

1 INTERNET-DRAFT
2 <draft-ietf-ipp-not-spec-04.txt>

R. Herriot (editor)
Xerox Corporation
T. Hastings
Xerox Corporation
R. deBry
Utah Valley State College
S. Isaacson
Novell, Inc.
J. Martin
Underscore
M. Shepherd
Xerox Corporation
R. Bergman
Hitachi Koki Imaging Solutions
July 13, 2000

16 Internet Printing Protocol (IPP):
17 **IPP Event Notification Specification**

18
19 Copyright (C) The Internet Society (2000). All Rights Reserved.
20

21 Status of this Memo

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

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

29 The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

30 The list of Internet-Draft Shadow Directories can be accessed as <http://www.ietf.org/shadow.html>.

31 **Abstract**

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

41

42 The full set of IPP documents includes:

43 Design Goals for an Internet Printing Protocol [RFC2567]

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

45 Internet Printing Protocol/1.1: Model and Semantics [IPP-MOD]

46 Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]

47 Internet Printing Protocol/1.1: Implementer's Guide [IPP-IIG]

48 Mapping between LPD and IPP Protocols [RFC2569]

49 The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed printing
 50 functionality, and it enumerates real-life scenarios that help to clarify the features that need to be included
 51 in a printing protocol for the Internet. It identifies requirements for three types of users: end users,
 52 Operators, and Administrators. It calls out a subset of end user requirements that are satisfied in IPP/1.0.
 53 Operator and Administrator requirements are out of scope for version 1.0. A few OPTIONAL Operator
 54 operations have been added to IPP/1.1.

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

58 The "Internet Printing Protocol/1.1: Model and Semantics", describes a simplified model with abstract
 59 objects, their attributes, and their operations that are independent of encoding and transport. It introduces a
 60 Printer object and a Job object. The Job object optionally supports multiple documents per Job. It also
 61 addresses security, internationalization, and directory issues.

62 The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the abstract
 63 operations and attributes defined in the model document onto HTTP/1.1. It defines the encoding rules for a
 64 new Internet MIME media type called "application/ipp". This document also defines the rules for
 65 transporting over HTTP a message body whose Content-Type is "application/ipp". This document defines
 66 a new scheme named 'ipp' for identifying IPP printers and jobs. Finally, this document defines
 67 interoperability rules for supporting IPP/1.0 clients.

68 The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to
 69 implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.0 and some of the
 70 considerations that may assist them in the design of their client and/or IPP object implementations. For
 71 example, a typical order of processing requests is given, including error checking. Motivation for some of
 72 the specification decisions is also included.

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

75 **Table of Contents**

76	1	Introduction	7
77		1.1 Notification Overview	7

78	2	Models for Notification	9
79		2.1 Model for Notification (Simple Case).....	9
80		2.2 Model for Notification with Cascading Printers	10
81		2.3 Distributed Model for Notification.....	10
82		2.4 Extended Notification Recipient	10
83	3	Terminology	10
84		3.1 Conformance Terminology.....	11
85		3.2 Other Terminology	11
86	4	Object Relationships.....	13
87		4.1 Printer and Per-Printer Subscription Objects	13
88		4.2 Printer, Job and Per-Job Subscription Objects	13
89	5	Subscription Object	13
90		5.1 Rules for Support of Subscription Template Attributes	14
91		5.2 Rules for Processing Subscription Template Attributes.....	15
92		5.3 Subscription Template Attributes.....	17
93		5.3.1 notify-recipient-uri (uri)	18
94		5.3.2 notify-events (1setOf type2 keyword)	19
95		5.3.3 notify-attributes (1setOf type2 keyword).....	23
96		5.3.4 notify-user-data (octetString(63)).....	24
97		5.3.5 notify-charset (charset)	25
98		5.3.6 notify-natural-language (naturalLanguage)	25
99		5.3.7 notify-lease-duration (integer(0:67108863))	25
100		5.3.8 notify-time-interval (integer(0:MAX)).....	26
101		5.4 Subscription Description Attributes	27
102		5.4.1 notify-subscription-id (integer (1:MAX)).....	27
103		5.4.2 notify-sequence-number (integer (0:MAX))	28
104		5.4.3 notify-lease-expiration-time (integer(0:MAX)).....	28
105		5.4.4 notify-printer-up-time (integer(1:MAX))	29
106		5.4.5 notify-printer-uri (uri).....	29
107		5.4.6 notify-job-id (integer(1:MAX)).....	29
108		5.4.7 notify-subscriber-user-name (name(MAX)).....	30
109	6	Printer Description Attributes Related to Notification.....	30
110		6.1 printer-state-change-time (integer(1:MAX)).....	30
111		6.2 printer-state-change-date-time (dateTime)	31
112	7	New Values for Existing Printer Description Attributes.....	31
113		7.1 operations-supported (1setOf type2 enum)	31
114	8	Attributes Only in Event Notifications.....	31
115		8.1 notify-subscribed-event (type2 keyword).....	31
116		8.2 notify-text (text(MAX)).....	32

