

1 Internet Printing Protocol WG  
2 INTERNET-DRAFT  
3 <draft-ietf-ipp-not-spec-08.txt>  
4 Updates RFC 2910 and 2911  
5 [Target Category: standards track]  
6 Expires: May 19, 2002

R. Herriot (editor)  
T. Hastings  
M. Shepherd  
Xerox Corporation  
R. deBry  
Utah Valley State College  
S. Isaacson  
Novell, Inc.  
J. Martin  
Underscore  
R. Bergman  
Hitachi Koki Imaging Solutions  
November 19, 2001

7  
8  
9  
10  
11  
12  
13  
14 Internet Printing Protocol (IPP):  
15 **Event Notifications and Subscriptions**  
16

17 Copyright (C) The Internet Society (2001). All Rights Reserved.

18 Status of this Memo

19 This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of  
20 [RFC2026]. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its  
21 areas, and its working groups. Note that other groups may also distribute working documents as  
22 Internet-Drafts.

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

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

28 **Abstract**

29 This document describes an OPTIONAL extension to the Internet Printing Protocol/1.0 (IPP)  
30 [RFC2566, RFC2565] and IPP/1.1 [RFC2911, RFC2910]. This extension allows a client to subscribe  
31 to printing related Events. Subscriptions are modeled as *Subscription Objects*. The Subscription  
32 Object specifies that when one of the specified *Events* occurs, the Printer sends an asynchronous *Event*  
33 *Notification* to the specified *Notification Recipient* via the specified *Delivery Method* (i.e., protocol). A  
34 client associates Subscription Objects with a particular Job by performing the Create-Job-Subscriptions  
35 operation or by submitting a Job with subscription information. A client associates Subscription Objects  
36 with the Printer by performing a Create-Printer-Subscriptions operation. Four other operations are  
37 defined for Subscription Objects: Get-Subscriptions-Attributes, Get-Subscriptions, Renew-Subscription,  
38 and Cancel-Subscription.  
39

40

40 **Table of Contents**

41	1 Introduction.....	6
42	1.1 Notification Overview.....	6
43	2 Models for Notification.....	8
44	2.1 Model for Notification (Simple Case).....	8
45	2.2 Model for Notification with Cascading Printers.....	9
46	2.3 Distributed Model for Notification.....	9
47	2.4 Extended Notification Recipient.....	10
48	3 Terminology.....	10
49	3.1 Conformance Terminology.....	10
50	3.2 Other Terminology.....	10
51	4 Object Relationships.....	13
52	4.1 Printer and Per-Printer Subscription Objects.....	13
53	4.2 Printer, Job and Per-Job Subscription Objects.....	13
54	5 Subscription Object.....	13
55	5.1 Rules for Support of Subscription Template Attributes.....	14
56	5.2 Rules for Processing Subscription Template Attributes.....	15
57	5.3 Subscription Template Attributes.....	18
58	5.3.1 notify-recipient-uri (uri) OR notify-pull-method (type2 keyword).....	19
59	5.3.1.1 notify-recipient-uri (uri).....	19
60	5.3.1.2 notify-pull-method (type2 keyword).....	20
61	5.3.2 notify-events (1setOf type2 keyword).....	20
62	5.3.2.1 Standard Values for Subscribed Events.....	21
63	5.3.2.1.1 No Events.....	21
64	5.3.2.1.2 Subscribed Printer Events.....	21
65	5.3.2.1.3 Subscribed Job Events.....	22
66	5.3.2.2 Rules for Matching of Subscribed Events.....	23
67	5.3.2.2.1 Rules for Matching of Printer Events.....	23
68	5.3.2.2.2 Rules for Matching of Job Events.....	24
69	5.3.2.2.3 Special Cases for Matching Rules.....	24
70	5.3.3 notify-attributes (1setOf type2 keyword).....	25
71	5.3.4 notify-user-data (octetString(63)).....	26
72	5.3.5 notify-charset (charset).....	27
73	5.3.6 notify-natural-language (naturalLanguage).....	27
74	5.3.7 notify-lease-duration (integer(0:67108863)).....	28
75	5.3.8 notify-time-interval (integer(0:MAX)).....	28
76	5.4 Subscription Description Attributes.....	29
77	5.4.1 notify-subscription-id (integer (1:MAX)).....	30
78	5.4.2 notify-sequence-number (integer (0:MAX)).....	30
79	5.4.3 notify-lease-expiration-time (integer(0:MAX)).....	31

80	5.4.4 notify-printer-up-time (integer(1:MAX)) .....	31
81	5.4.5 notify-printer-uri (uri).....	32
82	5.4.6 notify-job-id (integer(1:MAX)) .....	32
83	5.4.7 notify-subscriber-user-name (name(MAX)) .....	33
84	6 Printer Description Attributes Related to Notification.....	33
85	6.1 printer-state-change-time (integer(1:MAX)).....	33
86	6.2 printer-state-change-date-time (dateTime).....	34
87	7 New Values for Existing Printer Description Attributes.....	34
88	7.1 operations-supported (1setOf type2 enum) .....	34
89	8 Attributes Only in Event Notifications .....	34
90	8.1 notify-subscribed-event (type2 keyword).....	34
91	8.2 notify-text (text(MAX)) .....	35
92	9 Event Notification Content.....	35
93	9.1 Content of Machine Consumable Event Notifications .....	37
94	9.1.1 Event Notification Content Common to All Events.....	38
95	9.1.2 Additional Event Notification Content for Job Events.....	39
96	9.1.3 Additional Event Notification Content for Printer Events.....	39
97	9.2 Content of Human Consumable Event Notification.....	40
98	9.2.1 Event Notification Content Common to All Events.....	40
99	9.2.2 Additional Event Notification Content for Job Events.....	42
100	9.2.3 Additional Event Notification Content for Printer Events.....	42
101	10 Delivery Methods.....	43
102	11 Operations for Notification.....	44
103	11.1 Subscription Creation Operations .....	44
104	11.1.1 Create-Job-Subscriptions Operation .....	45
105	11.1.1.1 Create-Job-Subscriptions Request .....	45
106	11.1.1.2 Create-Job-Subscriptions Response .....	46
107	11.1.2 Create-Printer-Subscriptions operation.....	47
108	11.1.2.1 Create-Printer-Subscriptions Request .....	47
109	11.1.2.2 Create-Printer-Subscriptions Response .....	47
110	11.1.3 Job Creation Operations – Extensions for Notification.....	48
111	11.1.3.1 Job Creation Request .....	48
112	11.1.3.2 Job Creation Response .....	49
113	11.2 Other Operations.....	50
114	11.2.1 Restart-Job Operation – Extensions for Notification.....	50
115	11.2.2 Validate-Job Operation – Extensions for Notification .....	50
116	11.2.3 Get-Printer-Attributes – Extensions for Notification.....	51
117	11.2.4 Get-Subscription-Attributes operation.....	51
118	11.2.4.1 Get-Subscription-Attributes Request .....	51
119	11.2.4.2 Get-Subscription-Attributes Response .....	52

120	11.2.5 Get-Subscriptions operation .....	53
121	11.2.5.1 Get-Subscriptions Request .....	54
122	11.2.5.2 Get-Subscriptions Response .....	55
123	11.2.6 Renew-Subscription operation.....	56
124	11.2.6.1 Renew-Subscription Request .....	56
125	11.2.6.2 Renew-Subscription Response.....	57
126	11.2.7 Cancel-Subscription operation.....	58
127	11.2.7.1 Cancel-Subscription Request .....	58
128	11.2.7.2 Cancel-Subscription Response.....	59
129	12 Conformance Requirements .....	59
130	13 IANA Considerations .....	60
131	13.1 Attribute Registrations .....	60
132	13.2 Additional Enum Attribute Value Registrations for the “operations-supported” Printer Attribute.....	61
133	13.3 Operation Registrations .....	62
134	13.4 Status code Registrations .....	62
135	13.5 Attribute Group tag Registrations.....	63
136	13.6 Registration of Events .....	63
137	13.7 Registration of Event Notification Delivery Methods.....	63
138	13.7.1 Requirements for Registration of Event Notification Delivery Methods .....	63
139	13.7.1.1 Required Characteristics.....	64
140	13.7.1.2 Naming Requirements .....	64
141	13.7.1.3 Functionality Requirements .....	64
142	13.7.1.4 Usage and Implementation Requirements .....	64
143	13.7.1.5 Publication Requirements .....	65
144	13.7.2 Registration Procedure.....	65
145	13.7.2.1 Present the proposal to the Community .....	65
146	13.7.2.2 Delivery Method Reviewer .....	65
147	13.7.2.3 IANA Registration .....	66
148	13.7.3 Delivery Method Document Registrations .....	66
149	13.7.4 Registration Template .....	66
150	14 Internationalization Considerations.....	67
151	15 Security Considerations.....	67
152	16 Status Codes .....	68
153	16.1 successful-ok-ignored-subscriptions (0x0003) .....	68
154	16.2 client-error-ignored-all-subscriptions (0x0414).....	68
155	17 Status Codes in Subscription Attributes Groups .....	68
156	17.1 client-error-uri-scheme-not-supported (0x040C) .....	69
157	17.2 client-error-attributes-or-values-not-supported (0x040B) .....	69
158	17.3 client-error-too-many-subscriptions (0x0415).....	69
159	17.4 successful-ok-too-many-events (0x0005).....	69

160	17.5 successful-ok-ignored-or-substituted-attributes (0x0001) .....	69
161	18 Encodings of Additional Attribute Tags.....	69
162	19 References .....	70
163	20 Author's Addresses.....	71
164	A. Appendix - Model for Notification with Cascading Printers .....	73
165	B. Appendix - Distributed Model for Notification .....	74
166	C. Appendix - Extended Notification Recipient .....	75
167	D. Appendix - Details about Conformance Terminology .....	76
168	E. Appendix - Object Model for Notification .....	76
169	E.1 Appendix - Object relationships .....	77
170	E.2 Printer Object and Per-Printer Subscription Objects.....	77
171	E.3 Job Object and Per-Job Subscription Objects.....	78
172	F. Appendix - Per-Job versus Per-Printer Subscription Objects.....	78
173	G. Appendix - Description of the base IPP documents.....	78
174	H. Appendix - Full Copyright Statement .....	79
175		
176	<b>Tables</b>	
177	Table 1 – Subscription Template Attributes.....	19
178	Table 2 – Subscription Description Attributes .....	30
179	Table 3 – Printer Description Attributes Associated with Notification.....	33
180	Table 4 – Operation-id assignments.....	34
181	Table 5 – Attributes in Event Notification Content .....	38
182	Table 6 – Additional Event Notification Content for Job Events.....	39
183	Table 7 – Combinations of Events and Subscribed Events for “job-impressions-completed” .....	39
184	Table 8 – Additional Event Notification Content for Printer Events.....	40
185	Table 9 – Printer Name in Event Notification Content .....	41
186	Table 10 – Event Name in Event Notification Content.....	41
187	Table 11 – Event Time in Event Notification Content.....	41
188	Table 12 – Job Name in Event Notification Content .....	42
189	Table 13 – Job State in Event Notification Content .....	42
190	Table 14 – Printer State in Event Notification Content .....	43
191	Table 15 – Information about the Delivery Method .....	43
192	Table 16 – Printer Conformance Requirements for Operations .....	60
193		
194	<b>Figures</b>	
195	Figure 1 – Model for Notification.....	9

196	Figure 2 – Model for Notification with Cascading Printers .....	74
197	Figure 3 – Opaque Use of a Notification Service Transparent to the Client.....	75
198	Figure 4 – Use of an Extended Notification Recipient transparent to the Printer .....	76
199	Figure 5 – Object Model for Notification.....	77
200		

## 201 1 Introduction

202 This IPP notification specification is an OPTIONAL extension to Internet Printing Protocol/1.0 (IPP)  
 203 [RFC2566, RFC2565] and IPP/1.1 [RFC2911, RFC2910]. See Appendix G for a description of the  
 204 base IPP documents. This document in combination with the following documents is intended to meet  
 205 the notification requirements described in [ipp-not-req]:

206 Internet Printing Protocol (IPP): “Job Progress Attributes” [ipp-prog]  
 207 One or more Delivery Method Documents registered with IANA (see section 10).  
 208

209 Note: this document does not define any Delivery Methods, but it does define the rules for conformance  
 210 for Delivery Method Documents. Delivery Method Documents are in preparation (see section 10) and  
 211 will be registered with IANA (see section 13.7.3).

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

### 213 1.1 Notification Overview

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

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

- 226 1. The names of Subscribed Events that are of interest to the Notification Recipient.
- 227 2. The address (URL) of one Notification Recipient for a Push Delivery Method or the method for  
 228 a Pull Delivery Method.
- 229 3. The Delivery Method (i.e., the protocol) which the Printer uses to send the Event Notification.

- 230 4. Some opaque data that the Printer sends to the Notification Recipient in the Event Notification.  
231 The Notification Recipient might use this opaque data as a forwarding address for the Event  
232 Notification.
- 233 5. The charset to use in text fields within an Event Notification
- 234 6. The natural language to use in the text fields of the Event Notification
- 235 7. The requested lease time in seconds for the Subscription Object

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

- 238 - **Job Creation operation:** When a client performs such an operation (Print-Job, Print-URI, and  
239 Create-Job), a client can include zero or more Subscription Template Attributes Groups in the  
240 request. The Printer creates one Subscription Object for each Subscription Template  
241 Attributes Group in the request, and the Printer associates each such Subscription Object with  
242 the newly created Job. This document extends these operations' definitions in [RFC2911] by  
243 adding Subscription Template Attributes Groups in the request and Subscription Attributes  
244 Groups in the response.
- 245 - **Create-Job-Subscriptions operation:** A client can include one or more Subscription  
246 Template Attributes Groups in the request. The Printer creates one Subscription Object for  
247 each Subscription Template Attributes Group and associates each with the job that is the target  
248 of this operation.
- 249 - **Create-Printer-Subscriptions operation:** A client can include one or more Subscription  
250 Template Attributes Groups in the request. The Printer creates one Subscription Object for  
251 each Subscription Template Attributes Group and associates each with the Printer that is the  
252 target of this operation.

253 For each of the above operations:

- 254 - the Printer associates a Subscription Object with the Printer or a specific Job. When a  
255 Subscription Object is associated with a Job Object, it is called a *Per-Job Subscription Object*.  
256 When a Subscription Object is associated with a Printer Object, it is called a *Per-Printer*  
257 *Subscription Object*.
- 258 - the response contains one Subscription Attributes Group for each Subscription Template  
259 Attributes Group in the request and in the same order. When the Printer successfully creates a  
260 Subscription Object, its corresponding Subscription Attributes Group contains the “notify-  
261 subscription-id” attribute. This attribute uniquely identifies the Subscription Object and is  
262 analogous to a “job-id” for a Job object. Some operations described below use the “notify-  
263 subscription-id” to identify the target Subscription Object.

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

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

- 267 - **Validate-Job operation:** When a client performs this operation, a client can include zero or  
268 more Subscription Template Attributes Groups in the request. The Printer determines if it  
269 could create one Subscription Object for each Subscription Template Attributes Group in the  
270 request. This document extends this operation's definition in [RFC2911] by adding  
271 Subscription Template Attributes Groups in the request and Subscription Attributes Groups in  
272 the response.
- 273 - **Get-Subscription-Attributes operation:** This operation allows a client to obtain the specified  
274 attributes of a target Subscription Object.
- 275 - **Get-Subscriptions operation:** This operation allows a client to obtain the specified attributes  
276 of all Subscription Objects associated with the Printer or a specified Job.
- 277 - **Renew-Subscription operation:** This operation renews the lease on the target Per-Printer  
278 Subscription Object before it expires. A newly created Per-Printer Subscription Object receives  
279 an initial lease. It is the duty of the client to use this operation frequently enough to preserve a  
280 Per-Printer Subscription Object. The Printer deletes a Per-Printer Subscription Object when its  
281 lease expires. A Per-Job Subscription Object last exactly as long as its associated Job Object  
282 and thus doesn't have a lease.
- 283 - **Cancel-Subscription operation:** This operation (1) cancels the lease on the specified Per-  
284 Printer Subscription Object and thereby deletes the Per-Printer Subscription Object or (2)  
285 deletes the Per-Job Subscription Object.

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

- 288 a) generates an Event Notification with information specified in section 9, AND  
289 b) either:
- 290 i) If the Delivery Method is a Push Delivery Method as indicated by the presence of the  
291 Subscription Object's "notify-recipient-uri" attribute, delivers the Event Notification  
292 using the Delivery Method and target address identified in the Subscription Object's  
293 "notify-recipient-uri" attribute, OR
- 294 ii) If the Delivery Method is a Pull Delivery Method as indicated by the presence of the  
295 Subscription Object's "notify-pull-method" attribute, saves Event Notification for a time  
296 period called the Event Life defined by the Delivery Method, i.e., the Notification  
297 Recipient is expected to fetch the Event Notifications.

## 298 **2 Models for Notification**

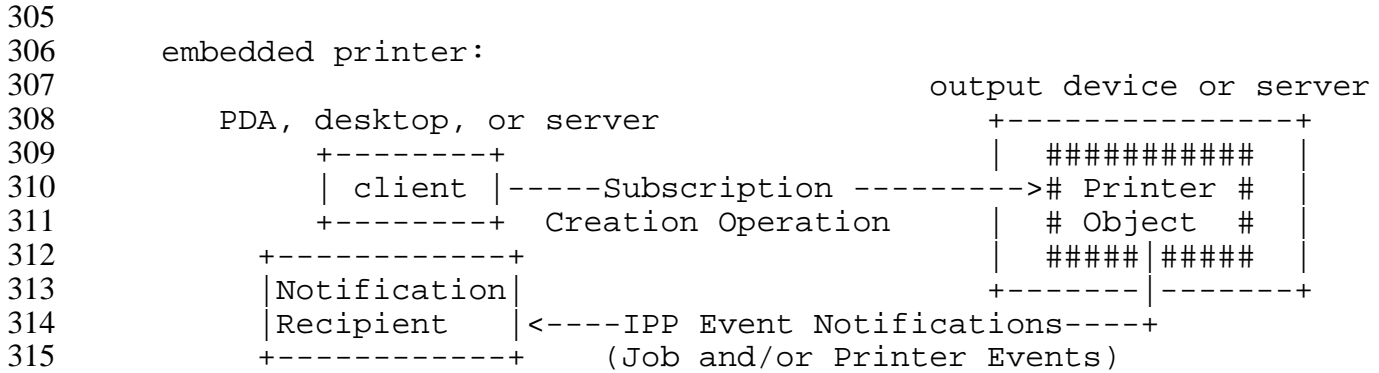
### 299 **2.1 Model for Notification (Simple Case)**

300 As part of a Subscription Creation Operation, an IPP Printer (i.e., located in an output device or a  
301 server) creates one or more Subscription Objects. In a Subscription Creation Operation, the client



302 specifies the Notification Recipient to which the Printer is to deliver Event Notifications. A Notification  
 303 Recipient can be the Subscribing Client or a third party.

