5 successful-ok-ignored-or-substituted-attributes (0x0001) .....	63
148	14 Encodings of Additional Attribute Tags .....	63
149	15 Conformance Requirements .....	63
150	15.1 Conformance requirements for clients .....	64
151	15.2 Conformance requirements for Printers .....	64
152	16 Appendix A - Model for Notification with Cascading Printers (Informative).....	65
153	17 Appendix B - Distributed Model for Notification (Informative) .....	66
154	18 Appendix C - Extended Notification Recipient (Informative).....	67

155	19 Appendix D - Details about Conformance Terminology (Normative) .....	68
156	20 Appendix E - Object Model for Notification (Normative).....	68
157	20.1 Object relationships.....	69
158	20.2 Printer Object and Per-Printer Subscription Objects.....	69
159	20.3 Job Object and Per-Job Subscription Objects .....	70
160	21 Appendix F - Per-Job versus Per-Printer Subscription Objects (Normative) .....	70
161	22 Normative References .....	70
162	23 Informative References .....	71
163	24 IANA Considerations .....	72
164	24.1 Attribute Registrations .....	72
165	24.2 Additional Enum Attribute Value Registrations .....	73
166	24.3 Operation Registrations.....	73
167	24.4 Status code Registrations.....	74
168	24.5 Attribute Group tag Registrations .....	74
169	24.6 Registration of Events .....	75
170	24.7 Registration of Event Notification Delivery Methods .....	75
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.....	76
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 .....	78
183	<u>25 Intellectual Property .....</u>	<u>79</u>
184	26 Internationalization Considerations.....	79
185	27 Security Considerations.....	79
186	27.1 Client access rights .....	80
187	27.2 Printer security threats.....	81
188	27.3 Notification Recipient security threats.....	81
189	28 Contributors.....	81
190	29 Author's Addresses .....	82

191	30 Appendix G - Description of the base IPP documents (Informative) .....	83
192	31 Appendix H - Full Copyright Statement (Informative).....	84
193		
194	<b>Tables</b>	
195	Table 1 – Subscription Template Attributes.....	19
196	Table 2 – Subscription Description Attributes.....	31
197	Table 3 – Printer Description Attributes Associated with Notification .....	34
198	Table 4 – Operation-id assignments.....	35
199	Table 5 – Attributes in Event Notification Content .....	39
200	Table 6 – Additional Event Notification Content for Job Events .....	40
201	Table 7 – Combinations of Events and Subscribed Events for “job-impressions-completed” .....	40
202	Table 8 – Additional Event Notification Content for Printer Events.....	41
203	Table 9 – Printer Name in Event Notification Content.....	42
204	Table 10 – Event Name in Event Notification Content.....	42
205	Table 11 – Event Time in Event Notification Content.....	42
206	Table 12 – Job Name in Event Notification Content .....	43
207	Table 13 – Job State in Event Notification Content.....	43
208	Table 14 – Printer State in Event Notification Content.....	44
209	Table 15 – Information about the Delivery Method.....	45
210	Table 16 – Printer Conformance Requirements for Operations.....	65
211		
212	<b>Figures</b>	
213	Figure 1 – Model for Notification .....	10
214	Figure 2 – Model for Notification with Cascading Printers.....	66
215	Figure 3 – Opaque Use of a Notification Server Transparent to the Client .....	67
216	Figure 4 – Use of an Extended Notification Recipient transparent to the Printer.....	68
217	Figure 5 – Object Model for Notification .....	69
218		

## 219 1 Introduction

220 This IPP notification specification is an OPTIONAL extension to Internet Printing Protocol/1.1: Model  
221 and Semantics [RFC2911, RFC2910]. See Appendix 30 for a description of the base IPP documents.  
222 This document in combination with the following documents is intended to meet the most important  
223 notification requirements described in [ipp-not-req]:

224 Internet Printing Protocol (IPP): "Job Progress Attributes" [RFC3381]

225 Internet Printing Protocol (IPP): "The 'ippget' Delivery Method for Event Notifications" [ipp-  
226 get-method]  
227

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

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

### 234 1.1 Notification Overview

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

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

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

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

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

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

275 For each of the above operations:

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

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

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



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

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

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

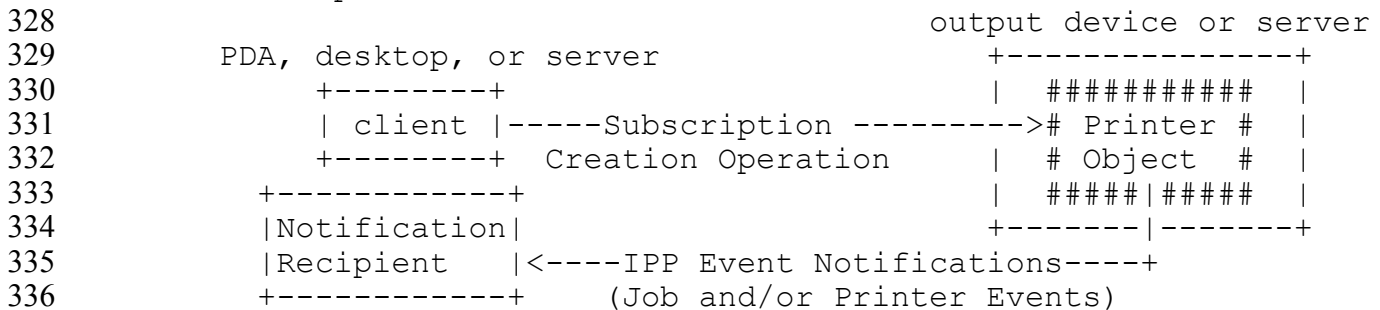
## 320 2 Models for Notification

### 321 2.1 Model for Simple Notification (Normative)

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

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

327 embedded printer:



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 3 Terminology

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

### 343 3.1 Conformance Terminology

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

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

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

### 354   **3.2 Other Terminology**

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

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

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

363       **Delivery Method Document** – a document, separate from this document, that defines a Delivery  
364       Method.

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

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

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

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

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

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

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

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

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

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

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

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

## 426   **4 Object Relationships**

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

### 429   **4.1 Printer and Per-Printer Subscription Objects**

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

### 432   **4.2 Printer, Job and Per-Job Subscription Objects**

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

## 437   **5 Subscription Object**

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

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

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

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

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

## 452 **5.1 Rules for Support of Subscription Template Attributes**

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

456 Subscription Objects attributes that are Subscription Template Attributes conform to the following  
457 rules:

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

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

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

## 490   **5.2 Rules for Processing Subscription Template Attributes**

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

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

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

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

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

- 514 4. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 and the attribute is  
515 REQUIRED for the client to supply, the Printer MUST reject the Subscription Creation Operation  
516 (including Job Creation operations) without creating a Subscription Object, and MUST return in  
517 the response:
- 518 c) the status code ‘client-error-bad-request’ AND
- 519 d) no Subscription Attribute Groups.
- 520 5. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 that is OPTIONAL for  
521 the client to supply, and column 2 of Table 1 either:
- 522 a) specifies a “notify-xxx-default” attribute, the Printer MUST behave as if the client had supplied  
523 the “notify-xxx-default” attribute (see step #1) and populate the Subscription object with the  
524 value of the “notify-xxx-default” attribute as part of the Subscription Creation operation (unlike  
525 Job Template attributes where the Printer does not populate the Job object with defaults – see  
526 [RFC2911]) OR
- 527 b) does not specify a “notify-xxx-default” attribute, the Printer MUST populate the “notify-xxx”  
528 attribute on the Subscription Object according to the definition of the “notify-xxx” attribute in a  
529 section 5.3. For some attributes, the “notify-xxx” is populated with the value of some other  
530 attribute, and for others, the “notify-xxx” is NOT populated on the Subscription object at all.
- 531 6. A Printer MUST create a Subscription Object for each Subscription Template Attributes group in a  
532 request unless the Printer:
- 533 a) encounters some attributes in a Subscription Template Attributes Group that require the Printer  
534 not to create the Subscription Object OR
- 535 b) would create a Per-Job Subscription Object when it doesn’t have space for another Per-Job  
536 Subscription Object OR
- 537 c) would create a Per-Printer Subscription Object when it doesn’t have space for another Per-  
538 Printer Subscription Object.
- 539 7. A response MUST contain one Subscription Attributes Group for each Subscription Template  
540 Attributes Group in the request (and in the same order) whether the Printer creates a Subscription  
541 Object from the Subscription Template Attributes Group or not. However, the attributes in each  
542 Subscription Attributes Group can be in any order.
- 543 8. The Printer MUST populate each Subscription Attributes Group of the response such that each  
544 contains:
- 545 a) the “notify-subscription-id” attribute (see section 5.4.1), if and only if the Printer creates a  
546 Subscription Object.
- 547 b) the “notify-lease-duration” attribute (see section 5.3.8), if and only if the Printer creates a Per-  
548 Printer Subscription Object. The value of this attribute is the value of the Subscription Object’s



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

### 590 **5.3 Subscription Template Attributes**

591 This section contains the Subscription Template Attributes defined for the Subscription and Printer  
592 objects.

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

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

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

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

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

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

617

**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))	

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

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

620 **5.3.1 notify-recipient-uri (uri)**

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

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

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

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

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

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

**637 5.3.1.1 notify-schemes-supported (1setOf uriScheme)**

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

**641 5.3.2 notify-pull-method (type2 keyword)**

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

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

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

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

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

**653 5.3.2.1 notify-pull-method-supported (1setOf type2 keyword)**

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

**655 5.3.3 notify-events (1setOf type2 keyword)**

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

659 A Printer MUST support this attribute.

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

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

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

#### 671 **5.3.3.1 notify-events-default (1setOf type2 keyword)**

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

#### 673 **5.3.3.2 notify-events-supported (1setOf type2 keyword)**

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

#### 675 **5.3.3.3 notify-max-events-supported (integer(2:MAX))**

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

#### 679 **5.3.3.4 Standard Values for Subscribed Events**

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

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

685 A Printer MUST support the Events indicated as “REQUIRED” and MAY support the Events  
686 indicated as “OPTIONAL”.

687 **5.3.3.4.1 No Events**

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

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

696 **5.3.3.4.2 Subscribed Printer Events**

697 The standard keyword values for Subscribed Printer Events are:

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

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

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

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

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

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

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

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

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

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

#### 736   **5.3.3.4.3 Subscribed Job Events**

737   The standard keyword values for Subscribed Job Events are:

738           **‘job-state-changed’**: REQUIRED – the job has changed from any state to any other state.  
739           Specifically, the Printer delivers this Event whenever the value of the “job-state” attribute or “job-  
740           state-reasons” attribute changes. When a Job is removed from the Job Retention or Job History  
741           phases (see [RFC2911] section 4.3.7.1), no Event is generated.  
742

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

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

750           **‘job-completed’**: REQUIRED – the job has reached one of the completed states, i.e., the  
751           value of the job’s “job-state” attribute has changed to: ‘completed’, ‘aborted’, or  
752           ‘canceled’. The Job’s “time-at-completed” and “date-time-at-completed” (if supported)  
753           attributes are set (see [RFC2911] section 4.3.14). When a Job completes, a Notification  
754           Recipient MAY query the Job using the Get-Job-Attributes operation. To allow such a  
755           query, the Printer retains the Job in the Job Retention and/or the Job History phases (see  
756           [RFC2911] section 4.3.7.1) for a suitable amount of time that depends on  
757           implementation and the Delivery Methods supported. The Printer also delivers this  
758           Event when a Job is removed with the Purge-Job operation (see [RFC2911] section  
759           3.2.9). In this case, the Event Notification MUST report the ‘job-state’ as ‘canceled’  
760           and the Job object is no longer present for query.

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

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

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

### 775   **5.3.3.5 Rules for Matching of Subscribed Events**

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

#### 779   **5.3.3.5.1 Rules for Matching of Printer Events**

780           Given that the Printer causes Printer Event E to occur, for each Per-Job or Per-Printer Subscription S  
781           in the Printer, if E equals a value of this attribute in S or E is a sub-value of a value of this attribute in  
782           S, the Printer MUST generate an Event Notification.

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



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

### 791 5.3.3.5.2 Rules for Matching of Job Events

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

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

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

### 812 5.3.3.5.3 Special Cases for Matching Rules

813 This section contains rule for special cases.

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

820 If an Event matches two values of this “notify-events” attribute in a single Subscription object (e.g., a  
821 value and its sub-value), a Printer MAY deliver one Event Notification for each matched value in the  
822 Subscription Object or it MAY deliver only one Event Notification per Subscription Object. The rules

823 in sections 5.3.3.5.1 and 5.3.3.5.2 are purposefully flexible about the number of Event Notifications  
824 sent when Event E matches two or more values in a Subscription Object.

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

#### 836 **5.3.4 notify-attributes (1setOf type2 keyword)**

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

841 A Printer MAY support this attribute.

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

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

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

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

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

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

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

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

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

#### 869 **5.3.4.1 notify-attributes-supported (1setOf type2 keyword)**

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

#### 871 **5.3.5 notify-user-data (octetString(63))**

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

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

880 A Printer MUST support this attribute.

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

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

**888 5.3.6 notify-charset (charset)**

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

891 A Printer MUST support this attribute.

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

899 The value of this attribute on a Subscription Object MUST be a value of the “charset-supported  
900 (1setOf charset)” Printer attribute.

**901 5.3.7 notify-natural-language (naturalLanguage)**

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

905 A Printer MUST support this attribute.

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

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

**916 5.3.8 notify-lease-duration (integer(0:67108863))**

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

921 This attribute is not present on a Per-Job Subscription Object because the Subscription Object lasts  
922 exactly as long as the associated Job object. See discussion of the 'job-completed' event in section  
923 5.3.3.4.3 about retention of the Job object after completion.

924 A Printer MUST support this attribute.

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

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

935 After the Printer has populated this attribute with a supported value, the value represents the "granted  
936 duration" of the lease in seconds and the Printer updates the value of the Subscription Object's "notify-  
937 lease-expiration-time" attribute as specified in section 5.4.3.

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

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

943 Note: The maximum value 67,108,863 is 2 raised to the 26 power minus 1 and is about 2 years in  
944 seconds. The value is considerably less than MAX so that there is virtually no chance of an overflow  
945 when the Printer adds it to the Printer's "printer-up-time" attribute value (see [RFC2911] section  
946 4.4.29) to produce the "notify-lease-expiration-time" Subscription Description attribute value (see  
947 section 5.4.3).

#### 948 **5.3.8.1 notify-lease-duration-default (integer(0:67108863))**

949 See sections 5.1 and 5.2 for the behavior of "xxx-default" Subscription Template Printer attributes.

#### 950 **5.3.8.2 notify-lease-duration-supported (1setOf (integer(0: 67108863) |** 951 **rangeOfInteger(0:67108863)))**

952 See sections 5.1 and 5.2 for the behavior of "xxx-supported" Subscription Template Printer attributes.

### 953 **5.3.9 notify-time-interval (integer(0:MAX))**

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

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

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

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

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

967 1. This attribute is not present on the Subscription Object or has the value of 0. The Printer MUST  
968 generate and deliver an Event Notification (as is the case with other Events).

969 2. This attribute is present with a nonzero value of N:

970 a) If the Printer has not sent an Event Notification for the 'job-progress' Event for the associated  
971 Subscription Object within the past N seconds, the Printer MUST deliver an Event Notification  
972 for the Event that just occurred. Note when the Printer completes the first page of a Job, this  
973 rule implies that the Printer delivers an Event Notification for a Per-Job Subscription Object.

974 b) Otherwise, the Printer MUST NOT generate or deliver an Event Notification for the associated  
975 Subscription Object. The Printer MUST NOT increase the value of the "notify-sequence-  
976 number" Subscription Object attribute (i.e., the sequence of values of the "notify-sequence-  
977 number" attribute counts the Event Notifications that the Printer sent and not the Events that do  
978 not cause an Event Notification to be sent).

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

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

### 983 **5.4 Subscription Description Attributes**

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

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

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

989 **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))

990

#### 991 **5.4.1 notify-subscription-id (integer (1:MAX))**

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

994 A Printer MUST support this attribute.

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

998 The Printer SHOULD avoid re-using recent values of this attribute during continuous operation of the  
999 Printer as well as across power cycles. Then a Subscribing Client is unlikely to find that a stale  
1000 reference accesses a new Subscription Object.