117	9	Event Notification Content.....	32
118	9.1	Content of Machine Consumable Event Notifications.....	34
119	9.1.1	Event Notification Content Common to All Events.....	34
120	9.1.2	Additional Event Notification Content for Job Events	35
121	9.1.3	Additional Event Notification Content for Printer Events	36
122	9.2	Content of Human Consumable Event Notification.....	36
123	9.2.1	Event Notification Content Common to All Events.....	37
124	9.2.2	Additional Event Notification Content for Job Events	38
125	9.2.3	Additional Event Notification Content for Printer Events	38
126	10	Delivery Methods	39
127	11	Operations for Notification.....	41
128	11.1	Subscription Creation Operations.....	41
129	11.1.1	Create-Job-Subscriptions Operation.....	41
130	11.1.2	Create-Printer-Subscriptions operation	43
131	11.1.3	Job Creation Operation – Extensions for Notification	44
132	11.2	Other Operations	46
133	11.2.1	Validate-Job Operation - Extensions for Notification.....	46
134	11.2.2	Get-Printer-Attributes - Extensions for Notification.....	46
135	11.2.3	Get-Subscription-Attributes operation	47
136	11.2.4	Get-Subscriptions operation	49
137	11.2.5	Renew-Subscription operation	51
138	11.2.6	Cancel-Subscription operation	53
139	12	Conformance Requirements	54
140	13	IANA Considerations	55
141	13.1	Format and Requirements for IPP Delivery Method Registration Proposals.....	56
142	14	Internationalization Considerations.....	56
143	15	Security Considerations.....	57
144	16	Status Codes	57
145	16.1	successful-ok-ignored-subscriptions (0x0003).....	57
146	16.2	client-error-ignored-all-subscriptions (0x0414)	58
147	17	Status Codes in Subscription Attributes Groups	58
148	17.1	client-error-uri-scheme-not-supported (0x040C)	58
149	17.2	client-error-too-many-subscriptions (0x0415)	58
150	17.3	successful-ok-too-many-events (0x0005).....	58
151	17.4	successful-ok-ignored-or-substituted-attributes (0x0001).....	58
152	18	Encodings of Additional Attribute Tags.....	59
153	19	References	59

154	20	Author's Addresses	60
155	A.	Appendix - Model for Notification with Cascading Printers	61
156	B.	Appendix - Distributed Model for Notification.....	62
157	C.	Appendix - Extended Notification Recipient	63
158	D.	Appendix - Details about Conformance Terminology	64
159	E.	Appendix - Object Model for Notification.....	64
160		E.1 Appendix - Object relationships	65
161		E.2 Printer Object and Per-Printer Subscription Objects.....	65
162		E.3 Job Object and Per-Job Subscription Objects	66
163	F.	Appendix - Per-Job versus Per-Printer Subscription Objects.....	66
164	G.	Appendix: Change History (to be removed for Internet-Draft).....	66
165		G.1 Changes to the June 30, 2000 version to create the July 13, 2000 version	66
166		G.2 Changes to the May 10, 2000 version to create the June 30, 2000 version.....	67
167		G.3 Changes to the March 8, 2000 version to create the May 10, 2000 version.....	68
168		G.4 Changes to the March 6, 2000 version to create the March 8, 2000 version.....	69
169		G.5 Changes to the February 2, 2000 version to create the March 6, 2000 version.....	69
170		G.6 Changes to the October 14, 1999 version to create the February 2, 2000 version	70
171	H.	Appendix: Full Copyright Statement.....	72