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



316 **Figure 1 – Model for Notification**

## 317 2.2 Model for Notification with Cascading Printers

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

- 321 - directly to the Notification Recipient specified by the Subscribing Client or
- 322 - via the Print Server to the Notification Recipient specified by the Subscribing Client.

323 See Appendix A for more details.

## 324 2.3 Distributed Model for Notification

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

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

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

335 Appendix B describes this example in detail.

## 336 2.4 Extended Notification Recipient

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

341 Appendix C describes this example in detail.

## 342 3 Terminology

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

### 345 3.1 Conformance Terminology

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

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

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

### 358 3.2 Other Terminology

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

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

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

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

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

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

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

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

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

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

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

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

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

387 **Notification Recipient** – the entity to which the Printer sends an Event Notification.

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

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

393 **Push Delivery Method** –The Printer sends the Event Notification shortly after an Event occurs. For  
394 some Push Delivery Methods, the Notification Recipient MUST send a response; for others it  
395 MUST NOT send a response.

396 **Pull Delivery Method** – The Printer saves Event Notifications for some event life time and expects the  
397 Notification Recipient to request Event Notifications. The Printer returns the Event Notifications in  
398 a response to such a request.

- 399       **Event Life** – For a Pull Delivery Method, the length of time in seconds after an Event occurs during  
400               which the Printer will return that Event in response to a request for Event Notifications. After  
401               the Event Life expires, the Printer will no longer return an Event Notification for that Event in  
402               such a response.
- 403       **Subscription Object** – An object containing a set of attributes that indicate: the Notification Recipient,  
404               the Delivery Method, the Subscribed Events that cause the Printer to send an Event Notification,  
405               and the information to send in an Event Notification.
- 406       **Per-Job Subscription Object** – A Subscription Object that is associated with a single Job. The Create-  
407               Job-Subscriptions operation and Job Creation operations create such an object.
- 408       **Per-Printer Subscription Object** – A Subscription Object that is associated with the Printer as a  
409               whole. The Create-Printer-Subscriptions operation creates such an object.
- 410       **Subscribing Client** – The client that creates the Subscription Object.
- 411       **Subscription Creation Operation** – An operation that creates a Subscription Object: Job Creation  
412               operations, Create-Job-Subscriptions operation, Create-Printer-Subscriptions operation. In the  
413               context of a Job Creation operation, a Subscription Creation Operation is the part of the Job  
414               Creation operation that creates a Subscription object. The Restart-Job operation [RFC2911] is not  
415               considered a Subscription Creation Operation, since the Printer re-uses the Job's existing  
416               Subscription Objects, rather than creating any new Subscription Objects.
- 417       **Subscription Creation Request** – The request portion of a Subscription Creation Operation.
- 418       **Subscription Template Attributes** – Subscription Object attributes that a client can supply in a  
419               Subscription Creation Operation and associated Printer Object attributes that specify supported and  
420               default values for the Subscription Object attributes.
- 421       **Subscription Description Attributes** – Subscription Object attributes that a Printer supplies during a  
422               Subscription Creation Operation.
- 423       **Subscription Template Attributes Group** – The attributes group in a request that contains  
424               Subscription Object attributes that are Subscription Template Attributes.
- 425       **Subscription Attributes Group** – The attributes group in a response that contains Subscription Object  
426               attributes.
- 427       **Human Consumable Event Notification** – localized text for human consumption only. There is no  
428               standardized format and thus programs should not try to parse this text.
- 429       **Machine Consumable Event Notification** – bytes for program consumption. The bytes are formatted  
430               according to the Delivery Method document.
- 431       **Printer** – the software that supports an output device or print server (see IPP/1.1 [RFC2911] which  
432               uses the terms Printer and Printer object interchangeably). This document extends the IPP/1.1

433 Printer definition to include the software that implements Subscription Creation Operations and the  
434 sending of Event Notifications, even if the software for such a Printer would be distributed across a  
435 network (see section 2.3).

436 **Notification** – when not in the phrases ‘Event Notification’ and ‘Notification Recipient’ — the  
437 concepts of this specification, i.e., Events, Subscription Objects, and Event Notifications.

## 438 **4 Object Relationships**

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

### 441 **4.1 Printer and Per-Printer Subscription Objects**

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

### 444 **4.2 Printer, Job and Per-Job Subscription Objects**

- 445 1. A Printer object is associated with zero or more Job objects.
- 446 2. Each Job object is associated with exactly one Printer object.
- 447 3. A Job object is associated with zero or more Per-Job Subscription Objects.
- 448 4. Each Per-Job Subscription Object is associated with exactly one Job object.

## 449 **5 Subscription Object**

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

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

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

- 458 1. Subscription Object attributes that a client can supply in a Subscription Creation Request and

- 459           2. their associated Printer Object attributes that specify supported and default values for the  
460           Subscription Object attributes

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

## 463 **5.1 Rules for Support of Subscription Template Attributes**

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

467           Subscription Objects attributes that are Subscription Template Attributes conform to the following  
468           rules:

- 469           1. Each attribute's name starts with the prefix string "notify-" and this document calls such  
470           attributes "notify-xxx".
- 471           2. For each "notify-xxx" Subscription Object attribute defined in column 1 of Table 1 in section 5.3,  
472           Table 1 specifies corresponding Printer attributes: "notify-xxx-default", "notify-xxx-supported",  
473           "yyy-supported" and "notify-max-xxx-supported" defined in column 2 of Table 1. Note "xxx"  
474           stands for the same string in each case and "yyy" stands for some other string.
- 475           3. If a Printer supports "notify-xxx" in column 1 of Table 1, then the Printer MUST support all  
476           associated attributes specified in column 2 of Table 1. For example, Table 1 shows that if the  
477           Printer supports "notify-events", it MUST support "notify-events-default", "notify-events-  
478           supported" and "notify-max-events-supported".
- 479           4. If a Printer does not support "notify-xxx" in column 1 of Table 1, then the Printer MUST NOT  
480           support any associated "notify-yyy" attributes specified in column 2 of Table 1. For example,  
481           Table 1 shows that if the Printer doesn't support "notify-events", it MUST NOT support "notify-  
482           events-default", "notify-events-supported" and "notify-max-events-supported". Note this rule  
483           does not apply to attributes whose names do not start with the string "notify-" and are thus  
484           defined in another object and used by other attributes.
- 485           5. Most "notify-xxx" attributes have a corresponding "yyy-supported" attribute that specifies the  
486           supported values for "notify-xxx". Column 2 of Table 1 specifies the name of each "yyy-  
487           supported" attribute. The naming rules of IPP/1.1 (see [RFC2911]) are used when "yyy-  
488           supported" is "notify-xxx-supported".
- 489           6. Some "notify-xxx" attributes have a corresponding "notify-xxx-default" attribute that specifies  
490           the value for "notify-xxx" if the client does not supply it. Column 2 of Table 1 specifies the name  
491           of each "notify-xxx-default" attribute. The naming rules of IPP/1.1 (see [RFC2911]) are used.

492           If a client wishes to present an end user with a list of supported values from which to choose, the client  
493           SHOULD query the Printer for its supported value attributes. The client SHOULD also query the

494 default value attributes. If the client then limits selectable values to only those values that are  
495 supported, the client can guarantee that the values supplied by the client in the create request all fall  
496 within the set of supported values at the Printer. When querying the Printer, the client MAY enumerate  
497 each attribute by name in the Get-Printer-Attributes Request, or the client MAY just supply the  
498 ‘subscription-template’ group name in order to get the complete set of supported attributes (both  
499 supported and default attributes – see section 11.2.3).

## 500 5.2 Rules for Processing Subscription Template Attributes

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

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

- 508 1. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer supports it and  
509 its value, the Printer MUST populate the attribute on the created Subscription Object.
- 510 2. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer doesn’t support  
511 it or its value, the Printer MUST NOT populate the attribute on the created Subscription Object  
512 with it. The Printer MUST do one of the following:
  - 513 a) If the value of the “notify-xxx” attribute is unsupported, the Printer MUST return the attribute  
514 with its value in the Subscription Attributes Group of the response.
  - 515 b) If “notify-xxx” is an unsupported attribute, the Printer MUST return the attribute in the  
516 Subscription Attributes Group of the response with the ‘unsupported’ out-of-band value.

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

- 521 3. If a client is REQUIRED to supply a “notify-xxx” attribute from column 1 of Table 1 and the Printer  
522 doesn’t support the supplied value, the Printer MUST NOT create a Subscription Object. The rules  
523 for Unsupported Attributes in step #2 still apply.
- 524 4. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 and the attribute is  
525 REQUIRED for the client to supply, the Printer MUST reject the Subscription Creation Operation  
526 (including Job Creation operations) without creating a Subscription Object, and MUST return in the  
527 response:
  - 528 c) the status code ‘client-error-bad-request’ AND

- 529           d) no Subscription Attribute Groups.
- 530           5. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 that is OPTIONAL for  
531           the client to supply, and column 2 of Table 1 either:
- 532           a) specifies a “notify-xxx-default” attribute, the Printer MUST behave as if the client had supplied  
533           the “notify-xxx-default” attribute (see step #1) and populate the Subscription object with the  
534           value of the “notify-xxx-default” attribute as part of the Subscription Creation operation (unlike  
535           Job Template attributes where the Printer does not populate the Job object with defaults – see  
536           [RFC2911]) OR
- 537           b) does not specify a “notify-xxx-default” attribute, the Printer MUST populate the “notify-xxx”  
538           attribute on the Subscription Object according to the definition of the “notify-xxx” attribute in a  
539           section 5.3. For some attributes, the “notify-xxx” is populated with the value of some other  
540           attribute, and for others, the “notify-xxx” is NOT populated on the Subscription object at all.
- 541           6. A Printer MUST create a Subscription Object for each Subscription Template Attributes group in a  
542           request unless the Printer:
- 543           a) encounters some attributes in a Subscription Template Attributes Group that require the Printer  
544           not to create the Subscription Object OR
- 545           b) would create a Per-Job Subscription Object when it doesn’t have space for another Per-Job  
546           Subscription Object OR
- 547           c) would create a Per-Printer Subscription Object when it doesn’t have space for another Per-  
548           Printer Subscription Object.
- 549           7. A response MUST contain one Subscription Attributes Group for each Subscription Template  
550           Attributes Group in the request (and in the same order) whether the Printer creates a Subscription  
551           Object from the Subscription Template Attributes Group or not. However, the attributes in each  
552           Subscription Attributes Group can be in any order.
- 553           8. The Printer MUST populate each Subscription Attributes Group of the response such that each  
554           contains:
- 555           a) the “notify-subscription-id” attribute (see section 5.4.1), if and only if the Printer creates a  
556           Subscription Object.
- 557           b) the “notify-lease-duration” attribute (see section 5.3.7), if and only if the Printer creates a Per-  
558           Printer Subscription Object. The value of this attribute is the value of the Subscription Object’s  
559           “notify-lease-duration” attribute. This value MAY be different from the client-supplied value  
560           (see section 5.3.7). If a client supplies this attribute in the creation of a Per-Job Subscription  
561           Object, it MUST appear in this group with the out-of-band value ‘unsupported’ to indicate that  
562           the Printer doesn’t support it in this context.



- 563 c) all of the unsupported Subscription Template Attributes from step #2. Note, they are not  
564 returned in the Unsupported Attributes Group in order to separate the unsupported attributes  
565 for each Subscription Object.
- 566 d) the “notify-status-code” attribute if the Printer does not create the Subscription Object or if  
567 there are unsupported attributes from step #2. The possible values of the “notify-status-code”  
568 attribute are shown below (see section 17 for more details). The Printer returns the first value in  
569 the list below that describes the status.
- 570 ‘client-error-uri-scheme-not-supported’: the Subscription Object was not created because  
571 the scheme of the “notify-recipient-uri” attribute is not supported. See section 17.1 for  
572 more details about this status code. See step #3 in this section for the case that causes  
573 this error, and the resulting step #6a) that causes the Printer not to create the  
574 Subscription Object.
- 575 ‘client-error-attributes-or-values-not-supported’: the Subscription Object was not created  
576 because the method of the “notify-pull-method” attribute is not supported. See section  
577 17.1 for more details about this status code. See step #3 in this section for the case that  
578 causes this error, and the resulting step #6a) that causes the Printer not to create the  
579 Subscription Object.
- 580 ‘client-error-too-many-subscriptions’: the Subscription Object was not created because the  
581 Printer has no space for additional Subscription Objects. The client SHOULD try again  
582 later. See section 17.3 for more details about this status code. See steps #6b) and #6c) in  
583 this section for the cases that causes this error.
- 584 ‘successful-ok-too-many-events’: the Subscription Object was created without the “notify-  
585 events” values included in this Subscription Attributes Group because the “notify-events”  
586 attribute contains too many values. See section 17.4 for more details about this status  
587 code. See step #2 in this section and section 5.3.2 for the cases that cause this status  
588 code.
- 589 ‘successful-ok-ignored-or-substituted-attributes’: the Subscription Object was created but  
590 some supplied Subscription Template Attributes are unsupported. These unsupported  
591 attributes are also in the Subscription Attributes Group. See section 17.5 for more details  
592 about this status code. See step #2 in this section for the cases that cause this status  
593 code.
- 594 9. The Printer MUST validate all Subscription Template Attributes and MUST return all unsupported  
595 attributes and values in the corresponding Subscription Attributes Group of the response (see step  
596 #2) unless it determines that it could not create additional Subscription Objects because of condition  
597 #6b) or condition #6c). Then, the Printer NEED NOT validate these additional Subscription  
598 Template Attributes and the client MUST NOT expect to find unsupported attributes from step #2  
599 in such additional Subscription Attribute Groups.

### 600 5.3 Subscription Template Attributes

601 This section contains the Subscription Template Attributes defined for the Subscription and Printer  
602 objects.

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

- 604 - **Attribute in Subscription Object:** the name and attribute syntax of each Subscription Object  
605 Attribute that is a Subscription Template Attribute
- 606 - **Default and Supported Printer Attributes:** the default attribute and supported Printer  
607 attributes that are associated with the attribute in column 1.

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

610 For Push Delivery Methods, a Printer MUST support all attributes in Table 1 below except for “notify-  
611 pull-method” and “notify-attributes” (and “notify-pull-method-supported” and “notify-attributes-  
612 supported”). For Pull Delivery Methods, a Printer MUST support all attributes in Table 1 below except  
613 for “notify-recipient-uri” and “notify-attributes” (and “notify-schemes-supported” and “notify-attributes-  
614 supported”). If a Printer supports both Push and Pull Delivery Methods, then it MUST support both  
615 “notify-recipient-uri” and “notify-pull-method” attributes.

616 For Pull Delivery Methods, a client MUST supply “notify-recipient-uri” and MAY omit any of the rest  
617 of the attributes in column 1 of Table 1 in a Subscription Creation Request. For Push Delivery  
618 Methods, a client MUST supply “notify-pull-method” and MAY omit any of the rest of the attributes in  
619 column 1 of Table 1 in a Subscription Creation Request. A client MUST NOT supply both “notify-  
620 recipient-uri” and “notify-pull-method” attributes in the same Subscription Creation Request.

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

627

**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))	

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

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

630 **5.3.1 notify-recipient-uri (uri) OR notify-pull-method (type2 keyword)**

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

633 **5.3.1.1 notify-recipient-uri (uri)**

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

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

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

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

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

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

### 650 **5.3.1.2 notify-pull-method (type2 keyword)**

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

652 If a Printer supports any Pull Delivery Methods, a Printer MUST support this attribute and return the  
653 value as supplied by the client in any operation response that includes this attribute.

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

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

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

### 661 **5.3.2 notify-events (1setOf type2 keyword)**

662 This attribute contains a set of Subscribed Events. When an Event occurs and it “matches” a value of  
663 this attribute, the Printer sends an Event Notification using information in the Subscription Object. The  
664 details of “matching” are described subsection 5.3.2.2.

665 A Printer MUST support this attribute.

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

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

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

### 677 5.3.2.1 Standard Values for Subscribed Events

678 Each value of this attribute is a keyword and it specifies a Subscribed Event that represents certain  
679 changes. Some keywords represent a subset of changes of another keyword, e.g., ‘job-completed’ is an  
680 Event value which is a sub-value of ‘job-state-change’. See section 5.3.2.2 for the case where this  
681 attribute contains both a value and a sub-value.

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

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

#### 685 5.3.2.1.1 No Events

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

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

#### 694 5.3.2.1.2 Subscribed Printer Events

695 The standard keyword values for Subscribed Printer Events are:

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

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

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

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

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

707 **‘printer-config-changed’**: OPTIONAL – when the configuration of a Printer has changed, i.e., the  
708 value of the “printer-message-from-operator” or any “configuration” Printer attribute has changed.