1001 The 0 value is not permitted in order to allow for compatibility with “job-id” and with SNMP index  
1002 values, which also cannot be 0.

#### 1003 **5.4.2 notify-sequence-number (integer (0:MAX))**

1004 The value of this attribute indicates the number of times that the Printer has generated and attempted to  
1005 deliver an Event Notification for this Subscription object. When an Event Notification contains this  
1006 attribute, the Notification Recipient can determine whether it missed some Event Notifications (i.e.,  
1007 numbers skipped) or received duplicates (i.e., same number twice).

1008 A Printer MUST support this attribute.

1009 When the Printer creates a Subscription Object, it MUST populate this attribute with a value of 0. This  
1010 value indicates that the Printer has not sent any Event Notifications for this Subscription Object.

1011 Each time the Printer delivers a newly generated Event Notification, it MUST increase the value of  
1012 this attribute by 1. For some Delivery Methods, the Printer MUST include this attribute in each Event  
1013 Notification, and the value MUST be the value after it is increased by 1. That is, the value of this  
1014 attribute in the first Event Notification after Subscription object creation MUST be 1, the second  
1015 MUST be 2, etc. If a Delivery Method is defined such that the Notification Recipient returns a  
1016 response, the Printer can re-try delivering an Event Notification a certain number of times with the  
1017 same sequence number when the Notification Recipient fails to return a response.

1018 If a Subscription Object lasts long enough to reach the value of MAX, its next value MUST be 0, i.e.,  
1019 it wraps.

#### 1020 **5.4.3 notify-lease-expiration-time (integer(0:MAX))**

1021 This attribute specifies the time in the future when the lease on the Per-Printer Subscription Object will  
1022 expire, i.e. the “printer-up-time” value at which the lease will expire. If the value is 0, the lease never  
1023 expires.

1024 A Printer MUST support this attribute.

1025 When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present – the  
1026 Subscription Object lasts exactly as long as the associated Job object. See also the discussion of the  
1027 ‘job-completed’ event in section 5.3.3.4.3 about retention of the Job object after completion so that a  
1028 Notification Recipient can query the Job object after receiving the ‘job-completed’ Event Notification.

1029 When the Printer creates a Per-Printer Subscription Object, it populates this attribute with a value that  
1030 is the sum of the values of the Printer’s “printer-up-time” attribute and the Subscription Object’s  
1031 “notify-lease-duration” attribute with the following exception. If the value of the Subscription  
1032 Object’s “notify-lease-duration” attribute is 0 (i.e., no expiration time), then the value of this attribute  
1033 MUST be set to 0 (i.e., no expiration time).

1034 When the Printer powers up, it MUST populate this attribute in each persistent Subscription Object  
1035 with a value using the algorithm in the previous paragraph.

1036 When the “printer-up-time” equals the value of this attribute, the Printer MUST delete the Subscription  
1037 Object. A client can extend a lease of a Per-Printer Subscription Object with the Renew-Subscription  
1038 operation (see section 11.2.6).

1039 Note: In order to compute the number of seconds remaining in a lease for a Per-Printer Subscription  
1040 Object, a client can subtract the Subscription’s “notify-printer-up-time” attribute (see section 5.4.4)  
1041 from the Subscription’s “notify-lease-expiration-time” attribute.



**1042 5.4.4 notify-printer-up-time (integer(1:MAX))**

1043 This attribute is an alias for the Printer's "printer-up-time" attribute " (see [RFC2911] section 4.4.29).  
1044 In other words, when this attribute is queried with the Get-Subscriptions or Get-Subscription-  
1045 Attributes operations (see sections 11.2.4 and 11.2.5), the value returned is the current value of the  
1046 Printer's "printer-up-time" attribute, rather than the time at which the Subscription Object was created.

1047 A Printer MUST support this attribute.

1048 When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present. When  
1049 the Printer creates a Per-Printer Subscription Object, this attribute MUST be present.

1050 Note: this attribute exists in a Per-Printer Subscription Object so that a client using the Get-  
1051 Subscription-Attributes or Get-Subscription operations can convert the Per-Printer Subscription's  
1052 "notify-lease-expiration-time" attribute to wall clock time with one request. If the value of the "notify-  
1053 lease-expiration-time" attribute is not 0 (i.e., no expiration time), then the difference between the  
1054 "notify-lease-expiration-time" attribute and the "notify-printer-up-time" is the remaining number of  
1055 seconds on the lease from the current time.

**1056 5.4.5 notify-printer-uri (uri)**

1057 This attribute identifies the Printer object that created this Subscription Object.

1058 A Printer MUST support this attribute.

1059 During a Subscription Creation Operation, the Printer MUST populate this attribute with the value of  
1060 the "printer-uri" operation attribute in the request. From the Printer URI, the client can, for example,  
1061 determine what security scheme was used.

**1062 5.4.6 notify-job-id (integer(1:MAX))**

1063 This attribute specifies whether the containing Subscription Object is a Per-Job or Per-Printer  
1064 Subscription Object, and for Per-Job Subscription Objects, it specifies the associated Job.

1065 A Printer MUST support this attribute.

1066 If this attribute is not present, the Subscription Object MUST be a Per-Printer Subscription. If this  
1067 attribute is present, the Subscription Object MUST be a Per-Job Subscription Object and this attribute  
1068 MUST identify the Job with which the Subscription Object is associated.

1069 Note: This attribute could be useful to a Notification Recipient that receives an Event Notification  
1070 generated from a Per-Job Subscription Object and caused by a Printer Event. The Event Notification  
1071 gives access to the Printer and the Subscription Object. The Event Notification gives access to the  
1072 associated Job only via this attribute. See discussion of the 'job-completed' event in section 5.3.3.4.3  
1073 about retention of the Job object after completion so that a Notification Recipient can query the Job  
1074 object after receiving the 'job-completed' Event Notification.

### 1075 **5.4.7 notify-subscriber-user-name (name(MAX))**

1076 This attribute contains the name of the user who performed the Subscription Creation Operation.

1077 A Printer **MUST** support this attribute.

1078 The Printer **MUST** populate this attribute with the most authenticated printable name that it can  
 1079 obtain from the authentication service over which the Subscription Creation Operation was received.  
 1080 The Printer uses the same mechanism for determining the value of this attribute as it does for a Job's  
 1081 "job-originating-user-name" (see [RFC2911] section 4.3.6).

1082 Note: To help with authentication, a Subscription Object may have additional private attributes about  
 1083 the user, e.g., a credential of a principal. Such private attributes are implementation-dependent and not  
 1084 defined in this document.

## 1085 **6 Printer Description Attributes Related to Notification**

1086 This section defines the Printer Description attributes that are related to Notification. Table 3 lists the  
 1087 Printer Description attributes, indicates the Printer support required for conformance, and whether or  
 1088 not the attribute is READ-ONLY (see section 3.1):

1089 **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

1090

### 1091 **6.1 printer-state-change-time (integer(1:MAX))**

1092 This **OPTIONAL** attribute records the most recent time at which the 'printer-state-changed' Printer  
 1093 Event occurred whether or not any Subscription objects were listening for this event. This attribute  
 1094 helps a client or operator to determine how long the Printer has been in its current state.

1095 A Printer **MAY** support this attribute and if so, the attribute **MUST** be READ-ONLY.

1096 On power-up, the Printer **MUST** populate this attribute with the value of its "printer-up-time" attribute,  
 1097 so that it always has a value. Whenever the 'printer-state-changed' Printer Event occurs, the Printer  
 1098 **MUST** update this attribute with the value of the Printer's "printer-up-time" attribute.

## 1099 **6.2 printer-state-change-date-time (dateTime)**

1100 This OPTIONAL attribute records the most recent time at which the ‘printer-state-changed’ Printer  
1101 Event occurred whether or not there were any Subscription Objects listening for this event. This  
1102 attribute helps a client or operator to determine how long the Printer has been in its current state.

1103 A Printer MAY support this attribute and if so, the attribute MUST be READ-ONLY.

1104 On power-up, the Printer MUST populate this attribute with the value of its “printer-current-time”  
1105 attribute, so that it always has a value (see [RFC2911] section 4.4.30 on “printer-current-time”).  
1106 Whenever the ‘printer-state-changed’ Printer Event occurs, the Printer MUST update this attribute  
1107 with the value of the Printer’s “printer-current-time” attribute.

## 1108 **7 New Values for Existing Printer Description Attributes**

1109 This section contains those attributes for which additional values are added.

### 1110 **7.1 operations-supported (1setOf type2 enum)**

1111 The following “operation-id” values are added in order to support the new operations defined in this  
1112 document:

1113 **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

## 1114 **8 Attributes Only in Event Notifications**

1115 This section contains those attributes that exist only in Event Notifications and do not exist in any  
1116 objects.

### 1117 **8.1 notify-subscribed-event (type2 keyword)**

1118 This attribute indicates the Subscribed Event that caused the Printer to deliver this Event Notification.  
1119 This attribute exists only in Event Notifications.

1120 This attribute MUST contain one of the values of the “notify-events” attribute in the Subscription  
1121 Object, i.e., one of the Subscribed Event values. Its value is the Subscribed Event that “matches” the

1122 Event that caused the Printer to deliver this Event Notification. This Subscribed Event value may be  
1123 identical to the Event or the Event may be a sub-value of the Subscribed Event. For example, the ‘job-  
1124 completed’ Event (which is a sub-event of the ‘job-state-changed’ event) would cause the Printer to  
1125 deliver an Event Notification for either the ‘job-completed’ or ‘job-state-changed’ Subscribed Events  
1126 and to deliver the ‘job-completed’ or ‘job-state-changed’ value for this attribute, respectively. See  
1127 section 5.3.3.5 for the “matching” rules of Subscribed Events and for additional examples.

1128 The Delivery Method Document specifies whether the Printer includes the value of this attribute in an  
1129 Event Notification.

## 1130 **8.2 notify-text (text(MAX))**

1131 This attribute contains a Human Consumable text message (see section 9.2). This message describes  
1132 the Event and is encoded as plain text, i.e., ‘text/plain’ with the charset specified by Subscription  
1133 Object’s “notify-charset” attribute.

1134 The Delivery Method Document specifies whether the Printer includes this attribute in an Event  
1135 Notification.

## 1136 **9 Event Notification Content**

1137 This section defines the Event Notification content that the Printer delivers when an Event occurs.

1138 When an Event occurs, the Printer MUST find each Subscription object whose “notify-events”  
1139 attribute “matches” the Event. See section 5.3.3.5 for details on “matching”. For each matched  
1140 Subscription Object, the Printer MUST create an Event Notification with the content and format that  
1141 the Delivery Method Document specifies. The content contains the value of attributes specified by the  
1142 Delivery Method Document. The Printer obtains the values immediately after the Event occurs. For  
1143 example, if the “printer-state” attribute changes from ‘idle’ to ‘processing’, the Event ‘printer-state-  
1144 changed’ occurs and the Printer puts various attributes into the Event Notification, including “printer-  
1145 up-time” and “printer-state” with the values that they have immediately after the Event occurs, i.e., the  
1146 value of “printer-state” is ‘processing’.

### 1147 **Event Notification Ordering:**

1148 When a Printer delivers Event Notifications, the Event Notifications from any given Subscription  
1149 Object MUST be in time stamp order, i.e., in order of increasing “printer-up-time” attribute value in  
1150 the Event Notification (see Table 5). These Event Notifications MAY be interleaved with those from  
1151 other Subscription Objects, as long as those others are also in time stamp order. The Printer MUST  
1152 observe these ordering requirements whether delivering multiple pending Events as multiple separate  
1153 Event Notifications or together in a single Compound Event Notification.

1154 If a Subscribing Client wants the Printer to deliver certain Event Notifications in time stamp order, the  
1155 Subscribing Client uses a single Subscription Object. Even so, depending on the underlying transport,

1156 the actual order that a Notification Recipient receives separate Event Notifications may differ from the  
1157 order sent by the Printer (e.g., email).

1158 Example: Consider two Per-Printer Subscription Objects: SO1 and SO2. SO1 requests 'job-state-  
1159 changed' events and SO2 requests 'printer-state-changed' events. The number in parens is the time  
1160 stamp. The following Event Notification sequences are the only ones that conform to the ordering  
1161 requirements for the Printer to deliver the Event Notifications:

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

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

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

1168 (d) SO2: 'printer-stopped' (1005), SO1: 'job-created' (1000), SO1: 'job-stopped' (1005), SO1: 'job-  
1169 completed' (1009)

1170 Examples (b) and (c) are interleaved; examples (a) and (d) are not interleaved and are not appropriate  
1171 for some Delivery Methods.

1172 If two different Events occur simultaneously, or nearly so (e.g., "printer-up-time" has the same value  
1173 for both), the Printer MUST create a separate Event Notification for each Event, even if the associated  
1174 Subscription Object is the same for both Events. However, the Printer MAY combine these distinct  
1175 Event Notifications into a single Compound Event Notification if the Delivery Method supports  
1176 Compound Event Notifications. For example, suppose that two nearly-simultaneously Events  
1177 represent two successive 'printer-state-changed' Events, one from 'idle' to 'processing' and another  
1178 from 'processing' to 'stopped'. These two Events have the same name but are different instances of  
1179 the Event. Then the Printer MUST create a separate Event Notification for each Event and SHOULD  
1180 accurately report the "printer-state" of the first Event as 'processing' and the second Event as  
1181 'stopped'.

1182 If a Subscription Object contains more than one Subscribed Event, and several Events occur in quick  
1183 succession each matching a different Subscribed Event in the Subscription Object, the Printer MUST  
1184 NOT generate a single Event Notification from several of these Events, but MAY combine distinct  
1185 Event Notifications into a single Compound Event Notification if the Delivery Method supports  
1186 Compound Event Notifications.

1187 After the Printer has created the Event Notification, the Printer delivers it via either a:

1188 Push Delivery Method: The Printer delivers the Event Notification shortly after an Event  
1189 occurs. For some Push Delivery Methods, the Notification Recipient MUST deliver a response;  
1190 for others it MUST NOT deliver a response.

1191 Pull Delivery Method: The Printer saves Event Notifications for some Event Life and expects  
1192 the Notification Recipient to request Event Notifications. The Printer returns the Event  
1193 Notifications in a response to such a request.

1194 If an error that meets the following conditions occurs, the Printer **MUST** cancel the Subscription  
1195 Object.

1196 a) the error occurs during the delivering of an Event Notification generated from Subscription Object  
1197 S **AND**

1198 b) the error would continue to occur every time the Printer delivers an Event Notification generated  
1199 from Subscription Object S in the future.

1200 For example, if the address of the “notify-recipient-uri” of Subscription Object A references a non-  
1201 existent target and the Printer determines this fact, it **MUST** delete Subscription Object A.

1202 The next two sections describe the values that a Printer delivers in the content of Machine Consumable  
1203 and Human Consumable Event Notifications, respectively.

1204 The tables in the sub-sections of this section contain the following columns:

1205 a) **Source Value:** the name of the attribute that supplies the value for the Event Notification.  
1206 Asterisks in this field refer to a note below the table.

1207 b) **Delivers:** if the Printer supports the value (column 1) on the Source Object (column 3) the  
1208 Delivery Method **MUST** specify:

1209 **MUST:** that the Printer **MUST** deliver the value.

1210 **SHOULD:** either that the Printer **MUST** deliver the value or that the value is incompatible  
1211 with the Delivery Method.

1212 **MAY:** that the Printer **MUST**, **SHOULD**, **MAY**, **MUST NOT**, **SHOULD NOT**, or **NEED**  
1213 **NOT** deliver the value. The Delivery Method specifies the level of conformance for the  
1214 Printer.

1215 c) **Source Object:** the object from which the source value comes. If the object is “Event  
1216 Notification”, the Printer fabricates the value when it delivers the Event Notification. See  
1217 section 8.

## 1218 9.1 Content of Machine Consumable Event Notifications

1219 This section defines the attributes that a Delivery Method **MUST** mention in a Delivery Method  
1220 Document when specifying the Machine Consumable Event Notification’s contents.