172

173

Tables

174	Table 1 – Subscription Template Attributes.....	18
175	Table 2 – Subscription Description Attributes	27
176	Table 3 – Printer Description Attributes Associated with Notification	30
177	Table 4 – Operation-id assignments	31
178	Table 5 – Attributes in Event Notification Content.....	34
179	Table 6 – Additional Event Notification Content for Job Events	35
180	Table 7 – Combinations of Events and Subscribed Events for “job-impressions-completed”	35
181	Table 8 – Additional Event Notification Content for Printer Events	36
182	Table 9 – Printer Name in Event Notification Content	37
183	Table 10 – Event Name in Event Notification Content.....	37
184	Table 11 – Event Time in Event Notification Content.....	38
185	Table 12 – Job Name in Event Notification Content	38
186	Table 13 – Job State in Event Notification Content.....	38
187	Table 14 – Printer State in Event Notification Content.....	39
188	Table 15 – Information about the Delivery Method.....	39
189	Table 16 – Conformance Requirements for Operations.....	55

190

Figures

191 Figure 1 – Model for Notification9
192 Figure 2 – Model for Notification with Cascading Printers62
193 Figure 3 – Opaque Use of a Notification Service Transparent to the Client.....63
194 Figure 4 – Use of an Extended Notification Recipient transparent to the Printer64
195 Figure 5 – Object Model for Notification.....65
196

197 **1 Introduction**

198 This IPP notification specification is an extension to IPP/1.0 [RFC2568, RFC2569] and IPP/1.1 [ipp-mod,
199 ipp-pro]. This document in combination with the following documents is intended to meet the notification
200 requirements described in [ipp-not-req]:

201 Internet Printing Protocol (IPP): “Job Progress Attributes” [ipp-prog]
202 One or more Delivery Method Documents registered with IANA (see section 13).
203

204 Note: this document does not define any Delivery Methods, but it does define the rules for conformance for
205 Delivery Method Documents.

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

207 **1.1 Notification Overview**

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

213 When a client (called a *Subscribing Client*) performs an operation that creates a Subscription Object, the
214 operation contains one or more *Subscription Template Attributes Groups*. Each such group holds
215 information used by the Printer to initialize a newly created Subscription Object. The Printer creates one
216 Subscription Object for each Subscription Template Attributes Group in the operation. This group is like
217 the Job Template Attributes group defined in [ipp-mod]. The following is an example of the information
218 included in a Subscription Template Attributes Group (see section 5 for details on the Subscription Object
219 attributes):

- 220 1. The names of Subscribed Events that are of interest to the Notification Recipient.
- 221 2. The address (URL) of one Notification Recipient.
- 222 3. The Delivery Method (i.e., the protocol) which the Printer uses to send the Event Notification.
- 223 4. Some opaque data that the Printer sends to the Notification Recipient in the Event Notification. The
224 Notification Recipient might use this opaque data as a forwarding address for the Event
225 Notification.
- 226 5. The charset to use in text fields within an Event Notification
- 227 6. The natural language to use in the text fields of the Event Notification
- 228 7. The requested lease time in seconds for the Subscription Object