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

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

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

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

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

### 735 5.3.2.1.3 Subscribed Job Events

736 The standard keyword values for Subscribed Job Events are:

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

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

744 **‘job-created’**: REQUIRED – the Printer has accepted a Job Creation operation, a Restart-  
745 Job operation [RFC2911], or any job operation that creates a Job object from an existing  
746 Job object. The Printer sets the job’s “time-at-creation” attribute value (see [RFC2911])

747 section 4.3.14.1). The Printer puts the job in the ‘pending’, ‘pending-held’ or  
748 ‘processing’ states.

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

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

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

770 **‘job-progress’**: OPTIONAL – when the Printer has completed Printing a sheet. See the separate [ipp-  
771 prog] specification for additional attributes that a Printer MAY send in an Event Notification  
772 caused by this Event. The “notify-time-interval” attribute affects this Event by causing the Printer  
773 NOT to send an Event Notification every time a ‘job-progress’ Events occurs. See section 5.3.8 for  
774 full details.

### 775 5.3.2.2 Rules for Matching of Subscribed Events

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

#### 779 5.3.2.2.1 Rules for Matching of Printer Events

780 Suppose that the Printer causes Printer Event E to occur. For each Per-Job or Per-Printer Subscription  
781 S in the Printer, if E equals a value of this attribute in S or E is a sub-value of a value of this attribute in  
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 Object  
785 B is a Per-Job Subscription Object for Job 1, and Subscription Object C is a Per-Job Subscription  
786 Object for Job 2. When the Printer enters the 'stopped' state, the Printer sends an Event Notification to  
787 the Notification Recipients of Subscription Objects A, B, and C because this is a Printer Event. Note if  
788 Job 1 has already completed, the Printer would not send an Event Notification for its Subscription  
789 Object, even if Job 1 is retained in the Job Retention and/or the Job History phases (see [RFC2911]  
790 section 4.3.7.1).

#### 791 **5.3.2.2.2 Rules for Matching of Job Events**

792 Suppose 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 or  
796 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 attribute  
799 in S or E is a sub-value of a value of this attribute in, the Printer MUST NOT generate an Event  
800 Notification from S.

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

#### 812 **5.3.2.2.3 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 send  
815 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 send both Event  
817 Notifications. That is, the Printer MUST NOT try to consolidate seemingly identical Event Notifications



818 that occur in separate Subscription objects. Incidentally, the Printer MUST NOT reject Subscription  
819 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 send one Event Notification for each matched value in the  
822 Subscription Object or it MAY send only one Event Notification per Subscription Object. The rules in  
823 sections 5.3.2.2.1 and 5.3.2.2.2 are purposefully ambiguous about the number of Event Notification  
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 the  
827 Subscribed Job Events ‘job-state-changed’ and ‘job-completed’. The Printer sends 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 sends either one or two Event  
830 Notifications to the Notification Recipient of Subscription Object B, depending on implementation. If it  
831 sends two Event Notifications, one has the value of ‘job-state-changed’ for the “notify-subscribing-  
832 event” attribute, and the other has the value of ‘job-completed’ for the “notify-subscribing-  
833 attribute. If it sends one Event Notification, it has the value of either ‘job-state-changed’ or ‘job-  
834 completed’ for the “notify-subscribing-event” attribute, depending on implementation. The algorithm for  
835 choosing such a value is implementation dependent.

### 836 5.3.3 notify-attributes (1setOf type2 keyword)

837 This attribute contains a set of attribute names. When a Printer sends 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. The “notify-attributes-supported”  
848 MAY contain any Printer attribute, Job attribute or Subscription Object attribute that the Printer  
849 supports in an Event Notification. It MUST NOT contain any of the attributes in Section 9.1 that a  
850 Printer automatically puts in an Event Notification; it would be redundant. If a client supplies an  
851 attribute in Section 9.1, the Printer MUST treat it as an unsupported attribute value of the “notify-  
852 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 notify-user-data (octetString(63))**

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

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

878 A Printer MUST support this attribute.

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

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

### 886 **5.3.5 notify-charset (charset)**

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

889 A Printer MUST support this attribute.

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

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

### 899 **5.3.6 notify-natural-language (naturalLanguage)**

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

903 A Printer MUST support this attribute.

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

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

### 913 **5.3.7 notify-lease-duration (integer(0:67108863))**

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

918 This attribute is not present on a Per-Job Subscription Object because the Subscription Object lasts  
919 exactly as long as the associated Job object. See discussion of the ‘job-completed’ event in section  
920 5.3.2.1.3 about retention of the Job object after completion.

921 A Printer MUST support this attribute.

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

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

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

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

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

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

### 943 **5.3.8 notify-time-interval (integer(0:MAX))**

944 The ‘job-progress’ Event occurs each time that a Printer completes a sheet. Some Notification  
945 Recipients do not want to receive an Event Notification every time this Event occurs. This attribute  
946 allows a Subscribing Client to request how often it wants to receive Event Notifications for ‘job-

947 progress' Events. The value of this attribute MAY be any nonnegative integer (0,MAX) indicating the  
948 minimum number of seconds between 'job-progress' Event Notifications.

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

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

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

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

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

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

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

## 973 **5.4 Subscription Description Attributes**

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

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

977 A client MUST NOT supply the attributes in Table 2 in a Subscription Template Attributes Group of a  
978 Subscription Creation Operation. If the client supplies them, the Printer MUST NOT set them and

979 MUST treat them as unsupported attributes. There are no corresponding default or supported  
 980 attributes.

981 **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))

982

983 **5.4.1 notify-subscription-id (integer (1:MAX))**

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

986 A Printer MUST support this attribute.

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

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

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

995 **5.4.2 notify-sequence-number (integer (0:MAX))**

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

1000 A Printer MUST support this attribute.

1001 When the Printer creates a Subscription Object, it MUST set the value of this attribute to 0. This value  
 1002 indicates that the Printer has not sent any Event Notifications for this Subscription Object.

1003 Each time the Printer sends a newly generated Event Notification, it MUST increase the value of this  
1004 attribute by 1. For some Delivery Methods, the Printer MUST include this attribute in each Event  
1005 Notification, and the value MUST be the value after it is increased by 1. That is, the value of this  
1006 attribute in the first Event Notification after Subscription object creation MUST be 1, the second  
1007 MUST be 2, etc. If a Delivery Method is defined such that the Notification Recipient returns a  
1008 response, the Printer can re-try sending an Event Notification a certain number of times with the same  
1009 sequence number when the Notification Recipient fails to return a response.

1010 If a Subscription Object lasts long enough to reach the value of MAX, its next value MUST be 0, i.e., it  
1011 wraps.

### 1012 **5.4.3 notify-lease-expiration-time (integer(0:MAX))**

1013 This attribute specifies the time in the future when the lease on the Per-Printer Subscription Object will  
1014 expire, i.e. the “printer-up-time” value at which the lease will expire. If the value is 0, the lease never  
1015 expires.

1016 A Printer MUST support this attribute.

1017 When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present – the  
1018 Subscription Object lasts exactly as long as the associated Job object. See also the discussion of the  
1019 ‘job-completed’ event in section 5.3.2.1.3 about retention of the Job object after completion so that a  
1020 Notification Recipient can query the Job object after receiving the ‘job-completed’ Event Notification.

1021 When the Printer creates a Per-Printer Subscription Object, it populates this attribute with a value that is  
1022 the sum of the values of the Printer’s “printer-up-time” attribute and the Subscription Object’s “notify-  
1023 lease-duration” attribute with the following exception. If the value of the Subscription Object’s “notify-  
1024 lease-duration” attribute is 0 (i.e., no expiration time), then the value of this attribute MUST be set to 0  
1025 (i.e., no expiration time).

1026 When the Printer powers up, it MUST set the value of this attribute in each persistent Subscription  
1027 Object using the algorithm in the previous paragraph.

1028 When the “printer-up-time” equals the value of this attribute, the Printer MUST delete the Subscription  
1029 Object. A client can extend a lease of a Per-Printer Subscription Object with the Renew-Subscription  
1030 operation (see section 11.2.6).

1031 Note: In order to compute the number of seconds remaining in a lease for a Per-Printer Subscription  
1032 Object, a client can subtract the Subscription’s “notify-printer-up-time” attribute (see section 5.4.4)  
1033 from the Subscription’s “notify-lease-expiration-time” attribute.

### 1034 **5.4.4 notify-printer-up-time (integer(1:MAX))**

1035 This attribute is an alias for the Printer’s “printer-up-time” attribute “ (see [RFC2911] section 4.4.29).  
1036 In other words, when this attribute is queried with the Get-Subscriptions or Get-Subscription-Attributes

1037 operations (see sections 11.2.4 and 11.2.5), the value returned is the current value of the Printer's  
1038 "printer-up-time" attribute, rather than the time at which the Subscription Object was created.

1039 A Printer MUST support this attribute.

1040 When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present. When  
1041 the Printer creates a Per-Printer Subscription Object, this attribute MUST be present.

1042 Note: this attribute exists in a Per-Printer Subscription Object so that a client using the Get-  
1043 Subscription-Attributes or Get-Subscription operations can convert the Per-Printer Subscription's  
1044 "notify-lease-expiration-time" attribute to wall clock time with one request. If the value of the "notify-  
1045 lease-expiration-time" attribute is not 0 (i.e., no expiration time), then the difference between the  
1046 "notify-lease-expiration-time" attribute and the "notify-printer-up-time" is the remaining number of  
1047 seconds on the lease from the current time.

#### 1048 **5.4.5 notify-printer-uri (uri)**

1049 This attribute identifies the Printer object that created this Subscription Object.

1050 A Printer MUST support this attribute.

1051 During a Subscription Creation Operation, the Printer MUST populate this attribute with the value of  
1052 the "printer-uri" operation attribute in the request. From the Printer URI, the client can, for example,  
1053 determine what security scheme was used.

#### 1054 **5.4.6 notify-job-id (integer(1:MAX))**

1055 This attribute specifies whether the containing Subscription Object is a Per-Job or Per-Printer  
1056 Subscription Object, and for Per-Job Subscription Objects, it specifies the associated Job.

1057 A Printer MUST support this attribute.

1058 If this attribute is not present, the Subscription Object MUST be a Per-Printer Subscription. If this  
1059 attribute is present, the Subscription Object MUST be a Per-Job Subscription Object and this attribute  
1060 MUST identify the Job with which the Subscription Object is associated.

1061 Note: This attribute could be useful to a Notification Recipient that receives an Event Notification  
1062 generated from a Per-Job Subscription Object and caused by a Printer Event. The Event Notification  
1063 gives access to the Printer and the Subscription Object. The Event Notification gives access to the  
1064 associated Job only via this attribute. See discussion of the 'job-completed' event in section 5.3.2.1.3  
1065 about retention of the Job object after completion so that a Notification Recipient can query the Job  
1066 object after receiving the 'job-completed' Event Notification.



### 1067 **5.4.7 notify-subscriber-user-name (name(MAX))**

1068 This attribute contains the name of the user who performed the Subscription Creation Operation.

1069 A Printer **MUST** support this attribute.

1070 The Printer sets this attribute to the most authenticated printable name that it can obtain from the  
 1071 authentication service over which the Subscription Creation Operation was received. The Printer uses  
 1072 the same mechanism for determining the value of this attribute as it does for a Job's "job-originating-  
 1073 user-name" (see [RFC2911] section 4.3.6).

1074 Note: To help with authentication, a Subscription Object may have additional private attributes about  
 1075 the user, e.g., a credential of a principal. Such private attributes are implementation-dependent and not  
 1076 defined in this document.

## 1077 **6 Printer Description Attributes Related to Notification**

1078 This section defines the Printer Description attributes that are related to Notification. Table 3 lists the  
 1079 Printer Description attributes, indicates the Printer support required for conformance, and whether or  
 1080 not the attribute is READ-ONLY (see section 3.1):

1081 **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

1082

### 1083 **6.1 printer-state-change-time (integer(1:MAX))**

1084 This **OPTIONAL** attribute records the most recent time at which the 'printer-state-changed' Printer  
 1085 Event occurred whether or not any Subscription objects were listening for this event. This attribute  
 1086 helps a client or operator to determine how long the Printer has been in its current state.

1087 A Printer **MAY** support this attribute and if so, the attribute **MUST** be READ-ONLY.

1088 On power-up, the Printer **MUST** set the value of this attribute to be the value of its "printer-up-time"  
 1089 attribute, so that it always has a value. Whenever the 'printer-state-changed' Printer Event occurs, the  
 1090 Printer **MUST** set this attribute to the value of the Printer's "printer-up-time" attribute.

## 1091 6.2 printer-state-change-date-time (dateTime)

1092 This OPTIONAL attribute records the most recent time at which the ‘printer-state-changed’ Printer  
1093 Event occurred whether or not there were any Subscription Objects listening for this event. This  
1094 attribute 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 set the value of this attribute to be the value of its “printer-current-  
1097 time” attribute, so that it always has a value (see [RFC2911] section 4.4.30 on “printer-current-time”).  
1098 Whenever the ‘printer-state-changed’ Printer Event occurs, the Printer MUST set this attribute to the  
1099 value of the Printer’s “printer-current-time” attribute.

## 1100 7 New Values for Existing Printer Description Attributes

1101 This section contains those attributes for which additional values are added.

### 1102 7.1 operations-supported (1setOf type2 enum)

1103 The following “operation-id” values are added in order to support the new operations defined in this  
1104 document:

1105 **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

## 1106 8 Attributes Only in Event Notifications

1107 This section contains those attributes that exist only in Event Notifications and do not exist in any  
1108 objects.

### 1109 8.1 notify-subscribed-event (type2 keyword)

1110 This attribute indicates the Subscribed Event that caused the Printer to send this Event Notification.  
1111 This attribute exists only in Event Notifications.

1112 This attribute MUST contain one of the values of the “notify-events” attribute in the Subscription  
1113 Object, i.e., one of the Subscribed Event values. Its value is the Subscribed Event that “matches” the  
1114 Event that caused the Printer to send this Event Notification. This Subscribed Event value may be  
1115 identical to the Event or the Event may be a sub-value of the Subscribed Event. For example, the ‘job-  
1116 completed’ Event (which is a sub-event of the ‘job-state-changed’ event) would cause the Printer to  
1117 send an Event Notification for either the ‘job-completed’ or ‘job-state-changed’ Subscribed Events and  
1118 to send the ‘job-completed’ or ‘job-state-changed’ value for this attribute, respectively,. See section  
1119 5.3.2.2 for the “matching” rules of Subscribed Events and for additional examples.

1120 The Delivery Method Document specifies whether the Printer includes the value of this attribute in an  
1121 Event Notification.

## 1122 **8.2 notify-text (text(MAX))**

1123 This attribute contains a Human Consumable text message (see section 9.2). This message describes the  
1124 Event and is encoded as plain text, i.e., ‘text/plain’ with the charset specified by Subscription Object’s  
1125 “notify-charset” attribute.

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

## 1128 **9 Event Notification Content**

1129 This section defines the Event Notification content that the Printer sends when an Event occurs.

1130 When an Event occurs, the Printer MUST find each Subscription object whose “notify-events” attribute  
1131 “matches” the Event. See section 5.3.2.2 for details on “matching”. For each matched Subscription  
1132 Object, the Printer MUST create an Event Notification with the content and format that the Delivery  
1133 Method Document specifies. The content contains the value of attributes specified by the Delivery  
1134 Method Document. The Printer obtains the values immediately after the Event occurs. For example, if  
1135 the “printer-state” attribute changes from ‘idle’ to ‘processing’, the Event ‘printer-state-changed’  
1136 occurs and the Printer puts various attributes into the Event Notification, including “printer-up-time”  
1137 and “printer-state” with the values that they have immediately after the Event occurs, i.e., the value of  
1138 “printer-state” is ‘processing’.

### 1139 **Event Notification Ordering:**

1140 When a Printer sends Event Notifications, the Event Notifications from any given Subscription Object  
1141 MUST be in time stamp order, i.e., in order of increasing “printer-up-time” attribute value in the Event  
1142 Notification (see Table 5). These Event Notifications MAY be interleaved with those from other  
1143 Subscription Objects, as long as those others are also in time stamp order. The Printer MUST observe  
1144 these ordering requirements whether sending multiple pending Events as multiple separate Event  
1145 Notifications or together in a single Compound Event Notification.

1146 If a Subscribing Client wants the Printer to send certain Event Notifications in time stamp order, the  
1147 Subscribing Client uses a single Subscription Object. Even so, depending on the underlying transport,  
1148 the actual order that a Notification Recipient receives separate Event Notifications may differ from the  
1149 order sent by the Printer (e.g., email).

1150 Example: Consider two Per-Printer Subscription Objects: SO1 and SO2. SO1 requests 'job-state-  
1151 changed' events and SO2 requests 'printer-state-changed' events. The number in parens is the time  
1152 stamp. The following Event Notification sequences are the only ones that conform to the ordering  
1153 requirements for the Printer to send the Event Notifications:

1154 (a) SO1: 'job-created' (1000), SO1: 'job-stopped' (1005), SO1: 'job-completed' (1009), SO2: 'printer-  
1155 stopped' (1005)

1156 (b) SO1: 'job-created' (1000), SO1: 'job-stopped' (1005), SO2: 'printer-stopped' (1005), SO1: 'job-  
1157 completed' (1009)

1158 (c) SO1: 'job-created' (1000), SO2: 'printer-stopped' (1005), SO1: 'job-stopped' (1005), SO1: 'job-  
1159 completed' (1009)

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

1162 Examples (b) and (c) are interleaved; examples (a) and (d) are not interleaved and are not appropriate  
1163 for some Delivery Methods.

1164 If two different Events occur simultaneously, or nearly so (e.g., "printer-up-time" has the same value for  
1165 both), the Printer **MUST** create a separate Event Notification for each Event, even if the associated  
1166 Subscription Object is the same for both Events. However, the Printer **MAY** combine these distinct  
1167 Event Notifications into a single Compound Event Notification if the Delivery Method supports  
1168 Compound Event Notifications. For example, suppose that two nearly-simultaneously Events represent  
1169 two successive 'printer-state-changed' Events, one from 'idle' to 'processing' and another from  
1170 'processing' to 'stopped'. These two Events have the same name but are different instances of the  
1171 Event. Then the Printer **MUST** create a separate Event Notification for each Event and **SHOULD**  
1172 accurately report the "printer-state" of the first Event as 'processing' and the second Event as  
1173 'stopped'.