1221 This document does not define the order of attributes in Event Notifications. However, Delivery  
1222 Method Documents **MAY** define the order of some or all of the attributes.

1223 A Delivery Method Document MUST specify additional attributes (if any) that a Printer  
1224 implementation delivers in a Machine Consumable Event Notification.

1225 Notification Recipients MUST be able to accept Event Notifications containing attributes they do not  
1226 recognize. What a Notification Recipient does with an unrecognized attribute is implementation-  
1227 dependent. Notification Recipients MAY attempt to display unrecognized attributes anyway or MAY  
1228 ignore them.

1229 The next three sections define the attributes in Event Notification Contents that are:

1230 1. for all Events

1231 2. for Job Events only

1232 3. for Printer Events only

### 1233 9.1.1 Event Notification Content Common to All Events

1234 This section lists the attributes that a Delivery Method Document MUST specify for all Events.

1235 Table 5 lists potential values in each Event Notification.

1236 **Table 5 – Attributes in Event Notification Content**

Source Value	Delivers	Source Object
notify-subscription-id (integer(1:MAX))	MUST	Subscription
notify-printer-uri (uri)	MUST	Subscription
notify-subscribed-event (type2 keyword)	MUST	Event Notification
printer-up-time (integer(MIN:MAX))	MUST	Printer
printer-current-time (dateTime) *	MUST	Printer
notify-sequence-number (integer(0:MAX))	SHOULD	Subscription
notify-charset (charset)	SHOULD	Subscription
notify-natural-language (naturalLanguage)	SHOULD	Subscription
notify-user-data (octetString(63)) **	SHOULD	Subscription
notify-text (text)	SHOULD	Event Notification
attributes from the “notify-attributes” attribute ***	MAY	Printer
attributes from the “notify-attributes” attribute ***	MAY	Job
attributes from the “notify-attributes” attribute ***	MAY	Subscription

1237  
1238 \*A Printer MUST deliver this value only if and only if it supports the Printer’s “printer-current-time”  
1239 attribute.

1240 \*\* If the Subscription Object does not contain a “notify-user-data” attribute and the Delivery Method  
1241 Document REQUIRES the Printer to deliver the “notify-user-data” source value in the Event  
1242 Notification, the Printer MUST deliver an octet-string of length 0.

1243 \*\*\* The last three rows represent additional attributes that a client MAY request via the “notify-  
 1244 attributes” attribute. A Printer MAY support the “notify-attributes” attribute. The Delivery Method  
 1245 MUST say that the Printer MUST, SHOULD, MAY, MUST NOT, SHOULD NOT, or NEED NOT  
 1246 support the “notify-attributes” attribute and specific values of this attribute. The Delivery Method  
 1247 MAY say that support for the “notify-attributes” is conditioned on support of the attribute by the  
 1248 Printer or it MAY say that Printer MUST support the “notify-attributes” attribute if the Printer  
 1249 supports the Delivery Method.

### 1250 9.1.2 Additional Event Notification Content for Job Events

1251 This section lists the additional attributes that a Delivery Method Document MUST specify for Job  
 1252 Events. See Table 6.

1253 **Table 6 – Additional Event Notification Content for Job Events**

Source Value	Delivers	Source Object
job-id (integer(1:MAX))	MUST	Job
job-state (type1 enum)	MUST	Job
job-state-reasons (1setOf type2 keyword)	MUST	Job
job-impressions-completed (integer(0:MAX)) *	MUST	Job

1254 \* The Printer MUST deliver the “job-impressions-completed” attribute in an Event Notification only  
 1255 for the combinations of Events and Subscribed Events shown in Table 7.  
 1256

1257 **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’

1258

### 1259 9.1.3 Additional Event Notification Content for Printer Events

1260 This section lists the additional attributes that a Delivery Method Document MUST specify for Printer  
 1261 Events. See Table 8.



1262

**Table 8 – Additional Event Notification Content for Printer Events**

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

1263

## 1264 9.2 Content of Human Consumable Event Notification

1265 This section defines the information that a Delivery Method MUST mention in a Delivery Method  
 1266 Document when specifying the Human Consumable Event Notifications contents or the value of the  
 1267 “notify-text” attribute.

1268 Such a Delivery Method MUST specify the following information and a Printer SHOULD deliver it:

- 1269 a) the Printer name (see Table 9)
- 1270 b) the time of the Event (see Table 11)
- 1271 c) for Printer Events only:
  - 1272 i) the Event (see Table 10) and/or Printer state information (see Table 14)
- 1273 d) for Job Events only:
  - 1274 i) the job identity (see Table 12)
  - 1275 ii) the Event (see Table 10) and/or Job state information (see Table 13)

1276

1277 The subsections of this section specify the attributes that a Printer MUST use to obtain this  
 1278 information.

1279 A Delivery Method Document MUST specify additional information (if any) that a Printer  
 1280 implementation delivers in a Human Consumable Event Notification or in the “notify-text” attribute.

1281 A client MUST NOT request additional attributes via the “notify-attributes” attribute because this  
 1282 attribute works only for Machine Consumable Event Notifications.

1283 Notification Recipients MUST NOT expect to be able to parse the Human Consumable Event  
 1284 Notification contents or the value of the “notify-text” attribute.

1285 The next three sections define the attributes in Event Notification Contents that are:

- 1286 a) for all Events
- 1287 b) for Job Events only
- 1288 c) for Printer Events only

1289

### 1290 9.2.1 Event Notification Content Common to All Events

1291 This section lists the source of the information that a Delivery Method MUST specify for all Events.

1292 There is a separate table for each piece of information. Each row in the table represents a source value  
 1293 for the information and the values are listed in order of preference, with the first one being the  
 1294 preferred one. An implementation SHOULD use the source value from the earliest row in each table.  
 1295 It MAY use the source value from another row instead, or it MAY combine the source values from  
 1296 several rows. An implementation is free to determine the best way to present this information.

1297 In all tables of this section, all rows contain a “MAY” in order to state that the Delivery Method  
 1298 specifies the conformance.

1299 Table 9 lists the source of the information for the Printer Name. The “printer-name” is more user-  
 1300 friendly unless the Notification Recipient is in a place where the Printer name is not meaningful. For  
 1301 example, an implementation could have the intelligence to deliver the value of the “printer-name”  
 1302 attribute to a Notification Recipient that can access the Printer via value of the “printer-name” attribute  
 1303 and otherwise deliver the value of the “notify-printer-uri” attribute.

1304 **Table 9 – Printer Name in Event Notification Content**

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

1305  
 1306 Table 10 lists the source of the information for the Event name. A Printer MAY combine this  
 1307 information with state information described for Jobs in Table 13 or for Printers in Table 14.

1308 **Table 10 – Event Name in Event Notification Content**

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

1309  
 1310 Table 11 lists the source of the information for the time that the Event occurred. A Printer can deliver  
 1311 this value only if it supports the Printer’s “printer-current-time” attribute. If a Printer does not support  
 1312 the “printer-current-time” attribute, it MUST NOT deliver the “printer-up-time” value instead, since it  
 1313 is not an allowed option for human consumable information.

1314 **Table 11 – Event Time in Event Notification Content**

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

1315

## 1316 9.2.2 Additional Event Notification Content for Job Events

1317 This section lists the source of the additional information that a Delivery Method MUST specify for  
1318 Job Events.

1319 Table 12 lists the source of the information for the job name. The “job-name” is likely more  
1320 meaningful to a user than “job-id”.

1321 **Table 12 – Job Name in Event Notification Content**

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

1322

1323 Table 13 lists the source of the information for the job state. If a Printer supports the “job-state-  
1324 message” and “job-detailed-state-message” attributes, it SHOULD use those attributes for the job state  
1325 information, otherwise, it should fabricate such information from the “job-state” and “job-state-  
1326 reasons”. For some Events, a Printer MAY combine this information with Event information.

1327 **Table 13 – Job State in Event Notification Content**

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

1328

## 1329 9.2.3 Additional Event Notification Content for Printer Events

1330 This section lists the source of the additional information that a Delivery Method MUST specify for  
1331 Printer Events.

1332 Table 14 lists the source of the information for the printer state. If a Printer supports the “printer-state-  
1333 message”, it SHOULD use that attribute for the job state information, otherwise it SHOULD fabricate  
1334 such information from the “printer-state” and “printer-state-reasons”. For some Events, a Printer  
1335 MAY combine this information with Event information.

1336

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

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

1337 **10 Delivery Methods**

1338 A Delivery Method is the mechanism, i.e., protocol, by which the Printer delivers an Event  
 1339 Notification to a Notification Recipient. There are several potential Delivery Methods for Event  
 1340 Notifications, standardized, as well as proprietary. This specification REQUIRES that the ‘ippget’  
 1341 Pull Delivery Method [ipp-get-method] be supported. Conforming implementations MAY support  
 1342 additional Push or Pull Delivery Methods as well. This document does not define any of these  
 1343 delivery mechanisms. Each Delivery Method MUST be defined in a Delivery Method Document that  
 1344 is separate from this document. New Delivery Methods will be created as needed using an extension  
 1345 to the registration procedures defined in [RFC2911]. Such documents are registered with IANA (see  
 1346 section 24.7.3).

1347 The following sorts of Delivery Methods are possible:

- 1348 – The Notification Recipient polls for Event Notifications at intervals directed by the Printer
- 1349 – The Printer delivers Event Notifications to the Notification Recipient using http as the transport.
- 1350 – The Printer delivers an email message.

1351 This section specifies how to define a Delivery Method Document and what to put in such a document.

1352 A Delivery Method Document MUST contain an exact copy of the following paragraph, caption and  
 1353 table. In addition, column 2 of the table in the Delivery Method Document MUST contain answers to  
 1354 questions in column 1 for the Delivery Method. Also, the Delivery Method document MUST contain  
 1355 a reference to this document and call that reference [ipp-ntfy] because the table contains an [ipp-ntfy]  
 1356 reference.

1357 If a Printer supports this Delivery Method, the following are its characteristics.

1358 **Table 15 – Information about the Delivery Method**

Document Method Conformance Requirement	Delivery Method Realization
1. What is the URL scheme name for the Push Delivery Method or the keyword method name for the Pull Delivery Method?	
2. Is the Delivery Method REQUIRED, RECOMMENDED, or OPTIONAL for an IPP Printer to support?	
3. What transport and delivery protocols does the Printer use to deliver the Event Notification Content, i.e., what is the entire network stack?	
4. Can several Event Notifications be combined into a Compound Event Notification?	
5. Is the Delivery Method initiated by the Notification Recipient (pull), or by the Printer (push)?	
6. Is the Event Notification content Machine Consumable or Human Consumable?	
7. What section in this document answers the following question? For a Machine Consumable Event Notification, what is the representation and encoding of values defined in section 9.1 of [ipp-ntfy] and the conformance requirements thereof? For a Human Consumable Event Notification, what is the representation and encoding of pieces of information defined in section 9.2 of [ipp-ntfy] and the conformance requirements thereof?	
8. What are the latency and reliability of the transport and delivery protocol?	
9. What are the security aspects of the transport and delivery protocol, e.g., how it is handled in firewalls?	
10. What are the content length restrictions?	
11. What are the additional values or pieces of information that a Printer delivers in an Event Notification content and the conformance requirements thereof?	
12. What are the additional Subscription Template and/or Subscription Description attributes and the conformance requirements thereof?	
13. What are the additional Printer Description attributes and the conformance requirements thereof?	

1359

## 1360 **11 Operations for Notification**

1361 This section defines all of the operations for Notification. Section 7.1 assigns the “operation-id” for  
1362 each operation. The following two sub-sections define Subscription Creation Operations, and other  
1363 operations.

### 1364 **11.1 Subscription Creation Operations**

1365 This section defines the Subscription Creation Operations. The first section on Create-Job-  
1366 Subscriptions gives most of the information. The other Subscription Creation Operations refer to the  
1367 section on Create-Job-Subscriptions, even though the Create-Job-Subscriptions operation is the only  
1368 OPTIONAL operation in this document (see section 12).

1369 A Printer MUST support Create-Printer-Subscriptions and the Subscription Template Attributes Group  
1370 in Job Creation operations. It MAY support Create-Job-Subscriptions operations.

#### 1371 **11.1.1 Create-Job-Subscriptions Operation**

1372 The operation creates one or more Per-Job Subscription Objects. The client supplies one or more  
1373 Subscription Template Attributes Groups each containing one or more of Subscription Template  
1374 Attributes (defined in section 5.3).

1375 Except for errors, the Printer MUST create exactly one Per-Job Subscription Object from each  
1376 Subscription Template Attributes Group in the request, even if the newly created Subscription Object  
1377 would have identical behavior to some existing Subscription Object. The Printer MUST associate  
1378 each newly created Per-Job Subscription Object with the target Job, which is specified by the “notify-  
1379 job-id” operation attribute.

1380 The Printer MUST accept the request in any of the target job’s ‘not-completed’ states, i.e., ‘pending’,  
1381 ‘pending-held’, ‘processing’, or ‘processing-stopped’. The Printer MUST NOT change the job’s “job-  
1382 state” attribute because of this operation. If the target job is in any of the ‘completed’ states, i.e.,  
1383 ‘completed’, ‘canceled’, or ‘aborted’, then the Printer MUST reject the request and return the ‘client-  
1384 error-not-possible’ status code; the response MUST NOT contain any Subscription Attribute Groups.

1385 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section  
1386 8.3) performing this operation MUST (1) be the job owner, (2) have Operator or Administrator access  
1387 rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the  
1388 Printer’s administrator-configured security policy to create Per-Job Subscription Objects for the target  
1389 job. Otherwise the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-  
1390 error-not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

**1391 11.1.1.1 Create-Job-Subscriptions Request**

1392 The following groups of attributes are part of the Create-Job-Subscriptions Request:

1393 Group 1: Operation Attributes

1394 Natural Language and Character Set:

1395 The "attributes-charset" and "attributes-natural-language" attributes as described in  
1396 [RFC2911] section 3.1.4.1.

1397

1398 Target:

1399 The "printer-uri" attribute which defines the target for this operation as described in  
1400 [RFC2911] section 3.1.5.

1401

1402 Requesting User Name:

1403 The "requesting-user-name" attribute SHOULD be supplied by the client as described in  
1404 [RFC2911] section 8.3.

**1405 11.1.1.1.1 notify-job-id (integer(1:MAX))**

1406 The client MUST supply this attribute and it MUST specify the Job object to associate the  
1407 Per-Job Subscription with. The value of "notify-job-id" MUST be the value of the "job-id" of  
1408 the associated Job object. If the client does not supply this attribute, the Printer MUST reject  
1409 this request with a 'client-error-bad-request' status code.

1410

1411 Group 2-N: Subscription Template Attributes

1412 For each occurrence of this group:

1413

1414 The client MUST supply one or more Subscription Template Attributes in any order. See  
1415 section 5.3 for a description of each such attribute. See section 5.2 for details on processing  
1416 these attributes.

**1417 11.1.1.2 Create-Job-Subscriptions Response**

1418 The Printer MUST return to the client the following sets of attributes as part of a Create-Job-  
1419 Subscriptions response:

1420 Group 1: Operation Attributes

1421 Status Message:

1422 In addition to the REQUIRED status code returned in every response, the response  
1423 OPTIONALLY includes a "status-message" (text(255)) and/or a "detailed-status-message"  
1424 (text(MAX)) operation attribute as described in [RFC2911] sections 13 and 3.1.6.

1425

1426 In this group, the Printer can return any status codes defined in [RFC2911] and section 12.  
1427 The following is a description of the important status codes:

- 1428
- 1429 **successful-ok:** the Printer created all Subscription Objects requested (see [RFC2911]).
- 1430 **successful-ok-ignored-subscriptions:** the Printer created some Subscription Objects  
1431 requested but some failed. The Subscription Attributes Groups with a “notify-status-  
1432 code” attribute are the ones that failed (see section 12.1).
- 1433 **client-error-ignored-all-subscriptions:** the Printer created no Subscription Objects  
1434 requested and all failed. The Subscription Attributes Groups with a “notify-status-  
1435 code” attribute are the ones that failed (see section 12.2).
- 1436 **client-error-not-possible:** For this operation and other Per-Job Subscription operations,  
1437 this error can occur because the specified Job has already completed (see  
1438 [RFC2911], whether or not the Job is retained in the Job Retention and/or Job  
1439 History phases (see [RFC2911] section 4.3.7.1).