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

- 231 • **Job Creation operation:** When a client performs such an operation (Print-Job, Print-URI, and
232 Create-Job), a client can include zero or more Subscription Template Attributes Groups in the
233 request. The Printer creates one Subscription Object for each Subscription Template Attributes
234 Group in the request, and the Printer associates each such Subscription Object with the newly
235 created Job. This document extends these operations' definitions in [ipp-mod] by adding
236 Subscription Template Attributes Groups in the request and Subscription Attributes Groups in the
237 response.
- 238 • **Create-Job-Subscriptions operation:** A client can include one or more Subscription Template
239 Attributes Groups in the request. The Printer creates one Subscription Object for each Subscription
240 Template Attributes Group and associates each with the job that is the target of this operation.
- 241 • **Create-Printer-Subscriptions operation:** A client can include one or more Subscription Template
242 Attributes Groups in the request. The Printer creates one Subscription Object for each Subscription
243 Template Attributes Group and associates each with the Printer that is the target of this operation.
- 244 For each of the above operations:
- 245 • the Printer associates a Subscription Object with the Printer or a specific Job. When a Subscription
246 Object is associated with a Job Object, it is called a *Per-Job Subscription Object*. When a
247 Subscription Object is associated with a Printer Object, it is called a *Per-Printer Subscription*
248 *Object*.
- 249 • the response contains one Subscription Attributes Group for each Subscription Template Attributes
250 Group in the request and in the same order. When the Printer successfully creates a Subscription
251 Object, its corresponding Subscription Attributes Group contains the "notify-subscription-id"
252 attribute. This attribute uniquely identifies the Subscription Object and is analogous to a "job-id" for
253 a Job object. Some operations described below use the "notify-subscription-id" to identify the target
254 Subscription Object.
- 255 This document adds the following additional operations (see section 11.2 for further details)::
- 256 • **Validate-Job operation:** When a client performs this operation, a client can include zero or more
257 Subscription Template Attributes Groups in the request. The Printer determines if it could create
258 one Subscription Object for each Subscription Template Attributes Group in the request. This
259 document extends this operation's definition in [ipp-mod] by adding Subscription Template
260 Attributes Groups in the request and Subscription Attributes Groups in the response.
- 261 • **Get-Subscription-Attributes operation:** This operation allows a client to obtain the specified
262 attributes of a target Subscription Object.
- 263 • **Get-Subscriptions operation:** This operation allows a client to obtain the specified attributes of all
264 Subscription Objects associated with the Printer or a specified Job.
- 265 • **Renew-Subscription operation:** This operation renews the lease on the target Per-Printer
266 Subscription Object before it expires. A newly created Per-Printer Subscription Object receives an

267 initial lease. It is the duty of the client to use this operation frequently enough to preserve a Per-
268 Printer Subscription Object. The Printer deletes a Per-Printer Subscription Object when its lease
269 expires. A Per-Job Subscription Object last exactly as long as its associated Job Object and thus
270 doesn't have a lease.

- 271 • **Cancel-Subscription operation:** This operation cancels the lease on the specified Per-Printer
272 Subscription Object and thereby deletes the Subscription Object.

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

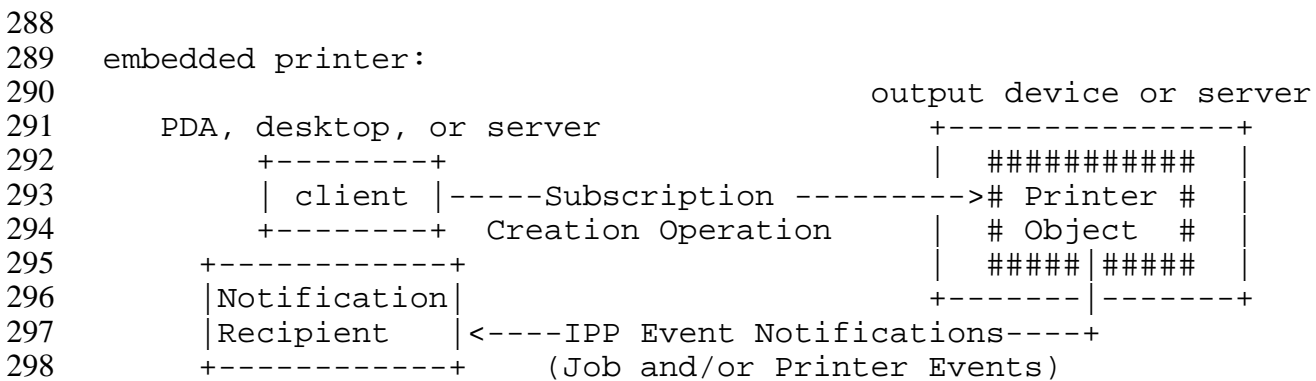
- 275 a) generates an Event Notification with information specified in section 9, AND
- 276 b) either:
 - 277 i) delivers the Event Notification using the Delivery Method and target address identified in the
278 Subscription Object's "notify-recipient-uri" attribute if the Delivery Method is a "push", OR
 - 279 ii) saves Event Notification for a time period defined by the Delivery Method if the Delivery
280 Method is a "pull", i.e., the Notification Recipient is expected to fetch the Event Notifications.