1174 If a Subscription Object contains more than one Subscribed Event, and several Events occur in quick  
1175 succession each matching a different Subscribed Event in the Subscription Object, the Printer **MUST**  
1176 **NOT** generate a single Event Notification from several of these Events, but **MAY** combine distinct  
1177 Event Notifications into a single Compound Event Notification if the Delivery Method supports  
1178 Compound Event Notifications.

1179 After the Printer has created the Event Notification, the Printer delivers it via either a:

1180 Push Delivery Method: The Printer sends the Event Notification shortly after an Event occurs.  
1181 For some Push Delivery Methods, the Notification Recipient **MUST** send a response; for others  
1182 it **MUST NOT** send a response.

1183 Pull Delivery Method: The Printer saves Event Notifications for some Event Life and expects the  
1184 Notification Recipient to request Event Notifications. The Printer returns the Event Notifications  
1185 in a response to such a request.

1186 If an error that meets the following conditions occurs, the Printer **MUST** cancel the Subscription  
1187 Object.

1188 a) the error occurs during the sending of an Event Notification generated from Subscription Object S  
1189 **AND**

1190 b) the error would continue to occur every time the Printer sends an Event Notification generated from  
1191 Subscription Object S in the future.

1192 For example, if the address of the “notify-recipient-uri” of Subscription Object A references a non-  
1193 existent target and the Printer determines this fact, it **MUST** delete Subscription Object A.

1194 The next two sections describe the values that a Printer sends in the content of Machine Consumable  
1195 and Human Consumable Event Notifications, respectively.

1196 The tables in the sub-sections of this section contain the following columns:

1197 a) **Source Value:** the name of the attribute that supplies the value for the Event Notification.  
1198 Asterisks in this field refer to a note below the table.

1199 b) **Sends:** if the Printer supports the value (column 1) on the Source Object (column 3) the Delivery  
1200 Method **MUST** specify:

1201 **MUST:** that the Printer **MUST** send the value.

1202 **SHOULD:** either that the Printer **MUST** send the value or that the value is incompatible with  
1203 the Delivery Method.

1204 **MAY:** that the Printer **MUST**, **SHOULD**, **MAY**, **MUST NOT**, **SHOULD NOT**, or **NEED**  
1205 **NOT** send the value. The Delivery Method specifies the level of conformance for the Printer.

1206 c) **Source Object:** the object from which the source value comes. If the object is “Event  
1207 Notification”, the Printer fabricates the value when it sends the Event Notification. See section 8.

## 1208 9.1 Content of Machine Consumable Event Notifications

1209 This section defines the attributes that a Delivery Method **MUST** mention in a Delivery Method  
1210 Document when specifying the Machine Consumable Event Notification’s contents.

1211 This document does not define the order of attributes in Event Notifications. However, Delivery  
1212 Method Documents **MAY** define the order of some or all of the attributes.

1213 A Delivery Method Document MUST specify additional attributes (if any) that a Printer implementation  
1214 sends in a Machine Consumable Event Notification.

1215 Notification Recipients MUST be able to accept Event Notifications containing attributes they do not  
1216 recognize. What a Notification Recipient does with an unrecognized attribute is implementation-  
1217 dependent. Notification Recipients MAY attempt to display unrecognized attributes anyway or MAY  
1218 ignore them.

1219 The next three sections define the attributes in Event Notification Contents that are:

- 1220 1. for all Events
- 1221 2. for Job Events only
- 1222 3. for Printer Events only

### 1223 9.1.1 Event Notification Content Common to All Events

1224 This section lists the attributes that a Delivery Method Document MUST specify for all Events.

1225 Table 5 lists potential values in each Event Notification.

1226 **Table 5 – Attributes in Event Notification Content**

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

1227 \*A Printer MUST send this value only if and only if it supports the Printer’s “printer-current-time”  
1228 attribute.  
1229

1230 \*\* If the Subscription Object does not contain a “notify-user-data” attribute and the Delivery Method  
 1231 Document REQUIRES the Printer to send the “notify-user-data” source value in the Event Notification,  
 1232 the Printer MUST send an octet-string of length 0.

1233 \*\*\* The last three rows represent additional attributes that a client MAY request via the “notify-  
 1234 attributes” attribute. A Printer MAY support the “notify-attributes” attribute. The Delivery Method  
 1235 MUST say that the Printer MUST, SHOULD, MAY, MUST NOT, SHOULD NOT, or NEED NOT  
 1236 support the “notify-attributes” attribute and specific values of this attribute. The Delivery Method MAY  
 1237 say that support for the “notify-attributes” is conditioned on support of the attribute by the Printer or it  
 1238 MAY say that Printer MUST support the “notify-attributes” attribute if the Printer supports the  
 1239 Delivery Method.

### 1240 9.1.2 Additional Event Notification Content for Job Events

1241 This section lists the additional attributes that a Delivery Method Document MUST specify for Job  
 1242 Events. See Table 6.

1243 **Table 6 – Additional Event Notification Content for Job Events**

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

1244 \* The Printer MUST send the “job-impressions-completed” attribute in an Event Notification only for  
 1245 the combinations of Events and Subscribed Events shown in Table 7.  
 1246

1247 **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’

1248

### 1249 9.1.3 Additional Event Notification Content for Printer Events

1250 This section lists the additional attributes that a Delivery Method Document MUST specify for Printer  
 1251 Events. See Table 8.

1252 **Table 8 – Additional Event Notification Content for Printer Events**

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

1253

1254 **9.2 Content of Human Consumable Event Notification**

1255 This section defines the information that a Delivery Method **MUST** mention in a Delivery Method  
 1256 Document when specifying the Human Consumable Event Notifications contents or the value of the  
 1257 “notify-text” attribute.

1258 Such a Delivery Method **MUST** specify the following information and a Printer **SHOULD** send it:

- 1259 a) the Printer name (see Table 9)
- 1260 b) the time of the Event (see Table 11)
- 1261 c) for Printer Events only:
- 1262 i) the Event (see Table 10) and/or Printer state information (see Table 14)
- 1263 d) for Job Events only:
- 1264 i) the job identity (see Table 12)
- 1265 ii) the Event (see Table 10) and/or Job state information (see Table 13)

1266

1267 The subsections of this section specify the attributes that a Printer **MUST** use to obtain this information.

1268 A Delivery Method Document **MUST** specify additional information (if any) that a Printer  
 1269 implementation sends in a Human Consumable Event Notification or in the “notify-text” attribute.

1270 A client **MUST NOT** request additional attributes via the “notify-attributes” attribute because this  
 1271 attribute works only for Machine Consumable Event Notifications.

1272 Notification Recipients **MUST NOT** expect to be able to parse the Human Consumable Event  
 1273 Notification contents or the value of the “notify-text” attribute.

1274 The next three sections define the attributes in Event Notification Contents that are:

- 1275 a) for all Events
- 1276 b) for Job Events only
- 1277 c) for Printer Events only

1278

1279 **9.2.1 Event Notification Content Common to All Events**

1280 This section lists the source of the information that a Delivery Method **MUST** specify for all Events.



1281 There is a separate table for each piece of information. Each row in the table represents a source value  
 1282 for the information and the values are listed in order of preference, with the first one being the preferred  
 1283 one. An implementation SHOULD use the source value from the earliest row in each table. It MAY use  
 1284 the source value from another row instead, or it MAY combine the source values from several rows. An  
 1285 implementation is free to determine the best way to present this information.

1286 In all tables of this section, all rows contain a “MAY” in order to state that the Delivery Method  
 1287 specifies the conformance.

1288 Table 9 lists the source of the information for the Printer Name. The “printer-name” is more user-  
 1289 friendly unless the Notification Recipient is in a place where the Printer name is not meaningful. For  
 1290 example, an implementation could have the intelligence to send the value of the “printer-name” attribute  
 1291 to a Notification Recipient that can access the Printer via value of the “printer-name” attribute and  
 1292 otherwise send the value of the “notify-printer-uri” attribute.

1293 **Table 9 – Printer Name in Event Notification Content**

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

1294

1295 Table 10 lists the source of the information for the Event name. A Printer MAY combine this  
 1296 information with state information described for Jobs in Table 13 or for Printers in Table 14.

1297 **Table 10 – Event Name in Event Notification Content**

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

1298

1299 Table 11 lists the source of the information for the time that the Event occurred. A Printer can send this  
 1300 value only if it supports the Printer’s “printer-current-time” attribute. If a Printer does not support the  
 1301 “printer-current-time” attribute, it MUST NOT send the “printer-up-time” value instead, since it is not  
 1302 an allowed option for human consumable information.

1303 **Table 11 – Event Time in Event Notification Content**

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

1304

### 1305 9.2.2 Additional Event Notification Content for Job Events

1306 This section lists the source of the additional information that a Delivery Method MUST specify for Job  
1307 Events.

1308 Table 12 lists the source of the information for the job name. The “job-name” is likely more meaningful  
1309 to a user than “job-id”.

1310 **Table 12 – Job Name in Event Notification Content**

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

1311

1312 Table 13 lists the source of the information for the job state. If a Printer supports the “job-state-  
1313 message” and “job-detailed-state-message” attributes, it SHOULD use those attributes for the job state  
1314 information, otherwise, it should fabricate such information from the “job-state” and “job-state-  
1315 reasons”. For some Events, a Printer MAY combine this information with Event information.

1316 **Table 13 – Job State in Event Notification Content**

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

1317

### 1318 9.2.3 Additional Event Notification Content for Printer Events

1319 This section lists the source of the additional information that a Delivery Method MUST specify for  
1320 Printer Events.

1321 Table 14 lists the source of the information for the printer state. If a Printer supports the “printer-state-  
1322 message”, it SHOULD use that attribute for the job state information, otherwise it SHOULD fabricate  
1323 such information from the “printer-state” and “printer-state-reasons”. For some Events, a Printer MAY  
1324 combine this information with Event information.

1325

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

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

1326 **10 Delivery Methods**

1327 A Delivery Method is the mechanism, i.e., protocol, by which the Printer delivers an Event Notification  
 1328 to a Notification Recipient. There are several potential Delivery Methods for Event Notifications,  
 1329 standardized, as well as proprietary. This document does not define any of these delivery mechanisms.  
 1330 Each Delivery Method **MUST** be defined in a Delivery Method Document that is separate from this  
 1331 document. New Delivery Methods will be created as needed using an extension to the registration  
 1332 procedures defined in [RFC2911]. Such documents are registered with IANA (see section 13.7.3).

1333 The following sorts of Delivery Methods are expected:

- 1334 – The Notification Recipient polls for Event Notifications at intervals directed by the Printer
- 1335 – The Printer sends Event Notifications to the Notification Recipient using http as the transport.
- 1336 – The Printer sends an email message.

1337 This section specifies how to define a Delivery Method Document and what to put in such a document.

1338 A Delivery Method Document **MUST** contain an exact copy of the following paragraph, caption and  
 1339 table. In addition, column 2 of the table in the Delivery Method Document **MUST** contain answers to  
 1340 questions in column 1 for the Delivery Method. Also, the Delivery Method document **MUST** contain a  
 1341 reference to this document and call that reference [ipp-ntfy] because the table contains an [ipp-ntfy]  
 1342 reference.

1343 If a Printer supports this Delivery Method, the following are its characteristics.

1344 **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 <b>REQUIRED</b> , <b>RECOMMENDED</b> , or <b>OPTIONAL</b> for an IPP Printer to support?	
3. What transport and delivery protocols does the Printer use	

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

1345

## 1346 **11 Operations for Notification**

1347 This section defines all of the operations for Notification. Section 7.1 assigns the “operation-id” for  
 1348 each operation. The following two sub-sections define Subscription Creation Operations, and other  
 1349 operations.

### 1350 **11.1 Subscription Creation Operations**

1351 This section defines the Subscription Creation Operations. The first section on Create-Job-Subscriptions  
 1352 gives most of the information. The other Subscription Creation Operations refer to the section on  
 1353 Create-Job-Subscriptions, even though the Create-Job-Subscriptions operation is the only OPTIONAL  
 1354 operation in this document (see section 12).

1355 A Printer MUST support Create-Printer-Subscriptions and the Subscription Template Attributes Group  
1356 in Job Creation operations. It MAY support Create-Job-Subscriptions operations.

### 1357 **11.1.1 Create-Job-Subscriptions Operation**

1358 The operation creates one or more Per-Job Subscription Objects. The client supplies one or more  
1359 Subscription Template Attributes Groups each containing one or more of Subscription Template  
1360 Attributes (defined in section 5.3).

1361 Except for errors, the Printer MUST create exactly one Per-Job Subscription Object from each  
1362 Subscription Template Attributes Group in the request, even if the newly created Subscription Object  
1363 would have identical behavior to some existing Subscription Object. The Printer MUST associate each  
1364 newly created Per-Job Subscription Object with the target Job, which is specified by the “notify-job-id”  
1365 operation attribute.

1366 The Printer MUST accept the request in any of the target job’s ‘not-completed’ states, i.e., ‘pending’,  
1367 ‘pending-held’, ‘processing’, or ‘processing-stopped’. The Printer MUST NOT change the job’s “job-  
1368 state” attribute because of this operation. If the target job is in any of the ‘completed’ states, i.e.,  
1369 ‘completed’, ‘canceled’, or ‘aborted, then the Printer MUST reject the request and return the ‘client-  
1370 error-not-possible’ status code; the response MUST NOT contain any Subscription Attribute Groups.

1371 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section  
1372 8.3) performing this operation MUST either be the job owner or have Operator or Administrator access  
1373 rights for this Printer (see [RFC2911] sections 1 and 8.5). Otherwise the Printer MUST reject the  
1374 operation and return: the ‘client-error-forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-  
1375 authorized’ status code as appropriate.

#### 1376 **11.1.1.1 Create-Job-Subscriptions Request**

1377 The following groups of attributes are part of the Create-Job-Subscriptions Request:

1378 Group 1: Operation Attributes

1379 Natural Language and Character Set:

1380 The “attributes-charset” and “attributes-natural-language” attributes as described in  
1381 [RFC2911] section 3.1.4.1.

1382

1383 Target:

1384 The “printer-uri” attribute which defines the target for this operation as described in  
1385 [RFC2911] section 3.1.5.

1386

1387 Requesting User Name:

1388 The “requesting-user-name” attribute SHOULD be supplied by the client as described in  
1389 [RFC2911] section 8.3.

1390

1391 notify-job-id (integer(1:MAX)):  
1392 The client MUST supply this attribute and it MUST specify the Job object to associate the Per-  
1393 Job Subscription with. The value of "notify-job-id" MUST be the value of the "job-id" of the  
1394 associated Job object. If the client does not supply this attribute, the Printer MUST reject this  
1395 request with a 'client-error-bad-request' status code.  
1396

1397 Group 2-N: Subscription Template Attributes

1398 For each occurrence of this group:

1399  
1400 The client MUST supply one or more Subscription Template Attributes in any order. See  
1401 section 5.3 for a description of each such attribute. See section 5.2 for details on processing  
1402 these attributes.

### 1403 11.1.1.2 Create-Job-Subscriptions Response

1404 The Printer MUST return to the client the following sets of attributes as part of a Create-Job-  
1405 Subscriptions response:

1406 Group 1: Operation Attributes

1407 Status Message:

1408 In addition to the REQUIRED status code returned in every response, the response  
1409 OPTIONALLY includes a "status-message" (text(255)) and/or a "detailed-status-message"  
1410 (text(MAX)) operation attribute as described in [RFC2911] sections 13 and 3.1.6.  
1411

1412 In this group, the Printer can return any status codes defined in [RFC2911] and section 16.  
1413 The following is a description of the important status codes:  
1414

1415 **successful-ok:** the Printer created all Subscription Objects requested (see [RFC2911]).

1416 **successful-ok-ignored-subscriptions:** the Printer created some Subscription Objects  
1417 requested but some failed. The Subscription Attributes Groups with a "notify-status-  
1418 code" attribute are the ones that failed (see section 16.1).

1419 **client-error-ignored-all-subscriptions:** the Printer created no Subscription Objects  
1420 requested and all failed. The Subscription Attributes Groups with a "notify-status-  
1421 code" attribute are the ones that failed (see section 16.2).

1422 **client-error-not-possible:** For this operation and other Per-Job Subscription operations,  
1423 this error can occur because the specified Job has already completed (see [RFC2911],  
1424 whether or not the Job is retained in the Job Retention and/or Job History phases (see  
1425 [RFC2911] section 4.3.7.1).  
1426

1427 Natural Language and Character Set:

1428 The "attributes-charset" and "attributes-natural-language" attributes as described in  
1429 [RFC2911] section 3.1.4.2.  
1430

1431 Group 2: Unsupported Attributes

1432 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes. This group does  
1433 not contain any unsupported Subscription Template Attributes; they are returned in the  
1434 Subscription Attributes Group (see below).

1435

1436 Group 3-N: Subscription Attributes

1437 These groups **MUST** be returned unless the Printer is unable to interpret the entire request,  
1438 e.g., the “status-code” parameter returned in Group 1 has the value: ‘client-error-bad-request’.

1439

1440 “notify-status-code” (type2 enum):

1441 Indicates the status of this subscription (see section 17 for the status code definitions).

1442 Section 5.2 defines when this attribute **MUST** be present in this group.

1443

1444 See section 5.2 for details on the contents of each occurrence of this group.

1445

1446 **11.1.2 Create-Printer-Subscriptions operation**

1447 The operation is identical to Create-Job-Subscriptions with exceptions noted in this section.

1448 The operation creates Per-Printer Subscription Objects instead of Per-Job Subscription Objects, and  
1449 associates each newly created Per-Printer Subscription Object with the Printer specified by the  
1450 operation target rather than with a specific Job.