1440

1441 Natural Language and Character Set:

1442 The “attributes-charset” and “attributes-natural-language” attributes as described in  
1443 [RFC2911] section 3.1.4.2.

1444

1445 Group 2: Unsupported Attributes

1446 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes. This group  
1447 does not contain any unsupported Subscription Template Attributes; they are returned in the  
1448 Subscription Attributes Group (see below).

1449

1450 Group 3-N: Subscription Attributes

1451 These groups MUST be returned unless the Printer is unable to interpret the entire request,  
1452 e.g., the “status-code” parameter returned in Group 1 has the value: ‘client-error-bad-request’.  
1453

1454 “notify-status-code” (type2 enum):  
1455 Indicates the status of this subscription (see section 13 for the status code definitions).  
1456 Section 5.2 defines when this attribute MUST be present in this group.

1457

1458 See section 5.2 for details on the contents of each occurrence of this group.  
1459

### 1460 **11.1.2 Create-Printer-Subscriptions operation**

1461 The operation is identical to Create-Job-Subscriptions with exceptions noted in this section.

1462 The operation creates Per-Printer Subscription Objects instead of Per-Job Subscription Objects, and  
1463 associates each newly created Per-Printer Subscription Object with the Printer specified by the  
1464 operation target rather than with a specific Job.

1465 The Printer MUST accept the request in any of its states, i.e., ‘idle’, ‘processing’, or ‘stopped’. The  
1466 Printer MUST NOT change its “printer-state” attribute because of this operation.



1467 Access Rights: To create Per-Printer Subscription Objects, the authenticated user (see [RFC2911]  
1468 section 8.3) performing this operation MUST have (1) Operator or Administrator access rights for this  
1469 Printer (see [RFC2911] sections 1 and 8.5), or (2) be otherwise authorized by the Printer's  
1470 administrator-configured security policy to create Per-Printer Subscription Objects for this Printer.  
1471 Otherwise, the Printer MUST reject the operation and return: the 'client-error-forbidden', 'client-error-  
1472 not-authenticated', or 'client-error-not-authorized' status code as appropriate.

#### 1473 **11.1.2.1 Create-Printer-Subscriptions Request**

1474 The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.1) except that the  
1475 Operation Attributes group MUST NOT contain the "notify-job-id" attribute. If the client does supply  
1476 the "notify-job-id" attribute, then the Printer MUST treat it as any other unsupported Operation  
1477 attribute and MUST return it in the Unsupported Attributes group.

#### 1478 **11.1.2.2 Create-Printer-Subscriptions Response**

1479 The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.2).

#### 1480 **11.1.3 Job Creation Operations – Extensions for Notification**

1481 This document extends the Job Creation operations (see section 3.2) to create Subscription Objects as a  
1482 part of the operation.

1483 The Job Creation operations are identical to Create-Job-Subscriptions operation with exceptions noted  
1484 in this section.

1485 Unlike the Create-Job-Subscriptions operation, a Job Creation operation associates the newly created  
1486 Subscription Objects with the Job object created by this operation. The operation succeeds if and only  
1487 if the Job creation succeeds. If the Printer does not create some or all of the requested Subscription  
1488 Objects, the Printer MUST return a 'successful-ok-ignored-subscriptions' status-code instead of a  
1489 'successful-ok' status-code, but the Printer MUST NOT reject the operation because of a failure to  
1490 create Subscription Objects.

1491 If the Job Creation operation includes a Job Template group, the client MUST supply it after the  
1492 Operation Attributes group and before the first Subscription Template Attributes Group.

1493 If a Printer does not support this Notification specification, then it MUST treat the Subscription  
1494 Attributes Group like an unknown group and ignore it (see [RFC2911] section 5.2.2). Because the  
1495 Printer ignores the Subscription Attributes Group, it doesn't return them in the response either, thus  
1496 indicating to the client that the Printer doesn't support Notification.

1497 After completion of a successful Job Creation operation, the Printer generates a 'job-created' event  
1498 (see section 5.3.3.4.3).

1499 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section  
1500 8.3) performing this operation MUST either have permission to create Jobs on the Printer or have  
1501 Operator or Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5). Otherwise  
1502 the Printer MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-  
1503 authenticated', or 'client-error-not-authorized' status code as appropriate.

#### 1504 **11.1.3.1 Job Creation Request**

1505 The groups for this operation are sufficiently different from the Create-Job-Subscriptions operation  
1506 that they are all presented here. The following groups of attributes are supplied as part of a Job  
1507 Creation Request:

1508 Group 1: Operation Attributes

1509 Same as defined in [RFC2911] for Print-Job, Print-URI, and Create-Job requests.

1510

1511 Group 2: Job Template Attributes

1512 The client OPTIONALLY supplies a set of Job Template attributes as defined in [RFC2911]  
1513 section 4.2.

1514

1515 Group 3 to N: Subscription Template Attributes

1516 The same as Group 2-N in Create-Job-Subscriptions. See section 11.1.1.1.

1517 Group N+1: Document Content (Print-Job only)

1518 The client MUST supply the document data to be processed.

1519

#### 1520 **11.1.3.2 Job Creation Response**

1521 The Printer MUST return to the client the following sets of attributes as part of a Print-Job, Print-URI,  
1522 and Create-Job Response:

1523 Group 1: Operation Attributes

1524 Status Message:

1525

1526 As defined in [RFC2911] for Print-Job, Print-URI, and Create-Job requests.

1527

1528 In this group, the Printer can return any status codes defined in [RFC2911] and section 12.  
1529 The following is a description of the important status codes:

1530

1531 **successful-ok:** the Printer created the Job and all Subscription Objects requested (see  
1532 [RFC2911]).

1533 **successful-ok-ignored-subscriptions:** the Printer created the Job and not all of the  
1534 Subscription Objects requested (see section 12.1). This status-code hides

1535 'successful-ok-xxx' status-codes that could reveal problems in Job creation. The  
1536 Printer MUST NOT return the 'client-error-ignored-all-subscriptions' status code for  
1537 Job Creation operations because the Printer returns an error status-code only when it  
1538 fails to create a Job.

1539

1540 Natural Language and Character Set:

1541 The "attributes-charset" and "attributes-natural-language" attributes as described in  
1542 [RFC2911] section 3.1.4.2.

1543

1544 Group 2: Unsupported Attributes

1545 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes. This group  
1546 does not contain any unsupported Subscription Template Attributes; they are returned in the  
1547 Subscription Attributes Group (see below).

1548

1549 Group 3: Job Object Attributes

1550 The "job-id" of the Job Object just created, etc., as defined in [RFC2911] for Print-Job, Print-  
1551 URI, and Create-Job requests.

1552

1553 Group 4 to N: Subscription Attributes

1554 These groups MUST be returned if and only if the client supplied Subscription Template  
1555 Attributes and the operation was accepted.

1556

1557 See section 5.2 for details on the contents of each occurrence of this group.

1557

## 1558 11.2 Other Operations

1559 This section defines other operations on Subscription objects.

### 1560 11.2.1 Restart-Job Operation – Extensions for Notification

1561 The Restart-Job operation [RFC2911] is neither a Job Creation operation nor a Subscription Creation  
1562 operation (see section 3.2). For the Restart-Job operation, the client MUST NOT supply any Job  
1563 Subscription Attributes Groups. The Printer MUST treat any supplied Job Subscription Attributes as  
1564 unsupported attributes.

1565 For this operation, the Printer does not return a job-id or any Subscription Attributes groups because  
1566 the Printer reuses the existing Job object with the same job-id and the existing Per-Job Subscription  
1567 Objects with the same subscription-ids. However, after successful completion of this operation, the  
1568 Printer generates a 'job-created' event (see section 5.3.3.4.3).

### 1569 **11.2.2 Validate-Job Operation – Extensions for Notification**

1570 A client can test whether one or more Subscription Objects could be created using the Validate-Job  
1571 operation. The client supplies one or more Subscription Template Attributes Groups (defined in  
1572 section 5.3), just as in a Job Creation request.

1573 A Printer MUST support this extension to this operation.

1574 The Printer MUST accept requests that are identical to the Job Creation request defined in section  
1575 11.1.3.1, except that the request MUST NOT contain document data.

1576 The Printer MUST return the same groups and attributes as the Print-Job operation (section 11.1.3.1)  
1577 with the following exceptions. The Printer MUST NOT return a Job Object Attributes Group because  
1578 no Job is created. The Printer MUST NOT return the “notify-subscription-id” attribute in any  
1579 Subscription Attribute Group because no Subscription Object is created.

1580 If the Printer would succeed in creating a Subscription Object, the corresponding Subscription  
1581 Attributes Group either has no ‘status-code’ attribute or a ‘status-code’ attribute with a value of  
1582 ‘successful-ok-too-many-events’ or ‘successful-ok-ignored-or-substituted-attributes’ (see sections 5.2  
1583 and 13). The status-codes have the same meaning as in Job Creation except the results state what  
1584 “would happen”.

1585 The Printer MUST validate Subscription Template Attributes Groups in the same manner as the Job  
1586 Creation operations.

### 1587 **11.2.3 Get-Printer-Attributes – Extensions for Notification**

1588 This operation is extended so that it returns Printer attributes defined in this document.

1589 A Printer MUST support this extension to this operation.

1590 In addition to the requirements of [RFC2911] section 3.2.5, a Printer MUST support the following  
1591 additional values for the “requested-attributes” Operation attribute in this operation and return such  
1592 attributes in the Printer Object Attributes group of its response.

- 1593 1. **Subscription Template Attributes:** Each supported attribute in column 2 of Table 1.
- 1594 2. **New Printer Description Attributes:** Each supported attribute in section 6.
- 1595 3. **New Group Name:** The ‘subscription-template’ group name, which names all supported  
1596 Subscription Template Attribute in column 2 of Table 1. This group name is also used in the  
1597 Get-Subscription-Attributes and Get-Subscriptions operation with an analogous meaning.
- 1598 4. **Extended Group Name:** The ‘all’ group name, which names all Printer attributes according to  
1599 [RFC2911] section 3.2.5. In this extension ‘all’ names all attributes specified in [RFC2911]  
1600 plus those named in items 1 and 2 of this list.

## 1601 **11.2.4 Get-Subscription-Attributes operation**

1602 This operation allows a client to request the values of the attributes of a Subscription Object.

1603 A Printer **MUST** support this operation.

1604 This operation is almost identical to the Get-Job-Attributes operation (see [RFC2911] section 3.3.4).  
1605 The only differences are that the operation is directed at a Subscription Object rather than a Job object,  
1606 and the returned attribute group contains Subscription Object attributes rather than Job object  
1607 attributes.

1608 Access Rights: The authenticated user (see [RFC2911] section 8.3) performing this operation **MUST**  
1609 (1) be the Subscription Object owner, (2) have Operator or Administrator access rights for this Printer  
1610 (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer's administrator-  
1611 configured security policy to query the Subscription Object for the target job. Otherwise the Printer  
1612 **MUST** reject the operation and return: the 'client-error-forbidden', 'client-error-not-authenticated', or  
1613 'client-error-not-authorized' status code as appropriate. Furthermore, the Printer's security policy  
1614 **MAY** limit which attributes are returned, in a manner similar to the Get-Job-Attributes operation (see  
1615 [RFC2911] end of section 3.3.4.2).

### 1616 **11.2.4.1 Get-Subscription-Attributes Request**

1617 The following groups of attributes are part of the Get-Subscription-Attributes request:

1618 Group 1: Operation Attributes

1619 Natural Language and Character Set:

1620 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1621 [RFC2911] 3.1.4.1.

1622  
1623 Target:

1624 The "printer-uri" attribute which defines the target for this operation as described in  
1625 [RFC2911] section 3.1.5.

1626  
1627 Requesting User Name:

1628 The "requesting-user-name" attribute **SHOULD** be supplied by the client as described in  
1629 [RFC2911] section 8.3.

#### 1630 **11.2.4.1.1 "notify-subscription-id" (integer (1:MAX))**

1631 The client **MUST** supply this attribute. The Printer **MUST** support this attribute. This  
1632 attribute specifies the Subscription Object from which the client is requesting attributes. If  
1633 the client omits this attribute, the Printer **MUST** reject this request with the 'client-error-bad-  
1634 request' status code.

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

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

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

A Printer MUST support all these group names.

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

**11.2.4.2 Get-Subscription-Attributes Response**

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

Group 1: Operation Attributes

Status Message:  
Same as [RFC2911].

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

Group 2: Unsupported Attributes

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

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

## 1675 Group 3: Subscription Attributes

1676 This group contains a set of attributes with their current values. Each attribute returned in this  
1677 group:

- 1678
- 1679 a) MUST be specified by the “requested-attributes” attribute in the request, AND
- 1680 b) MUST be present on the specified Subscription Object AND
- 1681 c) MUST NOT be restricted by the security policy in force. For example, a Printer MAY  
1682 prohibit a client who is not the creator of a Subscription Object from seeing some or all  
1683 of its attributes. See [RFC2911] end of section 3.3.4.2 and section 8.

1684 The Printer can return the attributes of the Subscription Object in any order. The client  
1685 MUST accept the attributes in any order.  
1686

1687 **11.2.5 Get-Subscriptions operation**

1688 This operation allows a client to retrieve the values of attributes of all Subscription Objects belonging  
1689 to a Job or Printer.

1690 A Printer MUST supported this operation.

1691 This operation is similar to the Get-Subscription-Attributes operation, except that this Get-  
1692 Subscriptions operation returns attributes from possibly more than one object.

1693 This operation is similar to the Get-Jobs operation (see [RFC2911] section 3.2.6), except that the  
1694 operation returns Subscription Objects rather than Job objects.

1695 Access Rights: To query Per-Job Subscription Objects of the specified job (client supplied the “notify-  
1696 job-id” operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)  
1697 performing this operation MUST (1) be the Subscription Object owner, (2) have Operator or  
1698 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise  
1699 authorized by the Printer’s administrator-configured security policy to query the Subscription Object  
1700 for the target job. To query Per-Printer Subscription Objects of the Printer (client omits the “notify-  
1701 job-id” operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)  
1702 performing this operation MUST (1) have Operator or Administrator access rights for this Printer (see  
1703 [RFC2911] sections 1 and 8.5), or (2) be otherwise authorized by the Printer’s administrator-  
1704 configured security policy to query Per-Printer Subscription Objects for the target Printer. Otherwise  
1705 the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-not-  
1706 authenticated’, or ‘client-error-not-authorized’ status code as appropriate. Furthermore, the Printer’s  
1707 security policy MAY limit which attributes are returned, in a manner similar to the Get-Jobs and Get-  
1708 Printer-Attributes operations (see [RFC2911] end of sections 3.2.6.2 and 3.2.5.2).

**1709 11.2.5.1 Get-Subscriptions Request**

1710 The following groups of attributes are part of the Get-Subscriptions request:

1711 Group 1: Operation Attributes

1712 Natural Language and Character Set:

1713 The “attributes-charset” and “attributes-natural-language” attributes as described in  
1714 [RFC2911] section 3.1.4.1.

1715

1716 Target:

1717 The “printer-uri” attribute which defines the target for this operation as described in  
1718 [RFC2911] section 3.1.5.

1719

1720 Requesting User Name:

1721 The “requesting-user-name” attribute SHOULD be supplied by the client as described in  
1722 [RFC2911] section 8.3.

**1723 11.2.5.1.1 “notify-job-id” (integer(1:MAX))**

1724 If the client specifies this attribute, the Printer returns the specified attributes of all Per-Job  
1725 Subscription Objects associated with the Job whose “job-id” attribute value equals the value  
1726 of this attribute. If the client does not specify this attribute, the Printer returns the specified  
1727 attributes of all Per-Printer Subscription Objects. Note: there is no way to get all Per-Job  
1728 Subscriptions known to the Printer in a single operation. A Get-Jobs operation followed by a  
1729 Get-Subscriptions operation for each Job will return all Per-Job Subscriptions.