281 2 Models for Notification

282 2.1 Model for Notification (Simple Case)

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

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



299 **Figure 1 – Model for Notification**

300 **2.2 Model for Notification with Cascading Printers**

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

- 304 • directly to the Notification Recipient specified by the Subscribing Client or
- 305 • via the Print Server to the Notification Recipient specified by the Subscribing Client.

306 See Appendix A for more details.

307 **2.3 Distributed Model for Notification**

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

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

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

317 Appendix B describes this example in detail.

318 **2.4 Extended Notification Recipient**

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

323 Appendix C describes this example in detail.

324 **3 Terminology**

325 This section defines terminology used throughout this document.

326 3.1 Conformance Terminology

327 Capitalized terms, such as **MUST**, **MUST NOT**, **REQUIRED**, **SHOULD**, **SHOULD NOT**, **MAY**,
328 **NEED NOT**, and **OPTIONAL**, have special meaning relating to conformance to this specification.
329 These terms are defined in [ipp-mod section 13.1 on conformance terminology, most of which is
330 taken from RFC 2119 [RFC2119]. See Appendix D for complete details.

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

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

337 3.2 Other Terminology

338 **Administrator** - A human user who establishes policy for and configures the print system.

339 **Operator** - A human user who carries out the policy established by the Administrator and controls the
340 day to day running of the print system.

341 **IPP Client (or client)** - The software component (PDA, desktop, or server) that performs an IPP
342 operation directed at an IPP Printer (server or output device).

343 **Job Creation operation** - One of the operations that creates a Job object: Print-Job, Print-URI and
344 Create-Job. The Validate-Job operation is not a Job Creation operation because no Job object is
345 created. Therefore, when a statement also applies to the Validate-Job operation, it is mentioned
346 explicitly.

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

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

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

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

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

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

- 358 **Event Notification** - the information about an Event that the Printer sends when an Event occurs.
- 359 **Notification Recipient** - the entity to which the Printer sends an Event Notification.
- 360 **Delivery Method** - the mechanism by which the Printer delivers the Event Notification, e.g., via email
361 or via SNMP.
- 362 **Delivery Method Document** - a document, separate from this document, that defines a Delivery
363 Method.
- 364 **Compound Event Notification** - two or more Event Notifications that a Printer sends together as a
365 single entity. The Delivery Method Document specifies whether the Delivery Method supports
366 Compound Event Notifications.
- 367 **Subscription Object** - An object containing a set of attributes that indicate: the Notification Recipient,
368 the Delivery Method, the Subscribed Events that cause the Printer to send an Event Notification,
369 and the information to send in an Event Notification.
- 370 **Per-Job Subscription Object** - A Subscription Object that is associated with a single Job. The Create-
371 Job-Subscriptions operation and Job Creation operations create such an object.
- 372 **Per-Printer Subscription Object** - A Subscription Object that is associated with the Printer as a
373 whole. The Create-Printer-Subscriptions operation creates such an object.
- 374 **Subscribing Client** - The client that creates the Subscription Object.
- 375 **Subscription Creation Operation** - An operation that creates a Subscription Object: Job Creation
376 operations, Create-Job-Subscriptions operation, and Create-Printer-Subscriptions operation. In the
377 context of a Job Creation operation, a Subscription Creation Operation is the part of the Job
378 Creation operation that creates a Subscription object.
- 379 **Subscription Creation Request** – The request portion of a Subscription Creation Operation.
- 380 **Subscription Template Attributes** – Subscription Object attributes that a client can supply in a
381 Subscription Creation Operation and associated Printer Object attributes that specify supported and
382 default values for the Subscription Object attributes.
- 383 **Subscription Description Attributes** – Subscription Object attributes that a Printer supplies during a
384 Subscription Creation Operation.
- 385 **Subscription Template Attributes Group** – The attributes group in a request that contains
386 Subscription Object attributes that are Subscription Template Attributes.
- 387 **Subscription Attributes Group** – The attributes group in a response that contains Subscription Object
388 attributes.