1451 The Printer **MUST** accept the request in any of its states, i.e., ‘idle’, ‘processing’, or ‘stopped’. The  
1452 Printer **MUST NOT** change its “printer-state” attribute because of this operation.

1453 Access Rights: To create Per-Printer Subscription Objects, the authenticated user (see [RFC2911]  
1454 section 8.3) performing this operation **MUST** have Operator or Administrator access rights for this  
1455 Printer (see [RFC2911] sections 1 and 8.5). Otherwise, the Printer **MUST** reject the operation and  
1456 return: the ‘client-error-forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-authorized’  
1457 status code as appropriate.

1458 **11.1.2.1 Create-Printer-Subscriptions Request**

1459 The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.1) except that the  
1460 Operation Attributes group **MUST NOT** contain the “notify-job-id” attribute. If the client does supply  
1461 the “notify-job-id” attribute, then the Printer **MUST** treat it as any other unsupported Operation  
1462 attribute and **MUST** return it in the Unsupported Attributes group.

1463 **11.1.2.2 Create-Printer-Subscriptions Response**

1464 The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.2).

### 1465 11.1.3 Job Creation Operations – Extensions for Notification

1466 This document extends the Job Creation operations (see section 3.2) to create Subscription Objects as a  
1467 part of the operation.

1468 The Job Creation operations are identical to Create-Job-Subscriptions operation with exceptions noted  
1469 in this section.

1470 Unlike the Create-Job-Subscriptions operation, a Job Creation operation associates the newly created  
1471 Subscription Objects with the Job object created by this operation. The operation succeeds if and only if  
1472 the Job creation succeeds. If the Printer does not create some or all of the requested Subscription  
1473 Objects, the Printer MUST return a 'successful-ok-ignored-subscriptions' status-code instead of a  
1474 'successful-ok' status-code, but the Printer MUST NOT reject the operation because of a failure to  
1475 create Subscription Objects.

1476 If the Job Creation operation includes a Job Template group, the client MUST supply it after the  
1477 Operation Attributes group and before the first Subscription Template Attributes Group.

1478 If a Printer does not support this Notification specification, then it MUST treat the Subscription  
1479 Attributes Group like an unknown group and ignore it (see [RFC2911] section 5.2.2). Because the  
1480 Printer ignores the Subscription Attributes Group, it doesn't return them in the response either, thus  
1481 indicating to the client that the Printer doesn't support Notification.

1482 After completion of a successful Job Creation operation, the Printer generates a 'job-created' event (see  
1483 section 5.3.2.1.3).

1484 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [RFC2911] section  
1485 8.3) performing this operation MUST either have permission to create Jobs on the Printer or have  
1486 Operator or Administrator access rights for this Printer (see [RFC2911] sections 1 and 8.5). Otherwise  
1487 the Printer MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-  
1488 authenticated', or 'client-error-not-authorized' status code as appropriate.

#### 1489 11.1.3.1 Job Creation Request

1490 The groups for this operation are sufficiently different from the Create-Job-Subscriptions operation that  
1491 they are all presented here. The following groups of attributes are supplied as part of a Job Creation  
1492 Request:

1493 Group 1: Operation Attributes

1494 Same as defined in [RFC2911] for Print-Job, Print-URI, and Create-Job requests.

1495  
1496 Group 2: Job Template Attributes

1497 The client OPTIONALLY supplies a set of Job Template attributes as defined in [RFC2911]  
1498 section 4.2.



1499  
1500 Group 3 to N: Subscription Template Attributes  
  
1501 The same as Group 2-N in Create-Job-Subscriptions. See section 11.1.1.1.  
1502 Group N+1: Document Content (Print-Job only)  
  
1503 The client MUST supply the document data to be processed.  
1504

### 1505 11.1.3.2 Job Creation Response

1506 The Printer MUST return to the client the following sets of attributes as part of a Print-Job, Print-URI,  
1507 and Create-Job Response:

1508 Group 1: Operation Attributes

1509 Status Message:

1510  
1511 As defined in [RFC2911] for Print-Job, Print-URI, and Create-Job requests.  
1512

1513 In this group, the Printer can return any status codes defined in [RFC2911] and section 16.  
1514 The following is a description of the important status codes:  
1515

1516 **successful-ok:** the Printer created the Job and all Subscription Objects requested (see  
1517 [RFC2911]).

1518 **successful-ok-ignored-subscriptions:** the Printer created the Job and not all of the  
1519 Subscription Objects requested (see section 16.1). This status-code hides ‘successful-  
1520 ok-xxx’ status-codes that could reveal problems in Job creation. The Printer MUST  
1521 NOT return the ‘client-error-ignored-all-subscriptions’ status code for Job Creation  
1522 operations because the Printer returns an error status-code only when it fails to create  
1523 a Job.  
1524

1525 Natural Language and Character Set:

1526 The “attributes-charset” and “attributes-natural-language” attributes as described in  
1527 [RFC2911] section 3.1.4.2.  
1528

1529 Group 2: Unsupported Attributes

1530 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes. This group does  
1531 not contain any unsupported Subscription Template Attributes; they are returned in the  
1532 Subscription Attributes Group (see below).  
1533

1534 Group 3: Job Object Attributes

1535 The “job-id” of the Job Object just created, etc., as defined in [RFC2911] for Print-Job, Print-  
1536 URI, and Create-Job requests.

1537

1538 Group 4 to N: Subscription Attributes

1539 These groups MUST be returned if and only if the client supplied Subscription Template  
1540 Attributes and the operation was accepted.

1541 See section 5.2 for details on the contents of each occurrence of this group.

1542

## 1543 11.2 Other Operations

1544 This section defines other operations on Subscription objects.

### 1545 11.2.1 Restart-Job Operation – Extensions for Notification

1546 The Restart-Job operation [RFC2911] is neither a Job Creation operation nor a Subscription Creation  
1547 operation (see section 3.2). For the Restart-Job operation, the client MUST NOT supply any Job  
1548 Subscription Attributes Groups. The Printer MUST treat any supplied Job Subscription Attributes as  
1549 unsupported attributes.

1550 For this operation, the Printer does not return a job-id or any Subscription Attributes groups because  
1551 the Printer reuses the existing Job object with the same job-id and the existing Per-Job Subscription  
1552 Objects with the same subscription-ids. However, after successful completion of this operation, the  
1553 Printer generates a ‘job-created’ event (see section 5.3.2.1.3).

### 1554 11.2.2 Validate-Job Operation – Extensions for Notification

1555 A client can test whether one or more Subscription Objects could be created using the Validate-Job  
1556 operation. The client supplies one or more Subscription Template Attributes Groups (defined in section  
1557 5.3), just as in a Job Creation request.

1558 A Printer MUST support this extension to this operation.

1559 The Printer MUST accept requests that are identical to the Job Creation request defined in section  
1560 11.1.3.1, except that the request MUST NOT contain document data.

1561 The Printer MUST return the same groups and attributes as the Print-Job operation (section 11.1.3.1)  
1562 with the following exceptions. The Printer MUST NOT return a Job Object Attributes Group because  
1563 no Job is created. The Printer MUST NOT return the “notify-subscription-id” attribute in any  
1564 Subscription Attribute Group because no Subscription Object is created.

1565 If the Printer would succeed in creating a Subscription Object, the corresponding Subscription  
1566 Attributes Group either has no ‘status-code’ attribute or a ‘status-code’ attribute with a value of  
1567 ‘successful-ok-too-many-events’ or ‘successful-ok-ignored-or-substituted-attributes’ (see sections 5.2  
1568 and 17). The status-codes have the same meaning as in Job Creation except the results state what  
1569 “would happen”.

1570 The Printer MUST validate Subscription Template Attributes Groups in the same manner as the Job  
1571 Creation operations.

### 1572 **11.2.3 Get-Printer-Attributes – Extensions for Notification**

1573 This operation is extended so that it returns Printer attributes defined in this document.

1574 A Printer MUST support this extension to this operation.

1575 In addition to the requirements of [RFC2911] section 3.2.5, a Printer MUST support the following  
1576 additional values for the “requested-attributes” Operation attribute in this operation and return such  
1577 attributes in the Printer Object Attributes group of its response.

- 1578 1. **Subscription Template Attributes:** Each supported attribute in column 2 of Table 1.
- 1579 2. **New Printer Description Attributes:** Each supported attribute in section 6.
- 1580 3. **New Group Name:** The ‘subscription-template’ group name, which names all supported  
1581 Subscription Template Attribute in column 2 of Table 1. This group name is also used in the Get-  
1582 Subscription-Attributes and Get-Subscriptions operation with an analogous meaning.
- 1583 4. **Extended Group Name:** The ‘all’ group name, which names all Printer attributes according to  
1584 [RFC2911] section 3.2.5. In this extension ‘all’ names all attributes specified in [RFC2911] plus  
1585 those named in items 1 and 2 of this list.

### 1586 **11.2.4 Get-Subscription-Attributes operation**

1587 This operation allows a client to request the values of the attributes of a Subscription Object.

1588 A Printer MUST support this operation.

1589 This operation is almost identical to the Get-Job-Attributes operation (see [RFC2911] section 3.3.4).  
1590 The only differences are that the operation is directed at a Subscription Object rather than a Job object,  
1591 and the returned attribute group contains Subscription Object attributes rather than Job object  
1592 attributes.

#### 1593 **11.2.4.1 Get-Subscription-Attributes Request**

1594 The following groups of attributes are part of the Get-Subscription-Attributes request:

1595 Group 1: Operation Attributes

1596 Natural Language and Character Set:

1597 The “attributes-charset” and “attributes-natural-language” attributes as described in section  
1598 [RFC2911] 3.1.4.1.

1599

1600

Target:

1601

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

1602

1603

1604

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

1605

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

1606

1607

1608

1609

Requesting User Name:

1610

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

1611

1612

1613

“requested-attributes” (1setOf keyword):

1614

The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute.

1615

This attribute specifies the attributes of the specified Subscription Object that the Printer

1616

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:

1617

1618

1619

1620

- ‘subscription-template’: all attributes that are both defined in section 5.3 and present on

1621

- the specified Subscription Object (column 1 of Table 1).

1622

- ‘subscription-description’: all attributes that are both defined in section 5.4 and present

1623

- on the specified Subscription Object (Table 2).

1624

- ‘all’: all attributes that are present on the specified Subscription Object.

1625

1626

A Printer MUST support all these group names.

1627

If the client omits this attribute, the Printer MUST respond as if this attribute had been

1628

supplied with a value of ‘all’.

1629

1630

#### 11.2.4.2 Get-Subscription-Attributes Response

1631

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

1632

Group 1: Operation Attributes

1633

Status Message:

1634

Same as [RFC2911].

1635

1636

Natural Language and Character Set:

1637

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

1638

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

1639

of the Subscription Object, rather than the one requested.

1640

1641 Group 2: Unsupported Attributes

1642 See [RFC2911] section 3.1.7 and section 3.2.5.2 for details on returning Unsupported  
1643 Attributes.

1644

1645 The response NEED NOT contain the "requested-attributes" operation attribute with any  
1646 supplied keyword values that were requested by the client but are not supported by the IPP  
1647 object. If the Printer object does return unsupported attributes referenced in the "requested-  
1648 attributes" operation attribute, the values of the "requested-attributes" attribute returned  
1649 MUST include only the unsupported keywords that were requested by the client. If the client  
1650 had requested a group name, such as 'all', the resulting unsupported attributes returned MUST  
1651 NOT include attribute keyword names described in the standard but not supported by the  
1652 implementation.

1653

1654 Group 3: Subscription Attributes

1655 This group contains a set of attributes with their current values. Each attribute in this group:

1656

1657 a) MUST be specified by the "requested-attributes" attribute in the request, AND

1658

b) MUST be present on the specified Subscription Object AND

1659

c) MUST NOT be restricted by the security policy in force. For example, a Printer MAY  
1660 prohibit a client who is not the creator of a Subscription Object from seeing some or all of  
1661 its attributes. See [RFC2911] section 8.

1662

The Printer can return the attributes of the Subscription Object in any order. The client MUST  
1663 accept the attributes in any order.

1664

### 1665 11.2.5 Get-Subscriptions operation

1666 This operation allows a client to retrieve the values of attributes of all Subscription Objects belonging to  
1667 a Job or Printer.

1668

A Printer MUST supported this operation.

1669

This operation is similar to the Get-Subscription-Attributes operation, except that this Get-  
1670 Subscriptions operation returns attributes from possibly more than one object.

1671

This operation is similar to the Get-Jobs operation (see [RFC2911] section 3.2.6), except that the  
1672 operation returns Subscription Objects rather than Job objects.

### 1673 11.2.5.1 Get-Subscriptions Request

1674 The following groups of attributes are part of the Get-Subscriptions request:

#### 1675 Group 1: Operation Attributes

1676 Natural Language and Character Set:

1677 The “attributes-charset” and “attributes-natural-language” attributes as described in  
1678 [RFC2911] section 3.1.4.1.

1679

1680 Target:

1681 The “printer-uri” attribute which defines the target for this operation as described in  
1682 [RFC2911] section 3.1.5.

1683

1684 Requesting User Name:

1685 The “requesting-user-name” attribute SHOULD be supplied by the client as described in  
1686 [RFC2911] section 8.3.

1687

1688 “notify-job-id” (integer(1:MAX)):

1689 If the client specifies this attribute, the Printer returns the specified attributes of all Per-Job  
1690 Subscription Objects associated with the Job whose “job-id” attribute value equals the value of  
1691 this attribute. If the client does not specify this attribute, the Printer returns the specified  
1692 attributes of all Per-Printer Subscription Objects. Note: there is no way to get all Per-Job  
1693 Subscriptions known to the Printer in a single operation. A Get-Jobs operation followed by a  
1694 Get-Subscriptions operation for each Job will return all Per-Job Subscriptions.

1695

1696 “limit” (integer(1:MAX)):

1697 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. It  
1698 is an integer value that determines the maximum number of Subscription Objects that a client  
1699 will receive from the Printer even if the “my-subscriptions” attribute constrains which  
1700 Subscription Objects are returned. The limit is a “stateless limit” in that if the value supplied  
1701 by the client is ‘N’, then only the first ‘N’ Subscription Objects are returned in the Get-  
1702 Subscriptions Response. There is no mechanism to allow for the next ‘M’ Subscription  
1703 Objects after the first ‘N’ Subscription Objects. If the client does not supply this attribute, the  
1704 Printer responds with all applicable Subscription Objects.

1705

1706 “requested-attributes” (1setOf type2 keyword):

1707 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute.  
1708 This attribute specifies the attributes of the specified Subscription Objects that the Printer  
1709 MUST return in the response. Each value of this attribute is either an attribute name (defined  
1710 in sections 5.3 and 5.4) or an attribute group name (defined in section 11.2.4.1). If the client  
1711 omits this attribute, the Printer MUST respond as if the client had supplied this attribute with  
1712 the one value: ‘notify-subscription-id’.

1713

1714 “my-subscriptions” (boolean):  
1715 The client **OPTIONALLY** supplies this attribute. The Printer **MUST** support this attribute. If  
1716 the value is ‘false’, the Printer **MUST** consider the Subscription Objects from all users as  
1717 candidates. If the value is ‘true’, the Printer **MUST** return the Subscription Objects created by  
1718 the requesting user of this request. If the client does not supply this attribute, the Printer  
1719 **MUST** respond as if the client had supplied the attribute with a value of ‘false’. The means for  
1720 authenticating the requesting user and matching the Subscription Objects is similar to that for  
1721 Jobs which is described in [RFC2911] section 8.  
1722

### 1723 **11.2.5.2 Get-Subscriptions Response**

1724 The Printer returns the following sets of attributes as part of the Get-Subscriptions Response:

1725 Group 1: Operation Attributes

1726 Status Message:  
1727 Same as [RFC2911].

1728  
1729 Natural Language and Character Set:  
1730 The “attributes-charset” and “attributes-natural-language” attributes as described in  
1731 [RFC2911] section 3.1.4.2.

1732  
1733 Group 2: Unsupported Attributes

1734 Same as for Get-Subscription-Attributes.

1735  
1736 Groups 3 to N: Subscription Attributes

1737 The Printer responds with one Subscription Attributes Group for each requested Subscription  
1738 Object (see the “notify-job-id” attribute in the Operation Attributes Group of this operation).

1739  
1740 The Printer returns Subscription Objects in any order.

1741  
1742 If the “limit” attribute is present in the Operation Attributes group of the request, the number  
1743 of Subscription Attributes Groups in the response **MUST NOT** exceed the value of the “limit”  
1744 attribute.

1745  
1746 If there are no Subscription Objects associated with the specified Job or Printer, the Printer  
1747 **MUST** return zero Subscription Attributes Groups and it **MUST NOT** treat this case as an  
1748 error, i.e., the status-code **MUST** be ‘successful-ok’ unless something else causes the status  
1749 code to have some other value.

1750  
1751 See the Group 3 response (Subscription Attributes Group) of the Get-Subscription-Attributes  
1752 operation (section 11.2.4.2) for the attributes that a Printer returns in this group.  
1753

## 1754 **11.2.6 Renew-Subscription operation**

1755 This operation allows a client to request the Printer to extend the lease on a Per-Printer Subscription  
1756 Object.

1757 The Printer **MUST** support this operation.

1758 The Printer **MUST** accept this request for a Per-Printer Subscription Object in any of the target  
1759 Printer's states, i.e., 'idle', 'processing', or 'stopped', but **MUST NOT** change the Printer's "printer-  
1760 state" attribute.

1761 The Printer **MUST** reject this request for a Per-Job Subscription Object because it has no lease (see  
1762 section 5.4.3). The status code returned **MUST** be 'client-error-not-possible'.

1763 *Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation **MUST**  
1764 either be the owner of the Per-Printer Subscription Object or have Operator or Administrator access  
1765 rights for the Printer (see [RFC2911] sections 1 and 8.5). Otherwise, the Printer **MUST** reject the  
1766 operation and return: the 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-  
1767 authorized' status code as appropriate.