**1730 11.2.5.1.2 “limit” (integer(1:MAX))**

1731 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. It  
1732 is an integer value that determines the maximum number of Subscription Objects that a client  
1733 will receive from the Printer even if the “my-subscriptions” attribute constrains which  
1734 Subscription Objects are returned. The limit is a “stateless limit” in that if the value supplied  
1735 by the client is ‘N’, then only the first ‘N’ Subscription Objects are returned in the Get-  
1736 Subscriptions Response. There is no mechanism to allow for the next ‘M’ Subscription  
1737 Objects after the first ‘N’ Subscription Objects. If the client does not supply this attribute, the  
1738 Printer responds with all applicable Subscription Objects.

**1739 11.2.5.1.3 “requested-attributes” (1setOf type2 keyword)**

1740 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute.  
1741 This attribute specifies the attributes of the specified Subscription Objects that the Printer  
1742 MUST return in the response. Each value of this attribute is either an attribute name (defined  
1743 in sections 5.3 and 5.4) or an attribute group name (defined in section 11.2.4.1). If the client  
1744 omits this attribute, the Printer MUST respond as if the client had supplied this attribute with  
1745 the one value: ‘notify-subscription-id’.



1746           **11.2.5.1.4 “my-subscriptions” (boolean)**

1747           The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. If  
1748           the value is ‘false’, the Printer MUST consider the Subscription Objects from all users as  
1749           candidates. If the value is ‘true’, the Printer MUST return the Subscription Objects created  
1750           by the requesting user of this request. If the client does not supply this attribute, the Printer  
1751           MUST respond as if the client had supplied the attribute with a value of ‘false’. The means  
1752           for authenticating the requesting user and matching the Subscription Objects is similar to that  
1753           for Jobs which is described in [RFC2911] section 8.  
1754

1755           **11.2.5.2 Get-Subscriptions Response**

1756           The Printer returns the following sets of attributes as part of the Get-Subscriptions Response:

1757           Group 1: Operation Attributes

1758                 Status Message:

1759                         Same as [RFC2911].

1760

1761                 Natural Language and Character Set:

1762                         The “attributes-charset” and “attributes-natural-language” attributes as described in  
1763                         [RFC2911] section 3.1.4.2.

1764

1765           Group 2: Unsupported Attributes

1766                         Same as for Get-Subscription-Attributes.

1767

1768           Groups 3 to N: Subscription Attributes

1769           The Printer responds with one Subscription Attributes Group for each requested Subscription  
1770           Object (see the “notify-job-id” attribute in the Operation Attributes Group of this operation).

1771

1772           The Printer returns Subscription Objects in any order.

1773

1774           If the “limit” attribute is present in the Operation Attributes group of the request, the number  
1775           of Subscription Attributes Groups in the response MUST NOT exceed the value of the “limit”  
1776           attribute.

1777

1778           If there are no Subscription Objects associated with the specified Job or Printer, the Printer  
1779           MUST return zero Subscription Attributes Groups and it MUST NOT treat this case as an  
1780           error, i.e., the status-code MUST be ‘successful-ok’ unless something else causes the status  
1781           code to have some other value.

1782

1783           See the Group 3 response (Subscription Attributes Group) of the Get-Subscription-Attributes  
1784           operation (section 11.2.4.2) for the attributes that a Printer returns in this group.  
1785

## 1786 11.2.6 Renew-Subscription operation

1787 This operation allows a client to request the Printer to extend the lease on a Per-Printer Subscription  
1788 Object.

1789 The Printer MUST support this operation.

1790 The Printer MUST accept this request for a Per-Printer Subscription Object in any of the target  
1791 Printer's states, i.e., 'idle', 'processing', or 'stopped', but MUST NOT change the Printer's "printer-  
1792 state" attribute.

1793 The Printer MUST reject this request for a Per-Job Subscription Object because it has no lease (see  
1794 section 5.4.3). The status code returned MUST be 'client-error-not-possible'.

1795 *Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation MUST  
1796 (1) be the owner of the Per-Printer Subscription Object, (2) have Operator or Administrator access  
1797 rights for the Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the  
1798 Printer's administrator-configured security policy to renew Per-Printer Subscription Objects for the  
1799 target Printer. Otherwise, the Printer MUST reject the operation and return: the 'client-error-  
1800 forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' status code as appropriate.

### 1801 11.2.6.1 Renew-Subscription Request

1802 The following groups of attributes are part of the Renew-Subscription Request:

1803 Group 1: Operation Attributes

1804 Natural Language and Character Set:

1805 The "attributes-charset" and "attributes-natural-language" attributes as described in  
1806 [RFC2911] section 3.1.4.1.

1807

1808 Target:

1809 The "printer-uri" attribute which defines the target for this operation as described in  
1810 [RFC2911] section 3.1.5.

1811

1812 Requesting User Name:

1813 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1814 described in [RFC2911] section 8.3.

1815

#### 1816 11.2.6.1.1 "notify-subscription-id" (integer (1:MAX))

1817 The client MUST supply this attribute. The Printer MUST support this attribute. This  
1818 attribute specifies the Per-Printer Subscription Object whose lease the Printer MUST renew.  
1819 If the client omits this attribute, the Printer MUST reject this request with the 'client-error-  
1820 bad-request' status code.

1821

1822 Group 2: Subscription Template Attributes

1823 **11.2.6.1.2 “notify-lease-duration” (integer(0:MAX))**

1824 The client MAY supply this attribute. It indicates the number of seconds to renew the lease  
1825 for the specified Subscription Object. A value of 0 requests an infinite lease (which MAY  
1826 require Operator access rights). If the client omits this attribute, the Printer MUST use the  
1827 value of the Printer’s “notify-lease-duration-default” attribute. See section 5.3.8 for more  
1828 details.  
1829

1830 **11.2.6.2 Renew-Subscription Response**

1831 The Printer returns the following sets of attributes as part of the Renew-Subscription Response:

1832 Group 1: Operation Attributes

1833 Status Message:

1834 Same as [RFC2911].  
1835

1836 The following are some of the status codes returned (see [RFC2911]):  
1837

1838 **successful-ok:** The operation successfully renewed the lease on the Subscription Object  
1839 for the requested duration.

1840 **successful-ok-ignored-or-substituted-attributes:** The operation successfully renewed  
1841 the lease on the Subscription Object for some duration other than the amount  
1842 requested.

1843 **client-error-not-possible:** The operation failed because the “notify-subscription-id”  
1844 Operation attribute identified a Per-Job Subscription Object.

1845 **client-error-not-found:** The operation failed because the “notify-subscription-id”  
1846 Operation attribute identified a non-existent Subscription Object.  
1847

1848 Natural Language and Character Set:

1849 The “attributes-charset” and “attributes-natural-language” attributes as described in  
1850 [RFC2911] section 3.1.4.2. The “attributes-natural-language” MAY be the natural language  
1851 of the Subscription Object, rather than the one requested.  
1852

1853 Group 2: Unsupported Attributes

1854 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes.  
1855

1856 Group 3: Subscription Attributes

1857 The Printer MUST return the following Subscription Attribute:

1858           **11.2.6.2.1 “notify-lease-duration” (integer(0:MAX))**

1859           The value of this attribute **MUST** be the number of seconds that the Printer has granted for the  
1860           lease of the Subscription Object (see section 5.3.8 for details, such as the value of this  
1861           attribute when the Printer doesn't support the requested value).

1862   **11.2.7 Cancel-Subscription operation**

1863           This operation allows a client to delete a Subscription Object and stop the Printer from delivering more  
1864           Event Notifications. Once performed, there is no way to reference the Subscription Object.

1865           A Printer **MUST** supported this operation.

1866           The Printer **MUST** accept this request in any of the target Printer's states, i.e., 'idle', 'processing', or  
1867           'stopped', but **MUST NOT** change the Printer's "printer-state" attribute.

1868           If the specified Subscription Object is a Per-Job Subscription Object, the Printer **MUST** accept this  
1869           request in any of the target Job's states, but **MUST NOT** change the Job's "job-state" attribute or affect  
1870           the Job.

1871           Note: There is no way to change any attributes on a Subscription Object, except the "notify-lease-  
1872           duration" attribute (using the Renew-Subscription operation). In order to change other attributes, a  
1873           client performs a Subscription Creation Operation and Cancel-Subscription operation on the old  
1874           Subscription Object. If the client wants to avoid missing Event Notifications, it performs the  
1875           Subscription Creation Operation first. If this order would create too many Subscription Objects on the  
1876           Printer, the client reverses the order.

1877           *Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation **MUST**  
1878           (1) be the owner of the Subscription Object, (2) have Operator or Administrator access rights for the  
1879           Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer's  
1880           administrator-configured security policy to cancel the target Subscription Object. Otherwise, the  
1881           Printer **MUST** reject the operation and return: the 'client-error-forbidden', 'client-error-not-  
1882           authenticated', or 'client-error-not-authorized' status code as appropriate.

1883   **11.2.7.1 Cancel-Subscription Request**

1884           The following groups of attributes are part of the Cancel-Subscription Request:

1885           Group 1: Operation Attributes

1886           Natural Language and Character Set:

1887           The "attributes-charset" and "attributes-natural-language" attributes as described in  
1888           [RFC2911] section 3.1.4.1.

1889

1890           Target:

1891           The "printer-uri" attribute which defines the target for this operation as described in  
1892           [RFC2911] section 3.1.5.

1893

1894

Requesting User Name:

1895

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

1896

1897

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

1898

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

1899

1900

1901

1902

1903

#### 11.2.7.2 Cancel-Subscription Response

1904

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

1905

Group 1: Operation Attributes

1906

Status Message:

1907

Same as [RFC2911].

1908

1909

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

1910

1911

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

1912

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

1913

Operation attribute identified a non-existent Subscription Object.

1914

1915

Natural Language and Character Set:

1916

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

1917

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

1918

1919

1920

Group 2: Unsupported Attributes

1921

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

1922

1923

### 12 Status Codes

1924

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

1925

1926

1927

**1928 12.1 successful-ok-ignored-subscriptions (0x0003)**

1929 The Subscription Creation Operation was unable to create all requested Subscription Objects.

1930 For a Create-Job-Subscriptions or Create-Printer-Subscriptions operation, this status code means that  
1931 the Printer created one or more Subscription Objects, but not all requested Subscription Objects.

1932 For a Job Creation operation, this status code means that the Printer created the Job along with zero or  
1933 more Subscription Objects. The Printer returns this status code even if other job attributes are  
1934 unsupported or in conflict. That is, if an IPP Printer finds a warning that would allow it to return  
1935 ‘successful-ok-ignored-subscriptions’ and either ‘successful-ok-ignored-or-substituted-attributes’  
1936 and/or ‘successful-ok-conflicting-attributes’, it MUST return ‘successful-ok-ignored-subscriptions’.

**1937 12.2 client-error-ignored-all-subscriptions (0x0414)**

1938 This status code is the same as ‘successful-ok-ignored-subscriptions’ except that only the Create-Job-  
1939 Subscriptions and Create-Printer-Subscriptions operation return it. They return this status code only  
1940 when the Printer creates zero Subscription Objects.

**1941 13 Status Codes in Subscription Attributes Groups**

1942 This section contains values of the “notify-status-code” (type2 enum) attribute that the Printer returns  
1943 in a Subscription Attributes Group in a response when the corresponding Subscription Object:

- 1944 1. is not created or  
1945 2. is created and some of the client-supplied attributes are not supported.

1946 The following sections are ordered in decreasing order of importance of the status-codes.

**1947 13.1 client-error-uri-scheme-not-supported (0x040C)**

1948 This status code is defined in [RFC2911]. This document extends its meaning and allows it to be in a  
1949 Subscription Attributes Group of a response.

1950 The scheme of the client-supplied URI in a “notify-recipient-uri” Subscription Template Attribute in a  
1951 Subscription Creation Operation is not supported. See section 5.3.1.

**1952 13.2 client-error-attributes-or-values-not-supported (0x040B)**

1953 This status code is defined in [RFC2911]. This document extends its meaning and allows it to be in a  
1954 Subscription Attributes Group of a response.

1955 The method of the client-supplied keyword in a “notify-pull-method” Subscription Template Attribute  
1956 in a Subscription Creation Operation is not supported. See section 5.3.2.

### 1957 **13.3 client-error-too-many-subscriptions (0x0415)**

1958 The number of Subscription Objects supported by the Printer would be exceeded if this Subscription  
1959 Object were created (see section 5.2).

### 1960 **13.4 successful-ok-too-many-events (0x0005)**

1961 The client supplied more Events in the “notify-events” operation attribute of a Subscription Creation  
1962 Operation than the Printer supports, as indicated in its “notify-max-events-supported” Printer attribute  
1963 (see section 5.3.3).

### 1964 **13.5 successful-ok-ignored-or-substituted-attributes (0x0001)**

1965 This status code is defined in [RFC2911]. This document extends its meaning to include unsupported  
1966 Subscription Template Attributes and it can appear in a Subscription Attributes Group.

## 1967 **14 Encodings of Additional Attribute Tags**

1968 This section assigns values to two attributes tags as extensions to the encoding defined in [RFC2910]).

1969 The “subscription-attributes-tag” delimits Subscription Template Attributes Groups in requests and  
1970 Subscription Attributes Groups in responses.

1971 The “event-notification-attributes-tag” delimits Event Notifications in Delivery Methods that use an  
1972 IPP-like encoding.

1973 The following table specifies the values for the delimiter tags:

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

## 1974 **15 Conformance Requirements**

1975 It is OPTIONAL for IPP clients and Printers to implement this Event Notification specification.

1976 **15.1 Conformance requirements for clients**

1977 If this Event Notification specification is implemented by a client, the client **MUST** support the  
1978 'ippget' Pull Delivery Method and meet the conformance requirements as defined in [ipp-get-method]  
1979 for clients. A client **MAY** support additional Delivery Methods.

1980 **15.2 Conformance requirements for Printers**

1981 If this Event Notification specification is implemented by a Printer, the Printer **MUST**:

- 1982 - meet the Conformance Requirements detailed in section 5 of [RFC2911].
- 1983 - support the Subscription Template Attributes Group in requests and the Subscription  
1984 Attributes Group in responses.
- 1985 - support all of the following attributes:
- 1986 a. **REQUIRED** Subscription Object attributes in section 5.
- 1987 b. **REQUIRED** Printer Description object attributes in section 6.
- 1988 c. **REQUIRED** attributes in Event Notification content in section 8.
- 1989 - support the 'ippget' Pull Delivery Method and meet the conformance requirements as defined  
1990 in [ipp-get-method] for Printers. The Printer **MAY** support additional Push and Pull Delivery  
1991 Methods.
- 1992 - deliver Event Notifications that conform to the requirements of section 9 and the requirements  
1993 of the Delivery Method Document for each supported Delivery Method (the conformance  
1994 requirements for Delivery Method Documents is specified in section 10).
- 1995 - for all of the Job Creation Operations that the Printer supports, **MUST** support the  
1996 **REQUIRED** extensions for notification defined in section 11.1.3.
- 1997 - meet the conformance requirements for operations as described in Table 16 and meet the  
1998 requirements for Printers as specified in the indicated sub-sections of section 11:



1999

**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

2000

2001

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

2002

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

2003

2004

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

2005

2006

2007

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

2008

2009

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

2010

2011

2012

b) Printer 1 performs the client-supplied Subscription Creation Operations and also forwards the Subscription Creation Operations to Printer 2 with the Notification Recipient changed to be the Printer 1. When an Event occurs in Printer 2, Printer 2 delivers the Event Notification (B) to Notification Recipient of Printer 1, which relays the received Event Notification (B) to the client-supplied Notification Recipient (as Event Notifications(A) in the diagram). Note, when a client performs a Subscription Creation Operation, Printer 1 need not forward the Subscription Creation Operation to Printer 2 if it would create a duplicate Subscription Object on Printer 2.

2013

2014

2015

2016

2017

2018

2019

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:

2020

2021

2022

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

2023

- 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.