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

391 **Machine Consumable Event Notification** - bytes for program consumption. The bytes are formatted
392 according to the Delivery Method document.

393 **Printer** – the software that supports an output device or print server (see IPP/1.1 [ipp-mod] which uses
394 the terms Printer and Printer object interchangeably). This document extends the IPP/1.1 Printer
395 definition to include the software that implements Subscription Creation Operations and the sending
396 of Event Notifications, even if the software for such a Printer would be distributed across a network
397 (see section 2.3).

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

400 **4 Object Relationships**

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

403 **4.1 Printer and Per-Printer Subscription Objects**

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

406 **4.2 Printer, Job and Per-Job Subscription Objects**

- 407 1. A Printer object is associated with zero or more Job objects.
- 408 2. Each Job object is associated with exactly one Printer object.
- 409 3. A Job object is associated with zero or more Per-Job Subscription Objects.
- 410 4. Each Per-Job Subscription Object is associated with exactly one Job object.

411 **5 Subscription Object**

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

416 Using the IPP Job Template attributes as a model (see [ipp-mod] section 4.2), the attributes of a
417 Subscription Object are divided into two categories: Subscription Template Attributes and Subscription
418 Description Attributes.

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

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

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

425 **5.1 Rules for Support of Subscription Template Attributes**

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

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

- 430 1. Each attribute's name starts with the prefix string "notify-" and this document calls such attributes
431 "notify-xxx".
- 432 2. For each "notify-xxx" Subscription Object attribute defined in column 1 of Table 1, Table 1
433 specifies corresponding Printer attributes: "notify-xxx-default", "notify-xxx-supported", "yyy-
434 supported" and "notify-max-xxx-supported" defined in column 2 of Table 1.
- 435 3. If a Printer supports "notify-xxx" in column 1 of Table 1, then the Printer MUST support all
436 associated attributes specified in column 2 of Table 1. For example, Table 1 shows that if the Printer
437 supports "notify-events", it MUST support "notify-events-default", "notify-events-supported" and
438 "notify-max-events-supported".
- 439 4. If a Printer does not support "notify-xxx" in column 1 of Table 1, then the Printer MUST NOT
440 support any associated "notify-yyy" attributes specified in column 2 of Table 1. For example, Table
441 1 shows that if the Printer doesn't support "notify-events", it MUST NOT support "notify-events-
442 default", "notify-events-supported" and "notify-max-events-supported". Note this rule does not
443 apply to attributes whose names do not start with the string "notify-" and are thus defined in another
444 object and used by other attributes.
- 445 5. Most "notify-xxx" attributes have a corresponding "yyy-supported" attribute that specifies the
446 supported values for "notify-xxx". Column 2 of Table 1 specifies the name of each "yyy-supported"
447 attribute. The naming rules of IPP/1.1 (see [ipp-mod]) are used when "yyy-supported" is "notify-
448 xxx-supported".

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

452 If a client wishes to present an end user with a list of supported values from which to choose, the client
453 SHOULD query the Printer for its supported value attributes. The client SHOULD also query the default
454 value attributes. If the client then limits selectable values to only those values that are supported, the client
455 can guarantee that the values supplied by the client in the create request all fall within the set of supported
456 values at the Printer. When querying the Printer, the client MAY enumerate each attribute by name in the
457 Get-Printer-Attributes Request, or the client MAY just supply the ‘subscription-template’ group name in
458 order to get the complete set of supported attributes (both supported and default attributes).