### 1768 **11.2.6.1 Renew-Subscription Request**

1769 The following groups of attributes are part of the Renew-Subscription Request:

1770 Group 1: Operation Attributes

1771 Natural Language and Character Set:

1772 The "attributes-charset" and "attributes-natural-language" attributes as described in  
1773 [RFC2911] section 3.1.4.1.

1774

1775 Target:

1776 The "printer-uri" attribute which defines the target for this operation as described in  
1777 [RFC2911] section 3.1.5.

1778

1779 "notify-subscription-id" (integer (1:MAX)):

1780 The client **MUST** supply this attribute. The Printer **MUST** support this attribute. This attribute  
1781 specifies the Per-Printer Subscription Object whose lease the Printer **MUST** renew. If the client  
1782 omits this attribute, the Printer **MUST** reject this request with the 'client-error-bad-request'  
1783 status code.

1784

1785 Requesting User Name:

1786 The "requesting-user-name" (name(MAX)) attribute **SHOULD** be supplied by the client as  
1787 described in [RFC2911] section 8.3.

1788

1789 Group 2: Subscription Template Attributes



1790  
1791  
1792  
1793  
1794  
1795  
1796

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

The client MAY supply this attribute. It indicates the number of seconds to renew the lease for the specified Subscription Object. A value of 0 requests an infinite lease (which MAY require Operator access rights). If the client omits this attribute, the Printer MUST use the value of the Printer’s “notify-lease-duration-default” attribute. See section 5.3.7 for more details.

## 1797 11.2.6.2 Renew-Subscription Response

1798 The Printer returns the following sets of attributes as part of the Renew-Subscription Response:

1799 Group 1: Operation Attributes

1800 Status Message:

1801 Same as [RFC2911].

1802  
1803  
1804

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

1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814

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

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

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

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

1815 Natural Language and Character Set:

1816 The “attributes-charset” and “attributes-natural-language” attributes as described in  
1817 [RFC2911] section 3.1.4.2. The “attributes-natural-language” MAY be the natural language  
1818 of the Subscription Object, rather than the one requested.

1819  
1820

Group 2: Unsupported Attributes

1821 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes.

1822  
1823

Group 3: Subscription Attributes

1824 The Printer MUST return the following Subscription Attribute:

1825 “notify-lease-duration” (integer(0:MAX)):

1826 The value of this attribute MUST be the number of seconds that the Printer has granted for the  
1827 lease of the Subscription Object (see section 5.3.7 for details, such as the value of this attribute  
1828 when the Printer doesn’t support the requested value).

1829  
1830

### 1831 **11.2.7 Cancel-Subscription operation**

1832 This operation allows a client to delete a Subscription Object and stop the Printer from sending more  
1833 Event Notifications. Once performed, there is no way to reference the Subscription Object.

1834 A Printer **MUST** supported this operation.

1835 The Printer **MUST** accept this request in any of the target Printer's states, i.e., 'idle', 'processing', or  
1836 'stopped', but **MUST NOT** change the Printer's "printer-state" attribute.

1837 If the specified Subscription Object is a Per-Job Subscription Object, the Printer **MUST** accept this  
1838 request in any of the target Job's states, but **MUST NOT** change the Job's "job-state" attribute or affect  
1839 the Job.

1840 *Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation **MUST**  
1841 either be the owner of the Subscription Object or have Operator or Administrator access rights for the  
1842 Printer (see [RFC2911] sections 1 and 8.5). Otherwise, the Printer **MUST** reject the operation and  
1843 return: the 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized'  
1844 status code as appropriate.

1845 Note: There is no way to change any attributes on a Subscription Object, except the "notify-lease-  
1846 duration" attribute (using the Renew-Subscription operation). In order to change other attributes, a  
1847 client performs a Subscription Creation Operation and Cancel-Subscription operation on the old  
1848 Subscription Object. If the client wants to avoid missing Event Notifications, it performs the  
1849 Subscription Creation Operation first. If this order would create too many Subscription Objects on the  
1850 Printer, the client reverses the order.

#### 1851 **11.2.7.1 Cancel-Subscription Request**

1852 The following groups of attributes are part of the Cancel-Subscription Request:

1853 Group 1: Operation Attributes

1854 Natural Language and Character Set:

1855 The "attributes-charset" and "attributes-natural-language" attributes as described in  
1856 [RFC2911] section 3.1.4.1.

1857

1858 Target:

1859 The "printer-uri" attribute which defines the target for this operation as described in  
1860 [RFC2911] section 3.1.5.

1861

1862 “notify-subscription-id” (integer (1:MAX)):  
1863 The client **MUST** supply this attribute. The Printer **MUST** support this attribute. This attribute  
1864 specifies the Subscription Object that the Printer **MUST** cancel. If the client omits this  
1865 attribute, the Printer **MUST** reject this request with the ‘client-error-bad-request’ status code.  
1866

1867 Requesting User Name:

1868 The “requesting-user-name” attribute **SHOULD** be supplied by the client as described in  
1869 [RFC2911] section 8.3.  
1870

## 1871 11.2.7.2 Cancel-Subscription Response

1872 The Printer returns the following sets of attributes as part of the Cancel-Subscription Response:

1873 Group 1: Operation Attributes

1874 Status Message:

1875 Same as [RFC2911].  
1876

1877 The following are some of the status codes returned (see [RFC2911]):  
1878

1879 **successful-ok:** The operation successfully canceled (deleted) the Subscription Object.

1880 **client-error-not-found:** The operation failed because the “notify-subscription-id”  
1881 Operation attribute identified a non-existent Subscription Object.  
1882

1883 Natural Language and Character Set:

1884 The “attributes-charset” and “attributes-natural-language” attributes as described in  
1885 [RFC2911] section 3.1.4.2. The “attributes-natural-language” **MAY** be the natural language  
1886 of the Subscription Object, rather than the one requested.  
1887

1888 Group 2: Unsupported Attributes

1889 See [RFC2911] section 3.1.7 for details on returning Unsupported Attributes.  
1890

## 1891 12 Conformance Requirements

1892 It is **OPTIONAL** for IPP clients and Printers to implement this Event Notification specification.

1893 If this Event Notification specification is implemented, Printers **MUST**:

1894 - meet the Conformance Requirements detailed in section 5 of [RFC2911].

1895 - support the Subscription Template Attributes Group in requests and the Subscription  
1896 Attributes Group in responses.

1897 - support all of the following attributes:

- 1898 a. REQUIRED Subscription Object attributes in section 5.  
 1899 b. REQUIRED Printer Description object attributes in section 6.  
 1900 c. REQUIRED attributes in Event Notification content in section 8.
- 1901 - send Event Notifications that conform to the requirements of section 9 and the requirements of  
 1902 the Delivery Method Document for each supported Delivery Method (the conformance  
 1903 requirements for Delivery Method Documents is specified in section 10).
- 1904 - for all of the Job Creation Operations that the Printer supports, MUST support the  
 1905 REQUIRED extensions for notification defined in section 11.1.3.
- 1906 - meet the conformance requirements for operations as described in Table 16 and meet the  
 1907 requirements for Printers as specified in the indicated sub-sections of section 11:

1908 **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

1909

## 1910 13 IANA Considerations

1911 This section contains the registration information for IANA to add to the various IPP Registries  
 1912 according to the procedures defined in RFC 2911 [RFC2911] section 6 to cover the definitions in this  
 1913 document. In addition, this section defines how Events and Delivery Methods will be registered when  
 1914 they are defined in other documents.

1915 *Note to RFC Editors: Replace RFC NNNN below with the RFC number for this document, so that it*  
 1916 *accurately reflects the content of the information for the IANA Registry.*

### 1917 13.1 Attribute Registrations

1918 The following table lists all the attributes defined in this document. These are to be registered  
 1919 according to the procedures in RFC 2911 [RFC2911] section 6.2.

1920	Subscription Template attributes:	Ref.	Section:
1921	notify-recipient-uri (uri)	RFC NNNN	5.3.1.1
1922	notify-schemes-supported (1setOf uriScheme)	RFC NNNN	5.3.1.1
1923	notify-pull-method (type2 keyword)	RFC NNNN	5.3.1.2

1924	notify-pull-method-supported (1setOf type2 keyword)			
1925		RFC NNNN		5.3.1.2
1926	notify-events (1setOf type2 keyword)	RFC NNNN		5.3.2
1927	notify-events-default (1setOf type2 keyword)	RFC NNNN		5.3.2
1928	notify-events-supported (1setOf type2 keyword)	RFC NNNN		5.3.2
1929	notify-max-events-supported (integer(2:MAX))	RFC NNNN		5.3.2
1930	notify-attributes (1setOf type2 keyword)	RFC NNNN		5.3.3
1931	notify-attributes-supported (1setOf type2 keyword)			
1932		RFC NNNN		5.3.3
1933	notify-user-data (octetString(63))	RFC NNNN		5.3.4
1934	notify-charset (charset)	RFC NNNN		5.3.5
1935	notify-natural-language (naturalLanguage)	RFC NNNN		5.3.6
1936	notify-lease-duration (integer(0:67108863))	RFC NNNN		5.3.7
1937	notify-lease-duration-default (integer(0:67108863))			
1938		RFC NNNN		5.3.7
1939	notify-lease-duration-supported (1setOf (integer(0: 67108863)			
1940	rangeOfInteger(0:67108863)))	RFC NNNN		5.3.7
1941	notify-time-interval (integer(0:MAX))	RFC NNNN		5.3.8
1942				
1943	Subscription Description Attributes:			
1944	notify-subscription-id (integer (1:MAX)))	RFC NNNN		5.4.1
1945	notify-sequence-number (integer (0:MAX)))	RFC NNNN		5.4.2
1946	notify-lease-expiration-time (integer(0:MAX)))	RFC NNNN		5.4.3
1947	notify-printer-up-time (integer(1:MAX)))	RFC NNNN		5.4.4
1948	notify-printer-uri (uri)	RFC NNNN		5.4.5
1949	notify-job-id (integer(1:MAX)))	RFC NNNN		5.4.6
1950	notify-subscriber-user-name (name(MAX)))	RFC NNNN		5.4.7
1951				
1952	Printer Description Attributes:			
1953	printer-state-change-time (integer(1:MAX)))	RFC NNNN		6.1
1954	printer-state-change-date-time (dateTime)	RFC NNNN		6.2
1955				
1956	Attributes Only in Event Notifications			
1957	notify-subscribed-event (type2 keyword)	RFC NNNN		8.1
1958	notify-text (text(MAX))	RFC NNNN		8.2
1959				
1960	The resulting attribute registrations will be published in the			
1961	<a href="ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attributes/">ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attributes/</a>			
1962	area.			
1963				

## 1964 13.2 Additional Enum Attribute Value Registrations for the “operations-supported” Printer 1965 Attribute

1966 The following table lists all the new enum attribute values defined in this document as additional type2  
1967 enum values for use with the “operations-supported” Printer Description attribute. These are to be  
1968 registered according to the procedures in RFC 2911 [RFC2911] section 6.1.

1969	type2 enum Attribute Values:	Value	Ref.	Section:
1970	Create-Printer-Subscriptions	0x0016	RFC NNNN	7.1

1971	Create-Job-Subscriptions	0x0017	RFC NNNN	7.1
1972	Get-Subscription-Attributes	0x0018	RFC NNNN	7.1
1973	Get-Subscriptions	0x0019	RFC NNNN	7.1
1974	Renew-Subscription	0x001A	RFC NNNN	7.1
1975	Cancel-Subscription	0x001B	RFC NNNN	7.1

1976  
 1977 The resulting enum attribute value registrations will be published in the  
 1978 <ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/operations-supported/>  
 1979 area.  
 1980

### 1981 13.3 Operation Registrations

1982 The following table lists all of the operations defined in this document. These are to be registered  
 1983 according to the procedures in RFC 2911 [RFC2911] section 6.4.

1984	Operations:	Ref.	Section:
1985	Create-Job-Subscriptions Operation	RFC NNNN	11.1.1
1986	Create-Printer-Subscriptions Operation	RFC NNNN	11.1.2
1987	Job Creation Operations - Extensions	RFC NNNN	11.1.3
1988	Validate-Job Operation - Extensions	RFC NNNN	0
1989	Get-Printer-Attributes - Extensions	RFC NNNN	11.2.3
1990	Get-Subscription-Attributes Operation	RFC NNNN	11.2.4
1991	Get-Subscriptions Operation	RFC NNNN	11.2.5
1992	Renew-Subscription Operation	RFC NNNN	11.2.6
1993	Cancel-Subscription Operation	RFC NNNN	11.2.7

1994  
 1995 The resulting operation registrations will be published in the  
 1996 <ftp://ftp.iana.org/in-notes/iana/assignments/ipp/operations/>  
 1997 area.  
 1998

### 1999 13.4 Status code Registrations

2000 The following table lists all the status codes defined in this document. These are to be registered  
 2001 according to the procedures in RFC 2911 [RFC2911] section 6.6.

2002	Status codes:	Ref.	Section:
2003	successful-ok-ignored-subscriptions (0x0003)	RFC NNNN	16.1
2004	client-error-ignored-all-subscriptions (0x0414)	RFC NNNN	16.2
2005			
2006	Status Codes in Subscription Attributes Groups:		
2007	client-error-uri-scheme-not-supported (0x040C)	RFC NNNN	17.1
2008	client-error-attributes-or-values-not-supported (0x040B)		
2009		RFC NNNN	17.2
2010	client-error-too-many-subscriptions (0x0415)	RFC NNNN	17.3
2011	successful-ok-too-many-events (0x0005)	RFC NNNN	17.4
2012	successful-ok-ignored-or-substituted-attributes (0x0001)		
2013		RFC NNNN	17.5

2014  
2015  
2016  
2017  
2018

The resulting status code registrations will be published in the  
ftp://ftp.iana.org/in-notes/iana/assignments/ipp/status-codes/  
area.

### 2019 **13.5 Attribute Group tag Registrations**

2020 The following table lists all the attribute group tags defined in this document. These are to be registered  
2021 according to the procedures in RFC 2911 [RFC2911] section 6.5.

2022	Attribute Group Tags:	Tag Value:	Ref.	Section:
2023	subscription-attributes-tag	0x06	RFC NNNN	18
2024	event-notification-attributes-tag	0x07	RFC NNNN	18

2025  
2026  
2027  
2028  
2029

The resulting attribute group tag registrations will be published in the  
ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-group-tags/  
area.

### 2030 **13.6 Registration of Events**

2031 When other document define additional type2 keywords to be used with the “notify-events”  
2032 Subscription Template attribute (see section 5.3.2)), these event keywords will be registered according  
2033 to the procedures of [RFC2911] section 7.1 as additional attribute values for use with the “notify-  
2034 events” Subscription Template attribute, i.e., the "notify-events", "notify-events-default", and "notify-  
2035 events-supported" attributes.

2036 Therefore, the IPP Registry entry for an Event will be of the form:

2037	type2 enum Attribute Values:	Ref.	Section:
2038	<scheme name>	RFC xxxxx	m.n

2039  
2040  
2041  
2042  
2043

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

### 2044 **13.7 Registration of Event Notification Delivery Methods**

2045 This section describes the requirements and procedures for registration and publication of Event  
2046 Notification Delivery Methods and for the submission of such proposals.

#### 2047 **13.7.1 Requirements for Registration of Event Notification Delivery Methods**

2048 Registered IPP Event Notification Delivery Methods are expected to follow a number of requirements  
2049 described below.

### 2050 **13.7.1.1 Required Characteristics**

2051 A Delivery Method Document MUST either (1) contain all of the semantics of the Delivery Method or  
2052 (2) contain the IPP Delivery Method registration requirements and a profile of some other protocol that  
2053 in combination is the Delivery Method (e.g., mailto). In either case, the Delivery Method Document  
2054 (and any documents it requires) MUST define a URL for a Push Delivery Method or a keyword for a  
2055 Pull Delivery method and be a standards track, informational, or experimental RFC that meets the  
2056 requirements of [RFC2717].

2057  
2058 IPP Event Notification Delivery Method Documents MUST meet the requirements of this document  
2059 (see sections 9 and 10).

2060 In addition, a Delivery Method Document MUST contain the following information:

2061  
2062 Type of registration: IPP Event Notification Delivery Method  
2063 Name of this delivery method:  
2064 Proposed URL scheme name of this Push Delivery Method or the keyword name of this Pull  
2065 Delivery Method:  
2066 Name of proposer:  
2067 Address of proposer:  
2068 Email address of proposer:  
2069 Is this delivery method REQUIRED or OPTIONAL for conformance to the IPP Event Notification  
2070 and Subscriptions document:  
2071 Is this delivery method defining Machine Consumable and/or Human Consumable content:  
2072

### 2073 **13.7.1.2 Naming Requirements**

2074 Exactly one (URL scheme or keyword) name MUST be assigned to each Delivery Method.

2075 Each assigned name MUST uniquely identify a single Delivery Method. All Push Delivery Method  
2076 names MUST conform to the rules for URL scheme names, according to [RFC2396] and [RFC2717]  
2077 for schemes in the IETF tree. All Pull Delivery Method names MUST conform to the rules for  
2078 keywords according to [RFC2911].

### 2079 **13.7.1.3 Functionality Requirements**

2080 Delivery Methods MUST function as a protocol that is capable of delivering (push or pull) IPP Event  
2081 Notifications to Notification Recipients.

### 2082 **13.7.1.4 Usage and Implementation Requirements**

2083 Use of a large number of Delivery Methods may hamper interoperability. However, the use of a large  
2084 number of undocumented and/or unlabelled Delivery Methods hampers interoperability even more.



2085 A Delivery Method should therefore be registered ONLY if it adds significant functionality that is  
2086 valuable to a large community, OR if it documents existing practice in a large community. Note that  
2087 Delivery Methods registered for the second reason should be explicitly marked as being of limited or  
2088 specialized use and should only be used with prior bilateral agreement.

### 2089 **13.7.1.5 Publication Requirements**

2090 Delivery Method Documents MUST be published in a standards track, informational, or experimental  
2091 RFCs.

### 2092 **13.7.2 Registration Procedure**

2093 The IPP WG is developing a small number of Delivery Methods which are intended to be published as  
2094 standards track RFCs. However, some parties may wish to register additional Delivery Methods in the  
2095 future. This section describes the procedures for these additional Delivery Methods.