2024

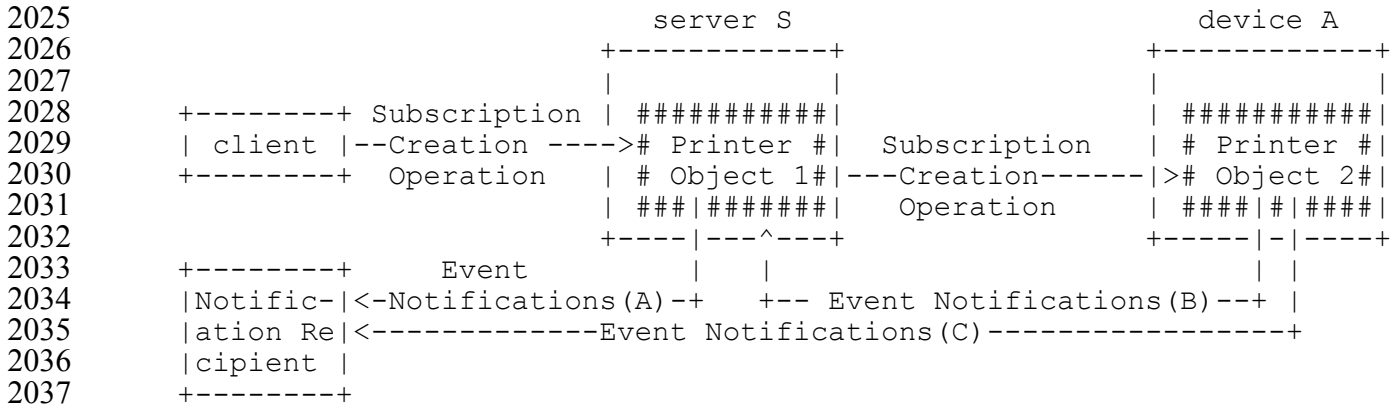


Figure 2 – Model for Notification with Cascading Printers

17 Appendix B - Distributed Model for Notification (Informative)

A Printer implementation could use some other remote notification server to provide some or most of the service. For example, the remote notification server could deliver Event Notifications using Delivery Methods that are not directly supported by the output device or Printer object. Or, the remote notification server 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 deliver the Event Notifications to the Notification Recipient(s).

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

2049

```

2050                                     *****
2051                                     *
2052                                     * Printer in combination
2053                                     * with the distributed
2054                                     * Notification Server)
2055                                     *
2056                                     * output device or server
2057                                     * +-----+
2058 PDA, desktop, or server             * + ##### +
2059 +-----+                             * | # # |
2060 | client |--IPP Subscription-----># Printer # |
2061 +-----+ Creation operation * | # Object # |
2062                                     * | #####|##### |
2063                                     * +-----|-----+
2064                                     * | Subscriptions
2065                                     * | OR Event
2066                                     * | Notifications
2067 +-----+                             * +-----v-----+
2068 |Notification| IPP-defined           * | Notification |
2069 |Recipient |<--Event Notifications--| Server |
2070 +-----+                             * +-----+
2071                                     *
2072                                     *****
2073 *** = Implementation configuration opaque boundary
2074

```

2075 **Figure 3 – Opaque Use of a Notification Server Transparent to the Client**

## 2076 **18 Appendix C - Extended Notification Recipient (Informative)**

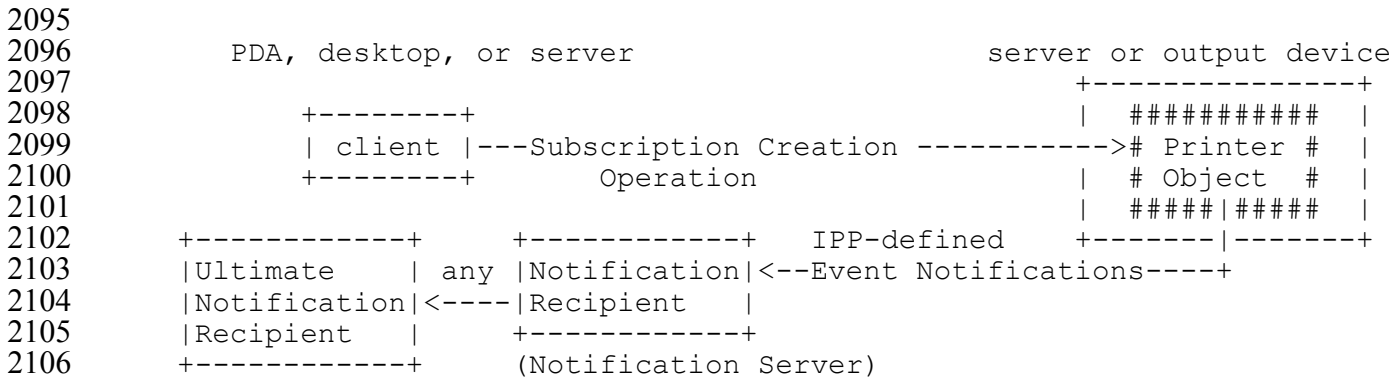
2077 The model allows for an extended Notification Recipient that is itself a notification server that  
 2078 forwards each Event Notification to another recipient (called the Ultimate Notification Recipient in  
 2079 this section). The Delivery Method to the Ultimate Recipient is probably different from the Delivery  
 2080 Method used by the Printer to the extended Notification Recipient.

2081 This extended Notification Recipient is transparent to the Printer but not to the client.

2082 When a client performs a Subscription Creation Operation, it specifies the extended Notification  
 2083 Recipient as it would any Notification Recipient. In addition, the client specifies the Ultimate  
 2084 Notification Recipient in the Subscription Creation Operation in a manner specified by the extended  
 2085 Notification Recipient. Typically, it is either some bytes in the value of “notify-user-data” or some  
 2086 additional parameter in the value of “notify-recipient-uri”. The client also subscribes directly with the  
 2087 extended Notification Recipient (by means outside this document), since it is a notification server in its  
 2088 own right.

2089 The IPP Printer treats the extended Notification Recipient like any other Notification Recipient and the  
 2090 IPP Printer is not aware of the forwarding. The Delivery Method that the extended Notification  
 2091 Recipient uses for delivering the Event Notification to the Ultimate Notification Recipient is beyond  
 2092 the scope of this document and is transparent to the IPP Printer.

2093 Examples of this extended Notification Recipient are paging, immediate messaging services, general  
 2094 notification services, and NOS vendors' infrastructure. Figure 4 shows this approach.



2107 **Figure 4 – Use of an Extended Notification Recipient transparent to the Printer**

2108 **19 Appendix D - Details about Conformance Terminology (Normative)**

2109 The following paragraphs provide more details about conformance terminology.

2110 **REQUIRED** - an adjective used to indicate that a conforming IPP Printer implementation **MUST**  
 2111 support the indicated operation, object, attribute, attribute value, status code, or out-of-band value  
 2112 in requests and responses. See [RFC2911] “Appendix A - Terminology for a definition of  
 2113 “support”. *Since support of this entire Notification specification is OPTIONAL for*  
 2114 *conformance to IPP/1.1, the use of the term REQUIRED in this document means “REQUIRED*  
 2115 *if this OPTIONAL Notification specification is implemented”.*

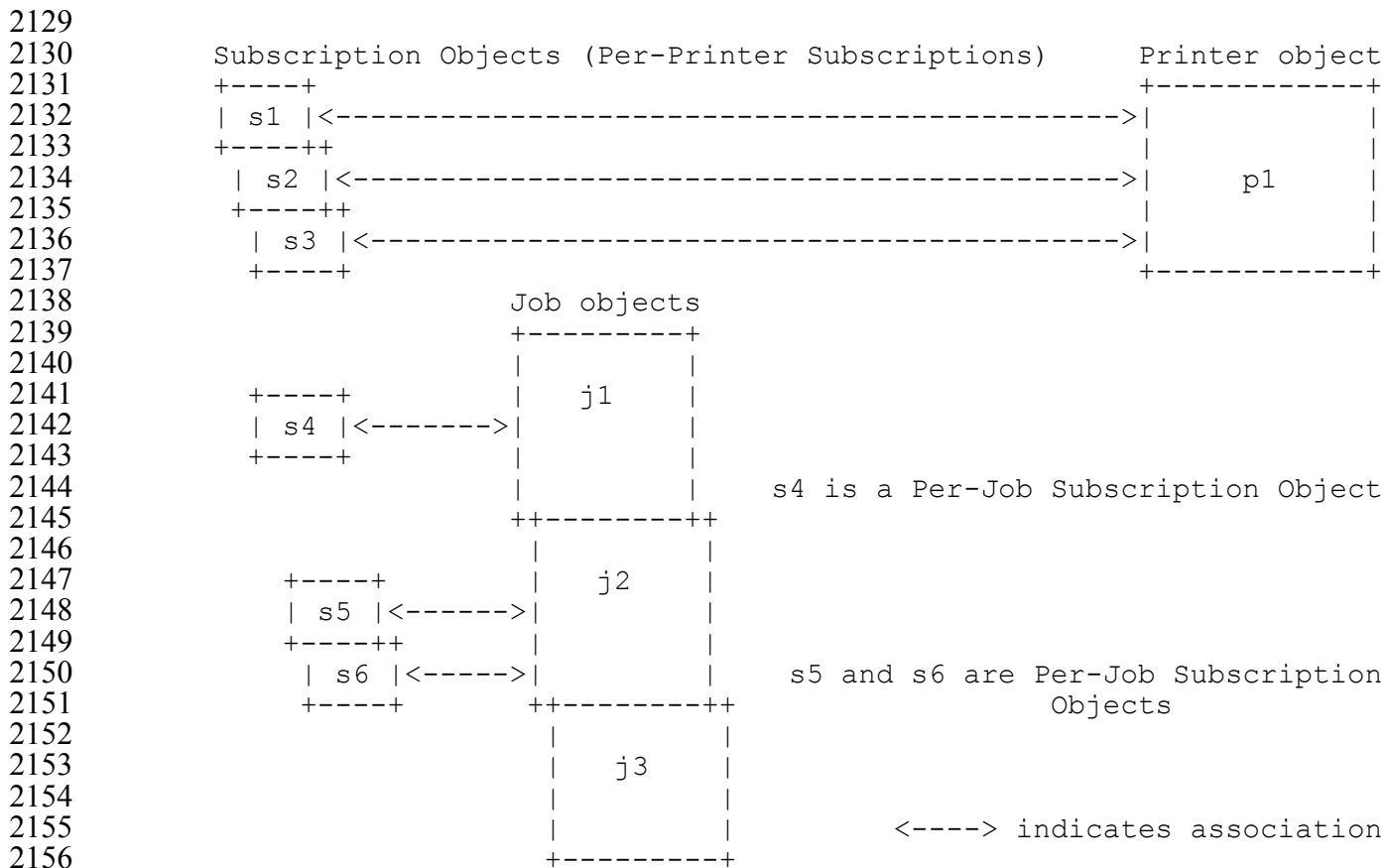
2116 **RECOMMENDED** - an adjective used to indicate that a conforming IPP Printer implementation is  
 2117 recommended to support the indicated operation, object, attribute, attribute value, status code, or  
 2118 out-of-band value in requests and responses. *Since support of this entire Notification*  
 2119 *specification is OPTIONAL for conformance to IPP/1.1, the use of the term RECOMMENDED*  
 2120 *in this document means “RECOMMENDED if this OPTIONAL Notification specification is*  
 2121 *implemented”.*

2122 **OPTIONAL** - an adjective used to indicate that a conforming IPP Printer implementation **MAY**, but is  
 2123 **NOT REQUIRED** to, support the indicated operation, object, attribute, attribute value, status code,  
 2124 or out-of-band value in requests and responses.

2125 **20 Appendix E - Object Model for Notification (Normative)**

2126 This section describes the Notification object model that adds a Subscription Object which together  
 2127 with the Job and Printer object provide the complete Notification semantics.

2128 The object relationships can be seen pictorially as:



2157 **Figure 5 – Object Model for Notification**

2158 s1, s2, and s3 are Per-Printer Subscription Objects and can identify Printer and/or Job Events.

2159 s4, s5, and s6 are Per-Job Subscription Objects and can identify Printer and/or Job Events.

## 2160 20.1 Object relationships

2161 This sub-section defines the object relationships between the Printer, Job, and Subscription Objects by  
 2162 example. Whether Per-Printer Subscription Objects are actually contained in a Printer object or are  
 2163 just bi-directionally associated with them in some way is IMPLEMENTATION DEPENDENT and is  
 2164 transparent to the client. Similarly, whether Per-Job Subscription Objects are actually contained in a  
 2165 Job object or are just bi-directionally associated with them in some way is IMPLEMENTATION  
 2166 DEPENDENT and is transparent to the client. The object relationships are defined as follows:

## 2167 20.2 Printer Object and Per-Printer Subscription Objects

- 2168 1. The Printer object contains (is associated with) zero or more Per-Printer Subscription Objects  
 2169 (p1 contains s1-s3 Per-Printer Subscription Objects).

- 2170           2. Each Per-Printer Subscription Object (s1, s2, and s3) is contained in (or is associated with)  
2171           exactly one Printer object (p1).

### 2172 **20.3 Job Object and Per-Job Subscription Objects**

- 2173           1. A Job object (j1, j2, j3) is associated with zero or more Per-Job Subscription Objects (s4-s6).  
2174           Job j1 is associated with Per-Job Subscription Object s4, Job j2 is associated with Per-Job  
2175           Subscription Objects s5 and s6, and Job j3 is not associated with any Per-Job Subscription  
2176           Object.
- 2177           2. Each Per-Job Subscription Object is associated with exactly one Job object.

## 2178 **21 Appendix F - Per-Job versus Per-Printer Subscription Objects (Normative)**

2179           Per-Job and Per-Printer Subscription Objects are quite similar. Either type of Subscription Object can  
2180           subscribe to Job Events, Printer Events, or both. Both types of Subscription Objects can be queried  
2181           using the Get-Subscriptions and Get-Subscription-Attributes operations and canceled using the Cancel-  
2182           Subscription operation. Both types of Subscription Objects create Subscription Objects which have  
2183           the same Subscription Object attributes defined. However, there are some semantic differences  
2184           between Per-Job Subscription Objects and Per-Printer Subscription Objects. A Per-Job Subscription  
2185           Object is established by the client when submitting a job and after creating the job using the Create-  
2186           Job-Subscriptions operation by specifying the “job-id” of the Job with the “notify-job-id” attribute. A  
2187           Per-Printer Subscription Object is established between a client and a Printer using the Create-Printer-  
2188           Subscriptions operation. Some specific differences are:

- 2189           1. A client usually creates one or more Per-Job Subscription Objects as part of the Job Creation  
2190           operations (Create-Job, Print-Job, and Print-URI), rather than using the OPTIONAL Create-Job-  
2191           Subscriptions operation, especially since Printer implementations NEED NOT support the  
2192           Create-Job-Subscriptions operation, since it is OPTIONAL.
- 2193           2. For Per-Job Subscription Objects, the Subscription Object is only valid while the job is “not-  
2194           complete” (see sections 5.4.3) while for the Per-Printer Subscription Objects, the Subscription  
2195           Object is valid until the time (in seconds) that the Printer returned in the “notify-lease-  
2196           expiration-time” operation attribute.
- 2197           3. Job Events in a Per-Job Subscription Object apply only to “one job” (the Job created by the Job  
2198           Creation operation or references by the Create-Job-Subscriptions operation) while Job Events  
2199           in a Per-Printer Subscription Object apply to ALL jobs contained in the IPP Printer.

## 2200 **22 Normative References**

- 2201           [ipp-get-method]  
2202           Herriot, R., and T. Hastings, "Internet Printing Protocol (IPP): The ‘ippget’ Delivery Method for  
2203           Event Notifications", <draft-ietf-ipp-notify-get-08.txt>, September 10, 2002.

- 2204 [RFC2119]  
2205 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March 1997
- 2206 [RFC2396]  
2207 Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic  
2208 Syntax", RFC 2396, August 1998.
- 2209 [RFC2717]  
2210 R. Petke and I. King, "Registration Procedures for URL Scheme Names", RFC 2717, November  
2211 1999.
- 2212 [RFC2910]  
2213 Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.1: Encoding and  
2214 Transport", RFC 2910, September 2000.
- 2215 [RFC2911]  
2216 deBry, R., Hastings, T., Herriot, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.1:  
2217 Model and Semantics", RFC 2911, September 2000.
- 2218 [RFC3381]  
2219 Hastings, T., Lewis, H., and R. Bergman, "IPP: Job Progress Attributes", RFC 3381, September  
2220 2002.