459 5.2 Rules for Processing Subscription Template Attributes

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

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

- 467 1. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer supports it and
468 its value, the Printer MUST populate the attribute on the created Subscription Object.
- 469 2. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer doesn’t support
470 it or its value, the Printer MUST NOT populate the attribute on the created Subscription Object with
471 it. The Printer MUST do one of the following:
 - 472 a) If the value of the “notify-xxx” attribute is unsupported, the Printer MUST return the attribute
473 with its value in the Subscription Attributes Group of the response.
 - 474 b) If “notify-xxx” is an unsupported attribute, the Printer MUST return the attribute in the
475 Subscription Attributes Group of the response with the ‘unsupported’ out-of-band value.

476 Note: The rules of this step are the same as for Unsupported Attributes [ipp-mod] section 3.1.7.
477 except that the unsupported attributes are returned in the Subscription Attributes Group rather than
478 the Unsupported Attributes Group because Subscription Creation Operations can create more than
479 one Subscription Object).

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

- 483 4. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 and the attribute is
484 REQUIRED for the client to supply, the Printer MUST reject the Subscription Creation Operation
485 (including Job Creation operations) without creating a Subscription Object, and MUST return in the
486 response:
- 487 c) the status code ‘client-error-bad-request’ AND
 - 488 d) no Subscription Attribute Groups.
- 489 5. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 that is OPTIONAL for
490 the client to supply, and column 2 of Table 1 either:
- 491 a) specifies a “notify-xxx-default” attribute, the Printer MUST behave as if the client had supplied
492 the “notify-xxx-default” attribute (see step #1) and populate the Subscription object with the
493 value of the “notify-xxx-default” attribute as part of the Subscription Creation operation (unlike
494 Job Template attributes where the Printer does not populate the Job object with defaults - see
495 [ipp-mod]) OR
 - 496 b) does not specify a “notify-xxx-default” attribute, the Printer MUST populate the “notify-xxx”
497 attribute on the Subscription Object according to the definition of the “notify-xxx” attribute in a
498 section 5.3. For some attributes, the “notify-xxx” is populated with the value of some other
499 attribute, and for others, the “notify-xxx” is NOT populated on the Subscription object at all.
- 500 6. A Printer MUST create a Subscription Object for each Subscription Template Attributes group in a
501 request unless the Printer:
- 502 a) encounters some attributes in a Subscription Template Attributes Group that require the Printer
503 not to create the Subscription Object OR
 - 504 b) would create a Per-Job Subscription Object when it doesn’t have space for another Per-Job
505 Subscription Object OR
 - 506 c) would create a Per-Printer Subscription Object when it doesn’t have space for another Per-
507 Printer Subscription Object.
- 508 7. A response MUST contain one Subscription Attributes Group for each Subscription Template
509 Attributes Group in the request (and in the same order) whether the Printer creates a Subscription
510 Object from the Subscription Template Attributes Group or not. However, the attributes in each
511 Subscription Attributes Group can be in any order.
- 512 8. The Printer MUST populate each Subscription Attributes Group of the response such that each
513 contains:
- 514 a) the “notify-subscription-id” attribute (see section 5.4.1), if and only if the Printer creates a
515 Subscription Object.