#### 2096 **13.7.2.1 Present the proposal to the Community**

2097 First the Delivery Method Document MUST be an Internet-Draft with a target category of standards  
2098 track, informational, or experimental. The same MUST be true for any documents that it references.

2099 Send the proposed Delivery Method Document proposal to the “ipp@pwg.org” mailing list. This  
2100 mailing list has been established by [RFC2911] for reviewing proposed registrations and discussing  
2101 other IPP matters. Proposed Delivery Method Documents are not formally registered and MUST NOT  
2102 be used until approved.

2103 The intent of the public posting is to solicit comments and feedback on the definition and suitability of  
2104 the Delivery Method and the name chosen for it over a four week period.

#### 2105 **13.7.2.2 Delivery Method Reviewer**

2106 The Delivery Method Reviewer is the same person who has been appointed by the IETF Application  
2107 Area Director(s) as the IPP Designated Expert according to [RFC2911] and [IANA-CON]. When the  
2108 four week period is over and the IPP Designated Expert is convinced that consensus has been achieved,  
2109 the IPP Designated Expert either approves the request for registration or rejects it. Rejection may  
2110 occur because of significant objections raised on the list or objections raised externally.

2111 Decisions made by the Reviewer must be posted to the ipp@pwg.org mailing list within 14 days.  
2112 Decisions made by the Reviewer may be appealed to the IESG.

### 2113 13.7.2.3 IANA Registration

2114 Provided that the Delivery Method registration proposal has either passed review or has been  
 2115 successfully appealed to the IESG, the IANA will register the Delivery Method and make it available to  
 2116 the community.

### 2117 13.7.3 Delivery Method Document Registrations

2118 Each Push Delivery Method Document defines a URI scheme which is registered as an additional value  
 2119 of the “notify-schemes-supported” Printer attribute. These uriScheme values will be registered  
 2120 according to the procedures of [RFC2911] section 7.1 for additional attribute values. Therefore, the  
 2121 IPP Registry entry for a Push Delivery Method will be of the form:

2122	uriScheme Attribute Values:	Ref.	Section:
2123	<scheme name>	RFC xxxx	m.n
2124			

2125 The resulting Delivery Method URI schemes will be published in the  
 2126 ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/notify-schemes-supported/  
 2127 area.

2129 Each Pull Delivery Method Document defines a keyword method which is registered as an additional  
 2130 value of the “notify-pull-method-supported” Printer attribute. These keyword values will be registered  
 2131 according to the procedures of [RFC2911] section 7.1 for additional attribute values. Therefore, the  
 2132 IPP Registry entry for a Pull Delivery Method will be of the form:

2133	keyword Attribute Values:	Ref.	Section:
2134	<method name>	RFC xxxx	m.n
2135			

2136 The resulting Delivery Method URI schemes will be published in the  
 2137 ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/notify-pull-method-supported/  
 2138 area.

### 2140 13.7.4 Registration Template

2141 To: ipp@pwg.org  
 2142 Subject: Registration of a new Delivery Method

2143  
 2144 Delivery Method name:

2145  
 2146 (All Push Delivery Method names must be suitable for use as the value of a URL scheme in the IETF  
 2147 tree and all Pull Delivery Method names must be suitable IPP keywords according to [RFC2911])

2148  
 2149 Published specification(s):

2150

2151 (A specification for the Delivery Method must be openly available that accurately describes what is  
2152 being registered.)

2153  
2154 Person & email address to contact for further information:

## 2155 **14 Internationalization Considerations**

2156 This IPP Notification specification continues support for the internationalization of [RFC2911] of  
2157 attributes containing text strings and names. Allowing a Subscribing Client to specify a different natural  
2158 language and charset for each Subscription Object increases the internationalization support.

2159 The Printer **MUST** be able to localize the content of Human Consumable Event Notifications and to  
2160 localize the value of “notify-text” attribute in Machine Consumable Event Notifications that it sends to  
2161 Notification Recipients. For localization, the Printer **MUST** use the value of the “notify-charset”  
2162 attribute and the “notify-natural-language” attribute in the Subscription Object supplied by the  
2163 Subscribing Client.

## 2164 **15 Security Considerations**

2165 By far the biggest security concern is the abuse of notification: sending unwanted Event Notifications to  
2166 third parties (i.e., spam). The problem is made worse by notification addresses that may be redistributed  
2167 to multiple parties (e.g., mailing lists). There exist scenarios where third party notification is required  
2168 (see Scenario #2 and #3 in [ipp-not-req]). The fully secure solution would require active agreement of  
2169 all recipients before sending out anything. However, requirement #9 in [ipp-req] (“There is no  
2170 requirement for IPP Printer receiving the print request to validate the identity of an Event recipient”)  
2171 argues against this. Certain systems may decide to disallow third party Event Notifications (a traditional  
2172 fax model).

2173 Clients submitting Notification requests to the IPP Printer have the same security issues as submitting an  
2174 IPP/1.1 print job request. The same mechanisms used by IPP/1.1 can therefore be used by the client  
2175 Notification submission. Operations that require authentication can use the HTTP authentication.  
2176 Operations that require privacy can use the HTTP/TLS privacy. As with IPP/1.1 Print Jobs, if there is  
2177 no security on Subscription Objects, sequential assignment of subscription-ids exposes the system to a  
2178 passive traffic monitoring threat.

2179 The Notification access control model should be similar to the IPP access control model for Jobs.  
2180 Creating a Per-Printer Subscription Object is associated with a user. Only the creator or an Operator  
2181 can cancel the Subscription Object. The system may limit the listing of items to only those items owned  
2182 by the user. Some Subscription Objects (e.g., those that have a lifetime longer than a job) can be done  
2183 only by privileged users (users having Operator and/or Administrator access rights), if that is the  
2184 authorization policy.

2185 The standard security concerns (delivery to the right user, privacy of content, tamper proof content)  
2186 apply to the Delivery Method. IPP should use the security mechanism of the Delivery Method used.

2187 Some delivery mechanisms are more secure than others. Therefore, sensitive Event Notifications should  
2188 use the Delivery Method that has the strongest security.

## 2189 **16 Status Codes**

2190 The following status codes are defined as extensions for Notification and are returned as the value of  
2191 the “status-code” parameter in the Operation Attributes Group of a response (see [RFC2911] section  
2192 3.1.6.1). Operations in this document can also return the status codes defined in section 13 of  
2193 [RFC2911]. The ‘successful-ok’ status code is an example of such a status code.

### 2194 **16.1 successful-ok-ignored-subscriptions (0x0003)**

2195 The Subscription Creation Operation was unable to create all requested Subscription Objects.

2196 For a Create-Job-Subscriptions or Create-Printer-Subscriptions operation, this status code means that  
2197 the Printer created one or more Subscription Objects, but not all requested Subscription Objects.

2198 For a Job Creation operation, this status code means that the Printer created the Job along with zero or  
2199 more Subscription Objects. The Printer returns this status code even if other job attributes are  
2200 unsupported or in conflict. That is, if an IPP Printer finds a warning that would allow it to return  
2201 ‘successful-ok-ignored-subscriptions’ and either ‘successful-ok-ignored-or-substituted-attributes’  
2202 and/or ‘successful-ok-conflicting-attributes’, it MUST return ‘successful-ok-ignored-subscriptions’.

### 2203 **16.2 client-error-ignored-all-subscriptions (0x0414)**

2204 This status code is the same as ‘successful-ok-ignored-subscriptions’ except that only the Create-Job-  
2205 Subscriptions and Create-Printer-Subscriptions operation return it. They return this status code only  
2206 when the Printer creates zero Subscription Objects.

## 2207 **17 Status Codes in Subscription Attributes Groups**

2208 This section contains values of the “notify-status-code” (type2 enum) attribute that the Printer returns in  
2209 a Subscription Attributes Group in a response when the corresponding Subscription Object:

- 2210 1. is not created or
- 2211 2. is created and some of the client-supplied attributes are not supported.

2212 The following sections are ordered in decreasing order of importance of the status-codes.

2213 **17.1 client-error-uri-scheme-not-supported (0x040C)**

2214 This status code is defined in [RFC2911]. This document extends its meaning and allows it to be in a  
2215 Subscription Attributes Group of a response.

2216 The scheme of the client-supplied URI in a “notify-recipient-uri” Subscription Template Attribute in a  
2217 Subscription Creation Operation is not supported. See section 5.3.1.1.

2218 **17.2 client-error-attributes-or-values-not-supported (0x040B)**

2219 This status code is defined in [RFC2911]. This document extends its meaning and allows it to be in a  
2220 Subscription Attributes Group of a response.

2221 The method of the client-supplied keyword in a “notify-pull-method” Subscription Template Attribute  
2222 in a Subscription Creation Operation is not supported. See section 5.3.1.2.

2223 **17.3 client-error-too-many-subscriptions (0x0415)**

2224 The number of Subscription Objects supported by the Printer would be exceeded if this Subscription  
2225 Object were created (see section 5.2).

2226 **17.4 successful-ok-too-many-events (0x0005)**

2227 The client supplied more Events in the “notify-events” operation attribute of a Subscription Creation  
2228 Operation than the Printer supports, as indicated in its “notify-max-events-supported” Printer attribute  
2229 (see section 5.3.2).

2230 **17.5 successful-ok-ignored-or-substituted-attributes (0x0001)**

2231 This status code is defined in [RFC2911]. This document extends its meaning to include unsupported  
2232 Subscription Template Attributes and it can appear in a Subscription Attributes Group.

2233 **18 Encodings of Additional Attribute Tags**

2234 This section assigns values to two attributes tags as extensions to the encoding defined in [RFC2910]).

2235 The “subscription-attributes-tag” delimits Subscription Template Attributes Groups in requests and  
2236 Subscription Attributes Groups in responses.

2237 The “event-notification-attributes-tag” delimits Event Notifications in Delivery Methods that use an  
2238 IPP-like encoding.

2239 The following table specifies the values for the delimiter tags:

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

## 2240 19 References

- 2241 [IANA-CON]  
 2242 Narte, T. and Alvestrand, H.T.: Guidelines for Writing an IANA Considerations Section in RFCs,  
 2243 BCP 26, RFC 2434, October 1998.
- 2244 [ipp-not-req]  
 2245 deBry, R., Lewis, H., Hastings, T., “Internet Printing Protocol/1.1: Requirements for IPP  
 2246 Notifications”, <draft-ietf-ipp-not-06.txt>, work in progress, July 17, 2001.
- 2247 [ipp-prog]  
 2248 Hastings, T., Bergman, R., Lewis, H., “IPP: Job Progress Attributes”, <draft-ietf-ipp-job-prog-  
 2249 03.txt> work in progress, July 17, 2001.
- 2250 [ipp-set]  
 2251 Kugler, C., Hastings, T., Herriot, R., Lewis, H., “Internet Printing Protocol (IPP): Job and Printer Set  
 2252 Operations”, <draft-ietf-ipp-job-printer-set-ops-04.txt>, work in progress, July 17, 2001.
- 2253 [RFC2026]  
 2254 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 2255 [RFC2119]  
 2256 S. Bradner, “Key words for use in RFCs to Indicate Requirement Levels”, RFC 2119 , March 1997
- 2257 [RFC2396]  
 2258 Berners-Lee, T., Fielding, R., Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax",  
 2259 RFC 2396, August 1998.
- 2260 [RFC2565]  
 2261 Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.0: Encoding and  
 2262 Transport", RFC 2565, April 1999.
- 2263 [RFC2566]  
 2264 deBry, R., , Hastings, T., Herriot, R., Isaacson, S., Powell, P., “Internet Printing Protocol/1.0:  
 2265 Model and Semantics”, RFC 2566, April 1999.
- 2266 [RFC2567]  
 2267 Wright, D., “Design Goals for an Internet Printing Protocol”, RFC 2567, April 1999.
- 2268 [RFC2568]  
 2269 Zilles, S., “Rationale for the Structure and Model and Protocol for the Internet Printing Protocol”,  
 2270 RFC 2568, April 1999.

- 2271 [RFC2569]  
2272 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC  
2273 2569, April 1999.
- 2274 [RFC2717]  
2275 R. Petke and I. King, "Registration Procedures for URL Scheme Names", RFC 2717, November  
2276 1999.
- 2277 [RFC2910]  
2278 Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.1: Encoding and  
2279 Transport", RFC 2910, September 2000.
- 2280 [RFC2911]  
2281 deBry, R., , Hastings, T., Herriot, R., Isaacson, S., Powell, P., "Internet Printing Protocol/1.1:  
2282 Model and Semantics", RFC 2911, September 2000.

## 2283 **20 Author's Addresses**

- 2284 Robert Herriot  
2285 2066 Byron St.  
2286 Palo Alto, CA 94301  
2287  
2288 Phone: 650-326-8279  
2289 Fax: 650-327-4466  
2290 Email: [bob@herriot.com](mailto:bob@herriot.com)  
2291
- 2292 Tom Hastings  
2293 Xerox Corporation  
2294 737 Hawaii St. ESAE 231  
2295 El Segundo, CA 90245  
2296  
2297 Phone: 310-333-6413  
2298 Fax: 310-333-5514  
2299 e-mail: [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)  
2300
- 2301 Scott A. Isaacson  
2302 Novell, Inc.  
2303 122 E 1700 S  
2304 Provo, UT 84606  
2305  
2306 Phone: 801-861-7366  
2307 Fax: 801-861-2517  
2308 e-mail: [sisaacson@novell.com](mailto:sisaacson@novell.com)  
2309
- 2310 Roger deBry

2311 Utah Valley State College  
2312 Orem, UT 84058

2313  
2314 Phone: (801) 222-8000  
2315 EMail: debryro@uvsc.edu  
2316

2317 Jay Martin  
2318 Underscore Inc.  
2319 9 Jacqueline St.  
2320 Hudson, NH 03051-5308  
2321 603-889-7000  
2322 fax: 775-414-0245  
2323 e-mail: [jkm@underscore.com](mailto:jkm@underscore.com)  
2324

2325 Michael Shepherd  
2326 Xerox Corporation  
2327 800 Phillips Road MS 128-51E  
2328 Webster, NY 14450  
2329  
2330 Phone: 716-422-2338  
2331 Fax: 716-265-8871  
2332 e-mail: [mshepherd@crt.xerox.com](mailto:mshepherd@crt.xerox.com)  
2333

2334 Ron Bergman  
2335 Hitachi Koki Imaging Solutions  
2336 1757 Tapo Canyon Road  
2337 Simi Valley, CA 93063-3394  
2338

2339 Phone: 805-578-4421  
2340 Fax: 805-578-4001  
2341 Email: [rbergma@hitachi-hkis.com](mailto:rbergma@hitachi-hkis.com)  
2342

2343 IPP Web Page: <http://www.pwg.org/ipp/>  
2344 IPP Mailing List: [ipp@pwg.org](mailto:ipp@pwg.org)  
2345

2346 To subscribe to the ipp mailing list, send the following email:

- 2347 1) send it to [majordomo@pwg.org](mailto:majordomo@pwg.org)  
2348 2) leave the subject line blank  
2349 3) put the following two lines in the message body:  
2350 subscribe ipp  
2351 end  
2352

2353 Implementers of this specification document are encouraged to join the IPP Mailing List in order to  
2354 participate in any discussions of clarification issues and review of registration proposals for additional



2355 attributes and values. In order to reduce spam the mailing list rejects mail from non-subscribers, so you  
2356 must subscribe to the mailing list in order to send a question or comment to the mailing list.

## 2357 **A. Appendix - Model for Notification with Cascading Printers**

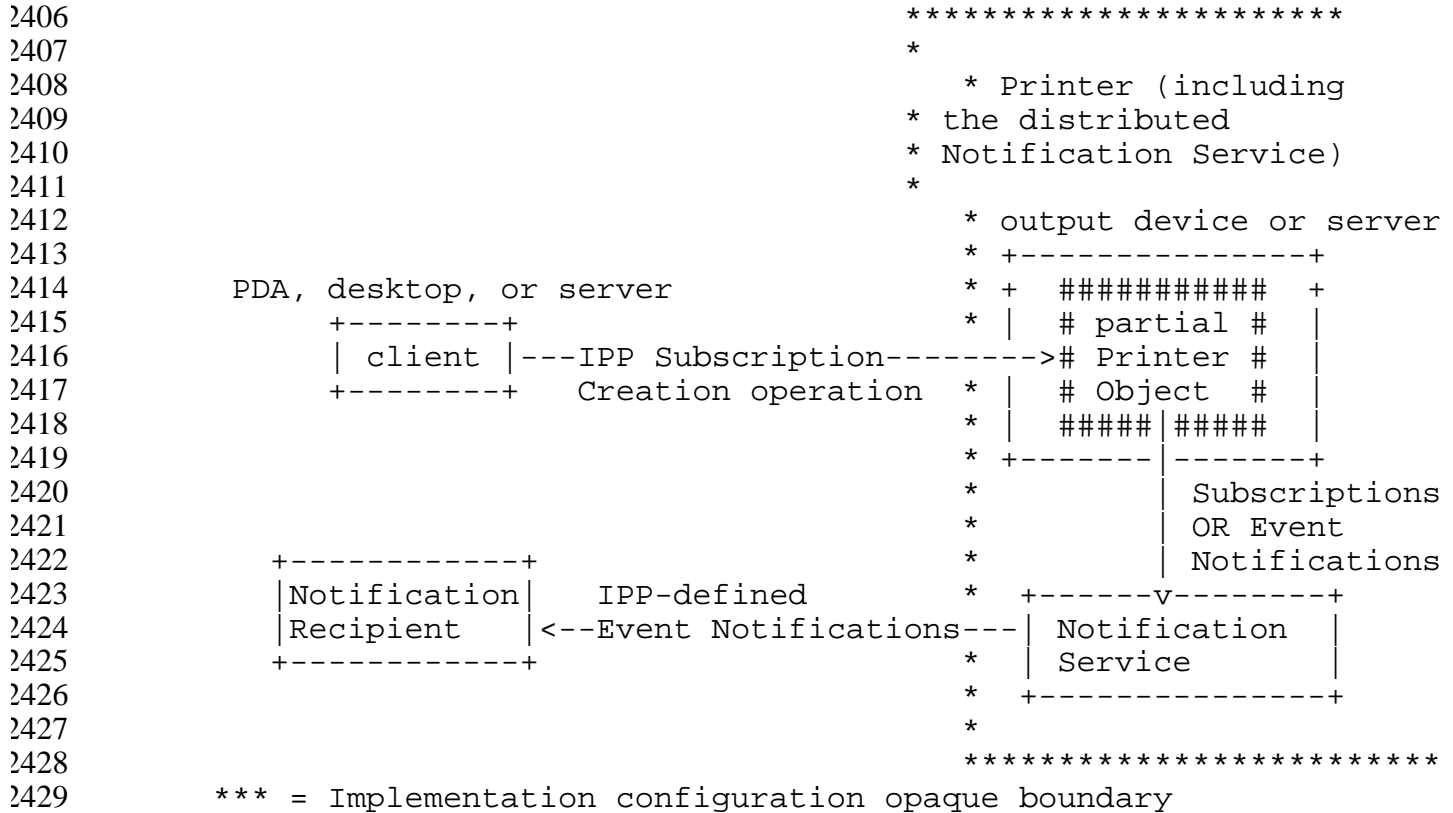
2358 With this model (see Figure 2), there is an intervening Print server between the human user and the  
2359 output-device. So the system effectively has two Printers. There are two cases to consider.