## 2221 **23 Informative References**

- 2222 [IANA-CON]  
2223 Narte, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs",  
2224 BCP 26, RFC 2434, October 1998.
- 2225 [ipp-not-req]  
2226 deBry, R., Lewis, H., and T. Hastings, "Internet Printing Protocol/1.1: Requirements for IPP  
2227 Notifications", <draft-ietf-ipp-not-06.txt>, work in progress, July 17, 2001.
- 2228 [RFC2565]  
2229 Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.0: Encoding and  
2230 Transport", RFC 2565, April 1999.
- 2231 [RFC2566]  
2232 deBry, R., , Hastings, T., Herriot, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.0:  
2233 Model and Semantics", RFC 2566, April 1999.
- 2234 [RFC2567]  
2235 Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.
- 2236 [RFC2568]  
2237 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",  
2238 RFC 2568, April 1999.

- 2239 [RFC2569]  
 2240 Herriot, R., Hastings, T., Jacobs, N., and J. Martin, "Mapping between LPD and IPP Protocols",  
 2241 RFC 2569, April 1999.
- 2242 [RFC2616]  
 2243 Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee,  
 2244 "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
- 2245 [RFC3196]  
 2246 Hastings, T., Manros, C., Zehler, P., Kugler, C., and H. Holst, "Internet Printing Protocol/1.1:  
 2247 Implementer's Guide", RFC3196, November 2001.

## 2248 24 IANA Considerations

2249 This section contains the registration information for IANA to add to the [various-IPP Registryies](#)  
 2250 according to the procedures defined in RFC 2911 [RFC2911] section 6 to cover the definitions in this  
 2251 document. In addition, this section defines how Events and Delivery Methods will be registered when  
 2252 they are defined in other documents. The resulting registrations will be published in the  
 2253 <http://www.iana.org/assignments/ipp-registrations> registry.

2254 *Note to RFC Editors: Replace RFC NNNN below ([but not RFC xxxx](#)) with the RFC number for this*  
 2255 *document, so that it accurately reflects the content of the information for the IANA Registry.*

### 2256 24.1 Attribute Registrations

2257 The following table lists all the attributes defined in this document. These are to be registered  
 2258 according to the procedures in RFC 2911 [RFC2911] section 6.2.

Subscription Template attributes:	Reference	Section
-----	-----	-----
2260 notify-attributes (1setOf type2 keyword)	[RFCNNNN]	5.3.4
2262 notify-attributes-supported (1setOf type2 keyword)	[RFCNNNN]	5.3.4.1
2263 notify-charset (charset)	[RFCNNNN]	5.3.6
2264 notify-events (1setOf type2 keyword)	[RFCNNNN]	5.3.3
2265 notify-events-default (1setOf type2 keyword)	[RFCNNNN]	5.3.3.1
2266 notify-events-supported (1setOf type2 keyword)	[RFCNNNN]	5.3.3.2
2267 notify-lease-duration (integer(0:67108863))	[RFCNNNN]	5.3.8
2268 notify-lease-duration-default (integer(0:67108863))	[RFCNNNN]	5.3.8.1
2269 notify-lease-duration-supported (1setOf (integer(0: 67108863)	[RFCNNNN]	5.3.8.2
2270 rangeOfInteger(0:67108863)))	[RFCNNNN]	5.3.3.3
2271 notify-max-events-supported (integer(2:MAX))	[RFCNNNN]	5.3.7
2272 notify-natural-language (naturalLanguage)	[RFCNNNN]	5.3.2
2273 notify-pull-method (type2 keyword)	[RFCNNNN]	5.3.2.1
2274 notify-pull-method-supported (1setOf type2 keyword)	[RFCNNNN]	
2275		
2276		
2277		



2278	notify-recipient-uri (uri)	[RFCNNNN]	5.3.1
2279	notify-schemes-supported (1setOf uriScheme)	[RFCNNNN]	5.3.1.1
2280	notify-time-interval (integer(0:MAX))	[RFCNNNN]	5.3.9
2281	notify-user-data (octetString(63))	[RFCNNNN]	5.3.5
2282			
2283	Subscription Description Attributes:		
2284	notify-job-id (integer(1:MAX))	[RFCNNNN]	5.4.6
2285	notify-lease-expiration-time (integer(0:MAX))	[RFCNNNN]	5.4.3
2286	notify-printer-up-time (integer(1:MAX))	[RFCNNNN]	5.4.4
2287	notify-printer-uri (uri)	[RFCNNNN]	5.4.5
2288	notify-sequence-number (integer (0:MAX))	[RFCNNNN]	5.4.2
2289	notify-subscriber-user-name (name(MAX))	[RFCNNNN]	5.4.7
2290	notify-subscription-id (integer (1:MAX))	[RFCNNNN]	5.4.1
2291			
2292	Printer Description Attributes:		
2293	printer-state-change-date-time (dateTime)	[RFCNNNN]	6.2
2294	printer-state-change-time (integer(1:MAX))	[RFCNNNN]	6.1
2295			
2296	Attributes Only in Event Notifications		
2297	notify-subscribed-event (type2 keyword)	[RFCNNNN]	8.1
2298	notify-text (text(MAX))	[RFCNNNN]	8.2
2299			
2300			

## 2301 24.2 Additional Enum Attribute Value Registrations

2302 The following table lists all the new enum attribute values defined in this document. These are to be  
 2303 registered according to the procedures in RFC 2911 [RFC2911] section 6.1.

2304	Attribute			
2305	Value	Name	Reference	Section
2306	-----	-----	-----	-----
2307	operations-supported (type2 enum)		[RFC2911]	4.4.15
2308	0x0016	Create-Printer-Subscriptions	[RFCNNNN]	7.1
2309	0x0017	Create-Job-Subscriptions	[RFCNNNN]	7.1
2310	0x0018	Get-Subscription-Attributes	[RFCNNNN]	7.1
2311	0x0019	Get-Subscriptions	[RFCNNNN]	7.1
2312	0x001A	Renew-Subscription	[RFCNNNN]	7.1
2313	0x001B	Cancel-Subscription	[RFCNNNN]	7.1
2314				

## 2315 24.3 Operation Registrations

2316 The following table lists all of the operations defined in this document. These are to be registered  
 2317 according to the procedures in RFC 2911 [RFC2911] section 6.4.

2318	Operation Name	Reference	Section
2319	-----	-----	-----
2320	<del>Cancel-Subscription-Operation</del>		[RFCNNNN]
2321	11.2.7		
2322	<del>Create-Job - Extensions</del>	[RFCNNNN]	11.1.3
2323	<del>Create-Job-Subscriptions-Operation</del>		[RFCNNNN]
2324	11.1.1		
2325	<del>Create-Printer-Subscriptions-Operation</del>		[RFCNNNN]
2326	11.1.2		
2327	Get-Printer-Attributes - Extensions	[RFCNNNN]	11.2.3
2328	<del>Get-Subscription-Attributes-Operation</del>		[RFCNNNN]
2329	11.2.4		
2330	<del>Get-Subscriptions-Operation</del>		[RFCNNNN]
2331	11.2.5		
2332	<del>Print-Job - Extensions</del>	[RFCNNNN]	11.1.3
2333	<del>Print-URI - Extensions</del>	[RFCNNNN]	11.1.3
2334	<del>Job Creation Operations - Extensions</del>	RFC NNNN	11.1.3
2335	<del>Renew-Subscription-Operation</del>		[RFCNNNN]
2336	11.2.6		
2337	Validate-Job Operation - Extensions	[RFCNNNN]	11.2.2
2338			

#### 2339 24.4 Status code Registrations

2340 The following table lists all the status codes defined in this document. These are to be registered  
 2341 according to the procedures in RFC 2911 [RFC2911] section 6.6.

2342	Value	Status Code Name	Reference	Section
2343	-----	-----	-----	-----
2344	0x0000:0x00FF - <del>"s</del> Successful:"			
2345	0x0003	successful-ok-ignored-subscriptions	[RFCNNNN]	12.1
2346				
2347	0x0400:0x04FF - <del>"e</del> Client_eError:"			
2348	0x0414	client-error-ignored-all-subscriptions	[RFCNNNN]	12.2
2349				

#### 2350 24.5 Attribute Group tag Registrations

2351 The following table lists all the attribute group tags defined in this document. These are to be  
 2352 registered according to the procedures in RFC 2911 [RFC2911] section 6.5.

2353	Value	Attribute Group Tag Name	Reference	Section
2354	-----	-----	-----	-----
2355	0x06	subscription-attributes-tag	[RFCNNNN]	14
2356	0x07	event-notification-attributes-tag	[RFCNNNN]	14
2357				

2358 **24.6 Registration of Events**

2359 The following table lists all the Events defined in this document as ~~When other document define~~  
 2360 ~~additional~~-type2 keywords to be used with the “notify-events”, “notify-events-default”, and “notify-  
 2361 events-supported” Subscription Template attributes (see section 5.3.3)).~~;~~ Rather than creating a  
 2362 separate section in the IPP Registry for Events, these event keywords will be registered according to  
 2363 the procedures of [RFC2911] section 7.1 as additional keyword attribute values for use with the  
 2364 “notify-events” Subscription Template attribute (see section 5.3.3), i.e., registered as keyword values  
 2365 for the "notify-events", "notify-events-default", and "notify-events-supported" attributes.;

2366 ~~Therefore, the IPP Registry entry for an Event will be of the form:~~

2367	Attribute ( <u>attribute syntax</u> )		Reference	Section
2368	Value			
2369	-----		-----	-----
2370	notify-events (1setOf type2 keyword)		[RFCNNNN]	5.3.3
2371	notify-events-default (1setOf type2 keyword)		[RFCNNNN]	5.3.3.1
2372	notify-events-supported (1setOf type2 keyword)		[RFCNNNN]	5.3.3.2
2373	<u>notify-subscribed-event (type2 keyword)</u>		<u>[RFCNNNN]</u>	<u>8.1</u>
2374	<u>No Events:</u>			
2375	<u>none</u>		[RFCNNNN]	5.3.3.4.1
2376				
2377	<u>Printer Events:</u>			
2378	<u>printer-state-changed</u>		[RFCNNNN]	5.3.3.4.2
2379	<u>printer-restarted</u>		[RFCNNNN]	5.3.3.4.2
2380	<u>printer-shutdown</u>		[RFCNNNN]	5.3.3.4.2
2381	<u>printer-stopped</u>		[RFCNNNN]	5.3.3.4.2
2382	<u>printer-config-changed</u>		[RFCNNNN]	5.3.3.4.2
2383	<u>printer-media-changed</u>		[RFCNNNN]	5.3.3.4.2
2384	<u>printer-finishings-changed</u>		[RFCNNNN]	5.3.3.4.2
2385	<u>printer-queue-order-changed</u>		[RFCNNNN]	5.3.3.4.2
2386	<u>Job Events:</u>			
2387	<u>job-state-changed</u>		[RFCNNNN]	5.3.3.4.3
2388	<u>job-created</u>		[RFCNNNN]	5.3.3.4.3
2389	<u>job-completed</u>		[RFCNNNN]	5.3.3.4.3
2390	<u>job-stopped</u>		[RFCNNNN]	5.3.3.4.3
2391	<u>job-config-changed</u>		[RFCNNNN]	5.3.3.4.3
2392	<u>job-progress</u>		[RFCNNNN]	5.3.3.4.3
2393	<del>&lt;event keyword name&gt;</del>		<del>RFC xxxx</del>	<del>m.n</del>
2394				

2395 **24.7 Registration of Event Notification Delivery Methods**

2396 This section describes the requirements and procedures for registration and publication of Event  
 2397 Notification Delivery Methods and for the submission of such proposals.

## 2398 **24.7.1 Requirements for Registration of Event Notification Delivery Methods**

2399 Registered IPP Event Notification Delivery Methods are expected to follow a number of requirements  
2400 described below.

### 2401 **24.7.1.1 Required Characteristics**

2402 A Delivery Method Document MUST either (1) contain all of the semantics of the Delivery Method or  
2403 (2) contain the IPP Delivery Method registration requirements and a profile of some other protocol that  
2404 in combination is the Delivery Method (e.g., mailto). The Delivery Method Document (and any  
2405 documents it requires) MUST define either (1) a URL for a Push Delivery Method that meets the  
2406 requirements of [RFC2717]. or (2) a keyword for a Pull Delivery method.

2407  
2408 IPP Event Notification Delivery Method Documents MUST meet the requirements of this document  
2409 (see sections 9 and 10).

2410 In addition, a Delivery Method Document MUST contain the following information:

2411

2412 Type of registration: IPP Event Notification Delivery Method

2413 Name of this delivery method:

2414 Proposed URL scheme name of this Push Delivery Method or the keyword name of this Pull  
2415 Delivery Method:

2416 Name of proposer:

2417 Address of proposer:

2418 Email address of proposer:

2419 Is this delivery method REQUIRED or OPTIONAL for conformance to the IPP Event Notification  
2420 and Subscriptions document:

2421 Is this delivery method defining Machine Consumable and/or Human Consumable content:

2422

### 2423 **24.7.1.2 Naming Requirements**

2424 Exactly one (URL scheme or keyword) name MUST be assigned to each Delivery Method.

2425 Each assigned name MUST uniquely identify a single Delivery Method. All Push Delivery Method  
2426 names MUST conform to the rules for URL scheme names, according to [RFC2396] and [RFC2717]  
2427 for schemes in the IETF tree. All Pull Delivery Method names MUST conform to the rules for  
2428 keywords according to [RFC2911].

### 2429 **24.7.1.3 Functionality Requirements**

2430 Delivery Methods MUST function as a protocol that is capable of delivering (push or pull) IPP Event  
2431 Notifications to Notification Recipients.

#### 2432 **24.7.1.4 Usage and Implementation Requirements**

2433 Use of a large number of Delivery Methods may hamper interoperability. However, the use of a large  
2434 number of undocumented and/or unlabelled Delivery Methods hampers interoperability even more.

2435 A Delivery Method should therefore be registered ONLY if it adds significant functionality that is  
2436 valuable to a large community, OR if it documents existing practice in a large community. Note that  
2437 Delivery Methods registered for the second reason should be explicitly marked as being of limited or  
2438 specialized use and should only be used with prior bilateral agreement.

#### 2439 **24.7.1.5 Publication Requirements**

2440 Delivery Method Documents MUST be published in a standards track, informational, or experimental  
2441 RFCs.

#### 2442 **24.7.2 Registration Procedure**

2443 The IPP WG is developing a small number of Delivery Methods which are intended to be published as  
2444 standards track RFCs. However, some parties may wish to register additional Delivery Methods in the  
2445 future. This section describes the procedures for these additional Delivery Methods.

##### 2446 **24.7.2.1 Present the proposal to the Community**

2447 First the Delivery Method Document MUST be an Internet-Draft with a target category of standards  
2448 track, informational, or experimental. The same MUST be true for any documents that it references.

2449 Deliver the proposed Delivery Method Document proposal to the “ipp@pwg.org” mailing list. This  
2450 mailing list has been established by [RFC2911] for reviewing proposed registrations and discussing  
2451 other IPP matters. Proposed Delivery Method Documents are not formally registered and MUST NOT  
2452 be used until approved.

2453 The intent of the public posting is to solicit comments and feedback on the definition and suitability of  
2454 the Delivery Method and the name chosen for it over a four week period.

##### 2455 **24.7.2.2 Delivery Method Reviewer**

2456 The Delivery Method Reviewer is the same person who has been appointed by the IETF Application  
2457 Area Director(s) as the IPP Designated Expert according to [RFC2911] and [IANA-CON]. When the  
2458 four week period is over and the IPP Designated Expert is convinced that consensus has been  
2459 achieved, the IPP Designated Expert either approves the request for registration or rejects it. Rejection  
2460 may occur because of significant objections raised on the list or objections raised externally.

2461 Decisions made by the Reviewer must be posted to the ipp@pwg.org mailing list within 14 days.  
2462 Decisions made by the Reviewer may be appealed to the IESG.