- 2360 1. When the Printer 1 (in the server) generates Events, the system behaves like the client and Printer in  
2361 Figure 1. In this case, Printer 1 sends Event Notifications that are shown as Event Notifications (A)  
2362 of Figure 2,.
- 2363 2. When the Printer 2 (in the output-device) generates Events, there are two possible system  
2364 configurations:
  - 2365 a) Printer 1 forwards the client-supplied Subscription Creation Operations to the downstream  
2366 Printer 2 and lets Printer 2 send the Event Notifications directly to the Notification Recipients  
2367 supplied by the Client (Event Notifications(C) in the diagram).
  - 2368 b) Printer 1 performs the client-supplied Subscription Creation Operations and also forwards the  
2369 Subscription Creation Operations to Printer 2 with the Notification Recipient changed to be the  
2370 Printer 1. When an Event occurs in Printer 2, Printer 2 sends the Event Notification (B) to  
2371 Notification Recipient of Printer 1, which relays the received Event Notification (B) to the client-  
2372 supplied Notification Recipient (as Event Notifications(A) in the diagram). Note, when a client  
2373 performs a Subscription Creation Operation, Printer 1 need not forward the Subscription  
2374 Creation Operation to Printer 2 if it would create a duplicate Subscription Object on Printer 2.

2375 Note: when Printer 1 is forwarding Subscription Creation Operations to Printer 2, it may request Printer  
2376 2 to create additional Subscription Objects (called "piggy-backing"). Piggy-backing is useful when:

- 2377 - Device A is configured to accept (IPP or non-IPP) requests from other servers.
- 2378 - Server S wants to receive Job Events that the client didn't request and Server S wants these  
2379 Events for jobs it submits and not for other jobs.





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

**C. Appendix - Extended Notification Recipient**

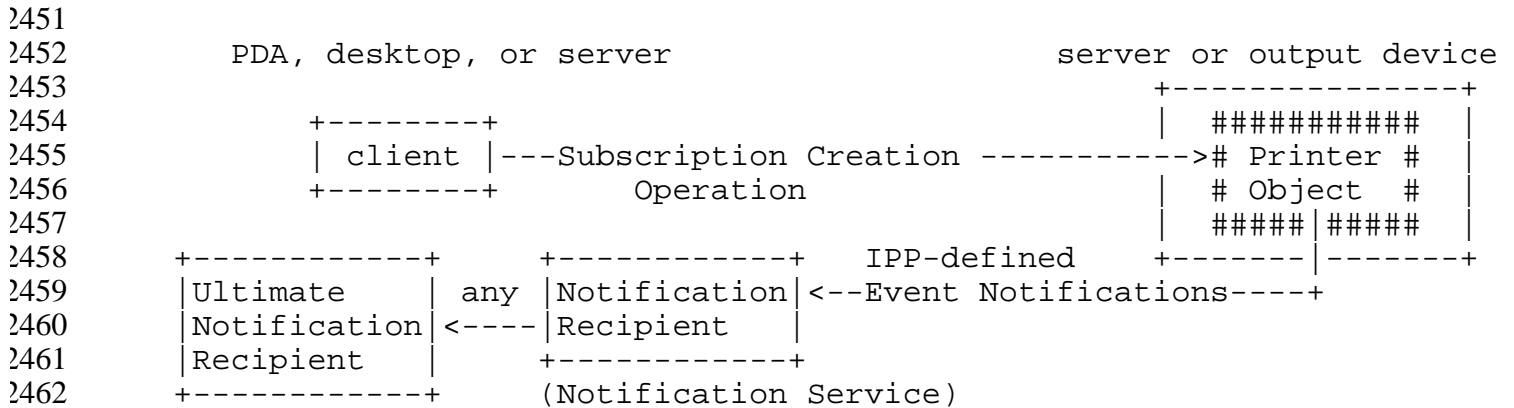
The model allows for an extended Notification Recipient that is itself a notification service that forwards each Event Notification to another recipient (called the Ultimate Notification Recipient in this section). The Delivery Method to the Ultimate Recipient is probably different from the Delivery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

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

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

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

2449 Examples of this extended Notification Recipient are paging, immediate messaging services, general  
 2450 notification services, and NOS vendors' infrastructure. Figure 4 shows this approach.



2463 **Figure 4 – Use of an Extended Notification Recipient transparent to the Printer**

2464 **D. Appendix - Details about Conformance Terminology**

2465 The following paragraphs provide more details about conformance terminology.

2466 **REQUIRED** - an adjective used to indicate that a conforming IPP Printer implementation **MUST**  
 2467 support the indicated operation, object, attribute, attribute value, status code, or out-of-band value  
 2468 in requests and responses. See [RFC2911] “Appendix A - Terminology for a definition of  
 2469 “support”. *Since support of this entire Notification specification is OPTIONAL for*  
 2470 *conformance to IPP/1.0 or IPP/1.1, the use of the term REQUIRED in this document means*  
 2471 *“REQUIRED if this OPTIONAL Notification specification is implemented”.*

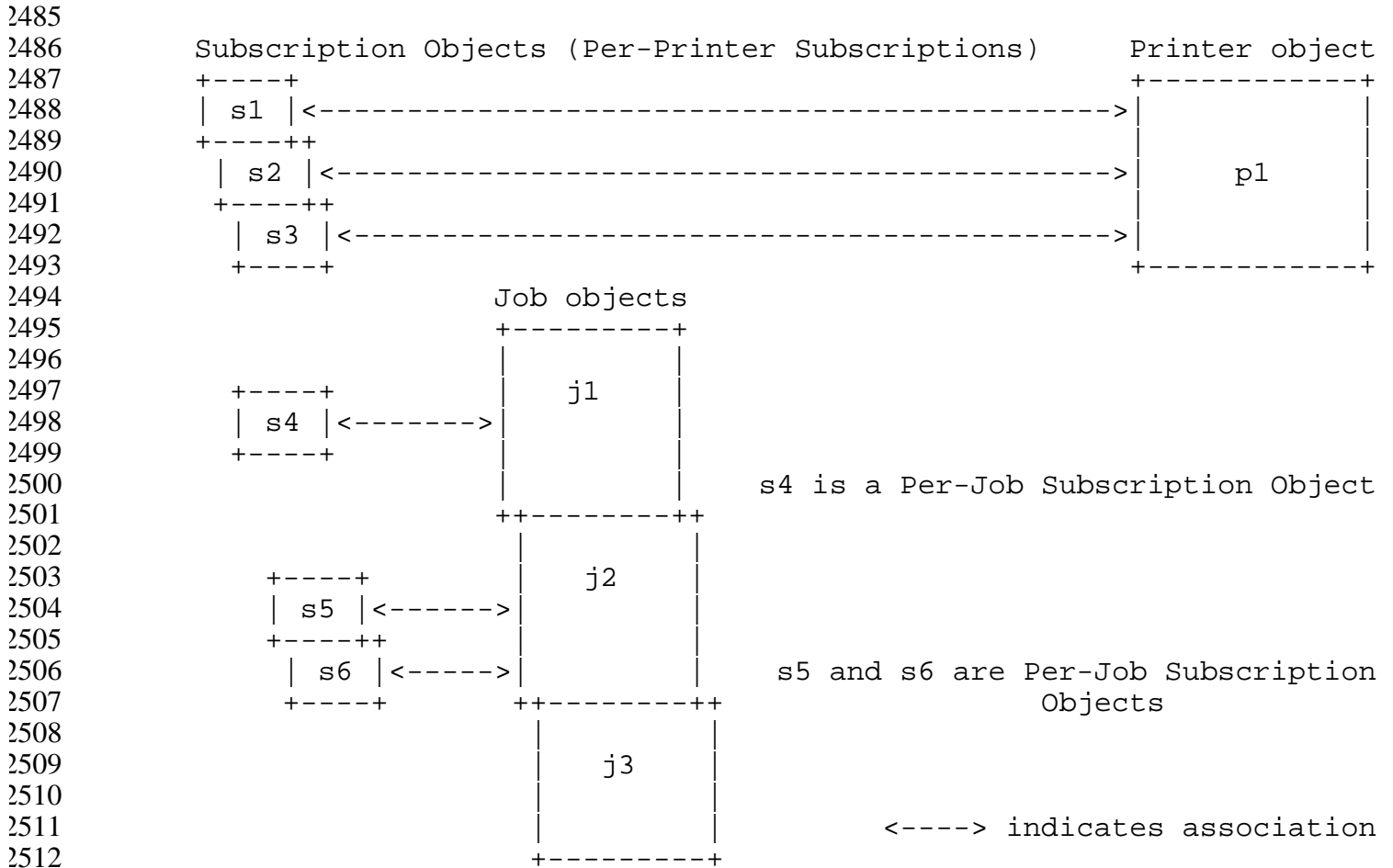
2472 **RECOMMENDED** - an adjective used to indicate that a conforming IPP Printer implementation is  
 2473 recommended to support the indicated operation, object, attribute, attribute value, status code, or  
 2474 out-of-band value in requests and responses. *Since support of this entire Notification*  
 2475 *specification is OPTIONAL for conformance to IPP/1.0 or IPP/1.1, the use of the term*  
 2476 *RECOMMENDED in this document means “RECOMMENDED if this OPTIONAL*  
 2477 *Notification specification is implemented”.*

2478 **OPTIONAL** - an adjective used to indicate that a conforming IPP Printer implementation **MAY**, but is  
 2479 NOT REQUIRED to, support the indicated operation, object, attribute, attribute value, status  
 2480 code, or out-of-band value in requests and responses.

2481 **E. Appendix - Object Model for Notification**

2482 This section describes the Notification object model that adds a Subscription Object which together  
 2483 with the Job and Printer object provide the complete Notification semantics.

2484 The object relationships can be seen pictorially as:



2513 **Figure 5 – Object Model for Notification**

2514 s1, s2, and s3 are Per-Printer Subscription Objects and can identify Printer and/or Job Events.

2515 s4, s5, and s6 are Per-Job Subscription Objects and can identify Printer and/or Job Events.

## 2516 E.1 Appendix - Object relationships

2517 This sub-section defines the object relationships between the Printer, Job, and Subscription Objects by  
 2518 example. Whether Per-Printer Subscription Objects are actually contained in a Printer object or are just  
 2519 bi-directionally associated with them in some way is IMPLEMENTATION DEPENDENT and is  
 2520 transparent to the client. Similarly, whether Per-Job Subscription Objects are actually contained in a  
 2521 Job object or are just bi-directionally associated with them in some way is IMPLEMENTATION  
 2522 DEPENDENT and is transparent to the client. The object relationships are defined as follows:

### 2523 E.2 Printer Object and Per-Printer Subscription Objects

- 2524 1. The Printer object contains (is associated with) zero or more Per-Printer Subscription Objects  
 2525 (p1 contains s1-s3 Per-Printer Subscription Objects).

- 2526 2. Each Per-Printer Subscription Object (s1, s2, and s3) is contained in (or is associated with)  
2527 exactly one Printer object (p1).

### 2528 **E.3 Job Object and Per-Job Subscription Objects**

- 2529 1. A Job object (j1, j2, j3) is associated with zero or more Per-Job Subscription Objects (s4-s6).  
2530 Job j1 is associated with Per-Job Subscription Object s4, Job j2 is associated with Per-Job  
2531 Subscription Objects s5 and s6, and Job j3 is not associated with any Per-Job Subscription  
2532 Object.

- 2533 2. Each Per-Job Subscription Object is associated with exactly one Job object.

## 2534 **F. Appendix - Per-Job versus Per-Printer Subscription Objects**

2535 Per-Job and Per-Printer Subscription Objects are quite similar. Either type of Subscription Object can  
2536 subscribe to Job Events, Printer Events, or both. Both types of Subscription Objects can be queried  
2537 using the Get-Subscriptions and Get-Subscription-Attributes operations and canceled using the Cancel-  
2538 Subscription operation. Both types of Subscription Objects create Subscription Objects which have the  
2539 same Subscription Object attributes defined. However, there are some semantic differences between  
2540 Per-Job Subscription Objects and Per-Printer Subscription Objects. A Per-Job Subscription Object is  
2541 established by the client when submitting a job and after creating the job using the Create-Job-  
2542 Subscriptions operation by specifying the “job-id” of the Job with the “notify-job-id” attribute. A Per-  
2543 Printer Subscription Object is established between a client and a Printer using the Create-Printer-  
2544 Subscriptions operation. Some specific differences are:

- 2545 1. A client usually creates one or more Per-Job Subscription Objects as part of the Job Creation  
2546 operations (Create-Job, Print-Job, and Print-URI), rather than using the OPTIONAL Create-Job-  
2547 Subscriptions operation, especially since Printer implementations NEED NOT support the  
2548 Create-Job-Subscriptions operation, since it is OPTIONAL.
- 2549 2. For Per-Job Subscription Objects, the Subscription Object is only valid while the job is “not-  
2550 complete” (see sections 5.4.3) while for the Per-Printer Subscription Objects, the Subscription  
2551 Object is valid until the time (in seconds) that the Printer returned in the “notify-lease-expiration-  
2552 time” operation attribute.
- 2553 3. Job Events in a Per-Job Subscription Object apply only to “one job” (the Job created by the Job  
2554 Creation operation or references by the Create-Job-Subscriptions operation) while Job Events in  
2555 a Per-Printer Subscription Object apply to ALL jobs contained in the IPP Printer.

## 2556 **G. Appendix - Description of the base IPP documents**

2557 The base set of IPP documents includes:

2558 Design Goals for an Internet Printing Protocol [RFC2567]  
2559 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]  
2560 Internet Printing Protocol/1.1: Model and Semantics [RFC2911]  
2561 Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]  
2562 Internet Printing Protocol/1.1: Implementer's Guide [IPP-IIG]  
2563 Mapping between LPD and IPP Protocols [RFC2569]  
2564

2565 The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed  
2566 printing functionality, and it enumerates real-life scenarios that help to clarify the features that need to  
2567 be included in a printing protocol for the Internet. It identifies requirements for three types of users:  
2568 end users, operators, and administrators. It calls out a subset of end user requirements that are satisfied  
2569 in IPP/1.0 [RFC2566, RFC2565]. A few OPTIONAL operator operations have been added to IPP/1.1  
2570 [RFC2911, RFC2910].

2571 The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document  
2572 describes IPP from a high level view, defines a roadmap for the various documents that form the suite of  
2573 IPP specification documents, and gives background and rationale for the IETF IPP working group's  
2574 major decisions.

2575 The "Internet Printing Protocol/1.1: Model and Semantics" document describes a simplified model with  
2576 abstract objects, their attributes, and their operations. The model introduces a Printer and a Job. The  
2577 Job supports multiple documents per Job. The model document also addresses how security,  
2578 internationalization, and directory issues are addressed.

2579 The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the  
2580 abstract operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It also  
2581 defines the encoding rules for a new Internet MIME media type called "application/ipp". This document  
2582 also defines the rules for transporting over HTTP a message body whose Content-Type is  
2583 "application/ipp". This document defines the 'ipp' scheme for identifying IPP printers and jobs.

2584 The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to  
2585 implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some  
2586 of the considerations that may assist them in the design of their client and/or IPP object  
2587 implementations. For example, a typical order of processing requests is given, including error checking.  
2588 Motivation for some of the specification decisions is also included.

2589 The "Mapping between LPD and IPP Protocols" document gives some advice to implementers of  
2590 gateways between IPP and LPD (Line Printer Daemon) implementations.

## 2591 H. Appendix - Full Copyright Statement

2592 Copyright (C) The Internet Society (1998,1999,2000,2001). All Rights Reserved

2593 This document and translations of it may be copied and furnished to others, and derivative works that  
2594 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published  
2595 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright

2596 notice and this paragraph are included on all such copies and derivative works. However, this  
2597 document itself may not be modified in any way, such as by removing the copyright notice or references  
2598 to the Internet Society or other Internet organizations, except as needed for the purpose of developing  
2599 Internet standards in which case the procedures for copyrights defined in the Internet Standards process  
2600 must be followed, or as required to translate it into languages other than English.

2601 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or  
2602 its successors or assigns.

2603 This document and the information contained herein is provided on an "AS IS" basis and THE  
2604 INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL  
2605 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY  
2606 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY  
2607 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
2608 PARTICULAR PURPOSE.

#### 2609 **Acknowledgement**

2610  
2611 Funding for the RFC Editor function is currently provided by the Internet Society.