### 2463 24.7.2.3 IANA Registration

2464 Provided that the Delivery Method registration proposal has either passed review or has been  
 2465 successfully appealed to the IESG, the IANA will be notified by the delivery method reviewer and  
 2466 asked to register the Delivery Method and make it available to the community.

### 2467 24.7.3 Delivery Method Document Registrations

2468 Each Push Delivery Method Document defines a URI scheme. Such a URI scheme is used in a URI  
 2469 value of the “notification-recipient” (uri) Subscription Template attribute (see section 5.3.1) and the  
 2470 uriScheme value of the “notify-schemes-supported” (1setOf uriScheme 5.3.1.1) Printer attribute(see  
 2471 section ). Rather than creating a separate section in the IPP Registry for Delivery Methods, Push  
 2472 Delivery Methods will be ~~which is~~ registered as an additional value of the “notify-schemes-supported”  
 2473 Printer attribute. These uriScheme values will be registered according to the procedures of [RFC2911]  
 2474 section 7.1 for additional attribute values. Therefore, the IPP Registry entry for a Push Delivery  
 2475 Method will be of the form:

2476	Attribute			
2477	Value		Ref.	Section
2478	-----		-----	-----
2479	notify-schemes-supported	( <u>1setOf uriScheme</u> <del>type2</del> keyword)		—RFC
2480	xxxx 5.3.1.1			
2481	<scheme name>		RFC xxxx	m.n

2483 Each Pull Delivery Method Document defines a keyword method which is registered as an additional  
 2484 value of the “notify-pull-method” and “notify-pull-method-supported” Printer attributes. These  
 2485 keyword values will be registered according to the procedures of [RFC2911] section 7.1 for additional  
 2486 attribute values. Therefore, the IPP Registry entry for a Pull Delivery Method will be of the form:

2487	Attribute			
2488	Value		Ref.	Section
2489	-----		-----	-----
2490	notify-pull-method	(type2 keyword)	[ipp-ntfy]	5.3.2
2491	notify-pull-method-supported	(1setOf type2 keyword)		
2492			[ipp-ntfy]	5.3.2.1
2493	<method keyword name>		RFC xxxx	m.n

### 2495 24.7.4 Registration Template

2496 To: ipp@pwg.org  
 2497 Subject: Registration of a new Delivery Method

2498  
 2499 Delivery Method name:

2500  
 2501 (All Push Delivery Method names must be suitable for use as the value of a URL scheme in the IETF  
 2502 tree and all Pull Delivery Method names must be suitable IPP keywords according to [RFC2911])

2503  
2504  
2505  
2506  
2507  
2508  
2509  
2510

Published specification(s):

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

Person & email address to contact for further information:

## 2511 **25 Intellectual Property**

2512 The IETF takes no position regarding the validity or scope of any intellectual property or other rights  
2513 that might be claimed to pertain to the implementation or use of the technology described in this  
2514 document or the extent to which any license under such rights might or might not be available; neither  
2515 does it represent that it has made any effort to identify any such rights. Information on the IETF's  
2516 procedures with respect to rights in standards-track and standards-related documentation can be found  
2517 in RFC 2028. Copies of claims of rights made available for publication and any assurances of licenses  
2518 to be made available, or the result of an attempt made to obtain a general license or permission for the  
2519 use of such proprietary rights by implementers or users of this specification can be obtained from the  
2520 IETF Secretariat.

2521 The IETF invites any interested party to bring to its attention any copyrights, patents or patent  
2522 applications, or other proprietary rights which may cover technology that may be required to practice  
2523 this standard. Please address the information to the IETF Executive Director.

## 2524 **26 Internationalization Considerations**

2525 This IPP Notification specification continues support for the internationalization of [RFC2911] of  
2526 attributes containing text strings and names. Allowing a Subscribing Client to specify a different  
2527 natural language and charset for each Subscription Object increases the internationalization support.

2528 The Printer MUST be able to localize the content of Human Consumable Event Notifications and to  
2529 localize the value of “notify-text” attribute in Machine Consumable Event Notifications that it delivers  
2530 to Notification Recipients. For localization, the Printer MUST use the value of the “notify-charset”  
2531 attribute and the “notify-natural-language” attribute in the Subscription Object supplied by the  
2532 Subscribing Client.

## 2533 **27 Security Considerations**

2534 Clients submitting Notification requests to the IPP Printer have the same security issues as submitting  
2535 an IPP/1.1 print job request (see [RFC2911] section 3.2.1 and section 8). The same mechanisms used  
2536 by IPP/1.1 can therefore be used by the client Notification submission. Operations that require  
2537 authentication can use the HTTP authentication. Operations that require privacy can use the  
2538 HTTP/TLS privacy. As with IPP/1.1 Print Job Objects, if there is no security on Subscription Objects,  
2539 sequential assignment of subscription-ids exposes the system to a passive traffic monitoring threat.

2540 **27.1 Client access rights**

2541 The Subscription Object access control model is the same as the access control model for Job objects.  
2542 The client MUST have the following access rights for the indicated Subscription operations:

- 2543 1. Create-Job-Subscriptions (see section 11.1.1): A Per-Job Subscription object is associated with  
2544 a Job. To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section  
2545 8.3) performing this operation MUST (1) be the job owner, (2) have Operator or Administrator  
2546 access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized  
2547 by the Printer's administrator-configured security policy to create Per-Job Subscription Objects  
2548 for the target job.
- 2549 2. Create-Printer-Subscriptions (see section 11.1.2): A Per-Printer Subscription object is  
2550 associated with the Printer. To create Per-Printer Subscription Objects, the authenticated user  
2551 (see [RFC2911] section 8.3) performing this operation MUST (1) have Operator or  
2552 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5) or (2) be  
2553 otherwise authorized by the Printer's administrator-configured security policy to create Per-  
2554 Printer Subscription Objects for this Printer.
- 2555 3. Get-Subscription-Attributes (see section 11.2.4): The access control model for this operation is  
2556 the same as that of the Get-Job-Attributes operation (see [RFC2911] section 3.3.4). The  
2557 primary difference is that a Get-Subscription-Attributes operation is directed at a Subscription  
2558 Object rather than at a Job object, and a returned attribute group contains Subscription Object  
2559 attributes rather than Job object attributes. To query the specified Subscription Object, the  
2560 authenticated user (see [RFC2911] section 8.3) performing this operation MUST (1) be the  
2561 Subscription Object owner, (2) have Operator or Administrator access rights for this Printer  
2562 (see [RFC2911] sections 1 and 8.5), or (3) be otherwise authorized by the Printer's  
2563 administrator-configured security policy to query the Subscription Object for the target job.  
2564 Furthermore, the Printer's security policy MAY limit which attributes are returned, in a manner  
2565 similar to the Get-Job-Attributes operation (see [RFC2911] end of section 3.3.4.2).
- 2566 4. Get-Subscriptions (see section 11.2.5): The access control model for this operation is the same  
2567 as that of the Get-Jobs operation (see [RFC2911] section 3.2.6). The primary difference is that  
2568 the operation is directed at Subscription Objects rather than at Job objects, and the returned  
2569 attribute groups contain Subscription Object attributes rather than Job object attributes. To  
2570 query Per-Job Subscription Objects of the specified job (client supplied the "notify-job-id"  
2571 operation attribute - see section 11.2.5.1.1), the authenticated user (see [RFC2911] section 8.3)  
2572 performing this operation MUST (1) be the Subscription Object owner, (2) have Operator or  
2573 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (3) be  
2574 otherwise authorized by the Printer's administrator-configured security policy to query the  
2575 Subscription Object for the target job. To query Per-Printer Subscription Objects of the Printer  
2576 (client omits the "notify-job-id" operation attribute - see section 11.2.5.1.1), the authenticated  
2577 user (see [RFC2911] section 8.3) performing this operation MUST (1) have Operator or  
2578 Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5), or (2) be  
2579 otherwise authorized by the Printer's administrator-configured security policy to query Per-  
2580 Printer Subscription Objects for the target Printer. Furthermore, the Printer's security policy



2581 MAY limit which attributes are returned, in a manner similar to the Get-Job-Attributes  
2582 operation (see [RFC2911] end of section 3.2.6.2).

2583 5. Renew-Subscriptions (see section 11.2.6): The authenticated user (see [RFC2911] section 8.3)  
2584 performing this operation MUST (1) be the owner of the Per-Printer Subscription Object, (2)  
2585 have Operator or Administrator access rights for the Printer (see [RFC2911] sections 1 and  
2586 8.5), or (3) be otherwise authorized by the Printer's administrator-configured security policy to  
2587 renew Per-Printer Subscription Objects for the target Printer

2588 6. Cancel-Subscription (see section 11.2.7): The authenticated user (see [RFC2911] section 8.3)  
2589 performing this operation MUST (1) be the owner of the Subscription Object, (2) have  
2590 Operator or Administrator access rights for the Printer (see [RFC2911] sections 1 and 8.5), or  
2591 (3) be otherwise authorized by the Printer's administrator-configured security policy to cancel  
2592 the target Subscription Object.

2593 The standard security concerns (delivery to the right user, privacy of content, tamper proof content)  
2594 apply to each Delivery Method. Some Delivery Methods are more secure than others. Each Delivery  
2595 Method Document MUST discuss its Security Considerations.

## 2596 **27.2 Printer security threats**

2597 Notification trap door: If a Printer supports the OPTIONAL "notify-attributes" Subscription Template  
2598 attribute (see section 5.3.4) where the client can request that the Printer return any specified Job,  
2599 Printer, and Subscription object attributes, the Printer MUST apply the same security policy to these  
2600 requested attributes in the Get-Notifications request as it does for the Get-Jobs, Get-Job-Attributes,  
2601 Get-Printer-Attributes, and Get-Subscription-Attributes requests.

## 2602 **27.3 Notification Recipient security threats**

2603 Unwanted Events Notifications (spam): For any Push Delivery Method, by far the biggest security  
2604 concern is the abuse of notification: delivering unwanted Event Notifications to third parties (i.e.,  
2605 spam). The problem is made worse by notification addresses that may be redistributed to multiple  
2606 parties. There exist scenarios where third party notification is used (see Scenario #2 and #3 in [ipp-  
2607 not-req]). Any fully secure solution would require active agreement of all recipients before delivering  
2608 anything.

## 2609 **28 Contributors**

2610 The following people made significant contributions to the design and review of this specification:

2611 Scott A. Isaacson  
2612 Novell, Inc.  
2613 122 E 1700 S  
2614 Provo, UT 84606  
2615

2616 Phone: 801-861-7366  
2617 Fax: 801-861-2517  
2618 e-mail: [sisaacson@novell.com](mailto:sisaacson@novell.com)  
2619  
2620 Roger deBry  
2621 Utah Valley State College  
2622 Orem, UT 84058  
2623  
2624 Phone: (801) 222-8000  
2625 EMail: [debryro@uvsc.edu](mailto:debryro@uvsc.edu)  
2626  
2627 Jay Martin  
2628 Underscore Inc.  
2629 9 Jacqueline St.  
2630 Hudson, NH 03051-5308  
2631 603-889-7000  
2632 fax: 775-414-0245  
2633 e-mail: [jkm@underscore.com](mailto:jkm@underscore.com)  
2634  
2635 Michael Shepherd  
2636 Xerox Corporation  
2637 800 Phillips Road MS 128-51E  
2638 Webster, NY 14450  
2639  
2640 Phone: 716-422-2338  
2641 Fax: 716-265-8871  
2642 e-mail: [mshepherd@usa.xerox.com](mailto:mshepherd@usa.xerox.com)  
2643  
2644 Ron Bergman  
2645 Hitachi Koki Imaging Solutions  
2646 1757 Tapo Canyon Road  
2647 Simi Valley, CA 93063-3394  
2648  
2649 Phone: 805-578-4421  
2650 Fax: 805-578-4001  
2651 Email: [rbergma@hitachi-hkis.com](mailto:rbergma@hitachi-hkis.com)

## 2652 **29 Author's Addresses**

2653 Robert Herriot  
2654 706 Colorado Ave.  
2655 Palo Alto, CA 94303  
2656  
2657 Phone: 650-327-4466  
2658 Fax: 650-327-4466  
2659 Email: [bob@herriot.com](mailto:bob@herriot.com)

2660

2661 Tom Hastings  
2662 Xerox Corporation  
2663 737 Hawaii St. ESAE 231  
2664 El Segundo, CA 90245

2665

2666 Phone: 310-333-6413

2667 Fax: 310-333-5514

2668 e-mail: [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)

2669

2670 IPP Web Page: <http://www.pwg.org/ipp/>2671 IPP Mailing List: [ipp@pwg.org](mailto:ipp@pwg.org)

2672

2673 To subscribe to the ipp mailing list, send the following email:

2674 1) send it to [majordomo@pwg.org](mailto:majordomo@pwg.org)

2675 2) leave the subject line blank

2676 3) put the following two lines in the message body:

2677 subscribe ipp

2678 end

2679

2680 Implementers of this specification document are encouraged to join the IPP Mailing List in order to  
2681 participate in any discussions of clarification issues and review of registration proposals for additional  
2682 attributes and values. In order to reduce spam the mailing list rejects mail from non-subscribers, so  
2683 you must subscribe to the mailing list in order to send a question or comment to the mailing list.

## 2684 **30 Appendix G - Description of the base IPP documents (Informative)**

2685 The base set of IPP documents includes:

2686 Design Goals for an Internet Printing Protocol [RFC2567]

2687 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]

2688 Internet Printing Protocol/1.1: Model and Semantics [RFC2911]

2689 Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]

2690 Internet Printing Protocol/1.1: Implementer's Guide [RFC3196]

2691 Mapping between LPD and IPP Protocols [RFC2569]

2692

2693 The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed  
2694 printing functionality, and it enumerates real-life scenarios that help to clarify the features that need to  
2695 be included in a printing protocol for the Internet. It identifies requirements for three types of users:  
2696 end users, operators, and administrators. It calls out a subset of end user requirements that are satisfied  
2697 in IPP/1.0 [RFC2566, RFC2565]. A few OPTIONAL operator operations have been added to IPP/1.1  
2698 [RFC2911, RFC2910].

2699 The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document  
2700 describes IPP from a high level view, defines a roadmap for the various documents that form the suite

2701 of IPP specification documents, and gives background and rationale for the IETF IPP working group's  
2702 major decisions.

2703 The "Internet Printing Protocol/1.1: Model and Semantics" document describes a simplified model  
2704 with abstract objects, their attributes, and their operations. The model introduces a Printer and a Job.  
2705 The Job supports multiple documents per Job. The model document also addresses how security,  
2706 internationalization, and directory issues are addressed.

2707 The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the  
2708 abstract operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It also  
2709 defines the encoding rules for a new Internet MIME media type called "application/ipp". This  
2710 document also defines the rules for transporting over HTTP a message body whose Content-Type is  
2711 "application/ipp". This document defines the 'ipp' scheme for identifying IPP printers and jobs.

2712 The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to  
2713 implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some  
2714 of the considerations that may assist them in the design of their client and/or IPP object  
2715 implementations. For example, a typical order of processing requests is given, including error  
2716 checking. Motivation for some of the specification decisions is also included.

2717 The "Mapping between LPD and IPP Protocols" document gives some advice to implementers of  
2718 gateways between IPP and LPD (Line Printer Daemon) implementations.

## 2719 **31 Appendix H - Full Copyright Statement (Informative)**

2720 Copyright (C) The Internet Society (1998,1999,2000,2001,2002). All Rights Reserved

2721 This document and translations of it may be copied and furnished to others, and derivative works that  
2722 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published  
2723 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright  
2724 notice and this paragraph are included on all such copies and derivative works. However, this  
2725 document itself may not be modified in any way, such as by removing the copyright notice or  
2726 references to the Internet Society or other Internet organizations, except as needed for the purpose of  
2727 developing Internet standards in which case the procedures for copyrights defined in the Internet  
2728 Standards process must be followed, or as required to translate it into languages other than English.

2729 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or  
2730 its successors or assigns.

2731 This document and the information contained herein is provided on an "AS IS" basis and THE  
2732 INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL  
2733 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY  
2734 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY  
2735 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
2736 PARTICULAR PURPOSE.

2737 **Acknowledgement**

2738

2739

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