- 516 b) the “notify-lease-duration” attribute (see section 5.3.7), if and only if the Printer creates a Per-
517 Printer Subscription Object. The value of this attribute is the value of the Subscription Object’s
518 “notify-lease-duration” attribute. This value MAY be different from the client-supplied value
519 (see section 5.3.7). If a client supplies this attribute in the creation of a Per-Job Subscription
520 Object, it MUST appear in this group with the out-of-band value ‘unsupported’ to indicate that
521 the Printer doesn’t support it in this context.
- 522 c) all of the unsupported Subscription Template Attributes from step #2.
- 523 d) the “notify-status-code” attribute if the Printer does not create the Subscription Object or if there
524 are unsupported attributes from step #2. The possible values of the “notify-status-code” attribute
525 are shown below (see section 17 for more details). The Printer returns the first value in the list
526 below that describes the status.
- 527 ‘client-error-uri-scheme-not-supported’: the Subscription Object was not created because
528 the scheme of the “notify-recipient-uri” attribute is not supported. See section 17.1 for
529 more details about this status code. See step #3 in this section for the case that causes
530 this error, and the resulting step #6a) that causes the Printer not to create the Subscription
531 Object.
- 532 ‘client-error-too-many-subscriptions’: the Subscription Object was not created because the
533 Printer has no space for additional Subscription Objects. The client SHOULD try again
534 later. See section 17.2 for more details about this status code. See steps #6b) and #6c) in
535 this section for the cases that causes this error.
- 536 ‘successful-ok-too-many-events’: the Subscription Object was created without the “notify-
537 events” values included in this Subscription Attributes Group because the “notify-
538 events” attribute contains too many values. See section 17.3 for more details about this
539 status code. See step #2 in this section and section 5.3.2 for the cases that cause this
540 status code.
- 541 ‘successful-ok-ignored-or-substituted-attributes’ : the Subscription Object was created but
542 some supplied Subscription Template Attributes are unsupported. These unsupported
543 attributes are also in the Subscription Attributes Group. See section 17.4 for more details
544 about this status code. See step #2 in this section for the cases that cause this status code.
- 545 9. The Printer MUST validate all Subscription Template Attributes and MUST return all unsupported
546 attributes and values in the corresponding Subscription Attributes Group of the response (see step
547 #2) unless it determines that it could not create additional Subscription Objects because of condition
548 #6b) or condition #6c). Then, the Printer NEED NOT validate these additional Subscription
549 Template Attributes and the client MUST NOT expect to find unsupported attributes from step #2
550 in such additional Subscription Attribute Groups.

551 **5.3 Subscription Template Attributes**

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

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

- 554 • **Attribute in Subscription Object:** the name and attribute syntax of each Subscription Object
 555 Attribute that is a Subscription Template Attribute
- 556 • **Default and Supported Printer Attributes:** the default attribute and supported Printer attributes
 557 that are associated with the attribute in column 1.

558 A Printer MUST support all attributes in Table 1 below except for “notify-attributes” (and “notify-
 559 attributes-supported”). A client MUST supply “notify-recipient-uri” and MAY omit any of the rest of the
 560 attributes in column 1 of Table 1 in a Subscription Creation Request.

561 **Table 1 – Subscription Template Attributes**

Attribute in Subscription Object	Default and Supported Printer Attributes
notify-recipient-uri (uri)	notify-schemes-supported (1setOf uriScheme)
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-languages (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))	

562 **5.3.1 notify-recipient-uri (uri)**

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

566 A Printer MUST support this attribute.

567 A client MUST supply this attribute in Subscription Creation Operation. Thus there is no need for a default
 568 attribute.

569 The “notify-schemes-supported (1setOf uriScheme)” attribute MUST specify the schemes supported for
 570 this attribute.

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

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

575 **5.3.2 notify-events (1setOf type2 keyword)**

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

579 A Printer MUST support this attribute.

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

583 Each value of this attribute on a Subscription Object MUST be one of the values of the “notify-events-
584 supported (1setOf type2 keyword)” attribute.

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

590 **5.3.2.1 Standard Values for Subscribed Events**

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

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

596 A Printer MUST support the Events indicated as “REQUIRED” and MAY support the Events indicated as
597 “OPTIONAL”.

598 **5.3.2.1.1 No Events**

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

600 **‘none’**: REQUIRED - no Event Notifications for any Events. As the sole value of “notify-events-
601 supported”, this value means that the Printer does not support the sending of Event Notifications. As
602 the sole value of “notify-events-default”, this value means that a client MUST specify the “notify-
603 events” attribute in order for a Subscription Creation Operation to succeed. If the Printer receives

604 this value as the sole value of a Subscription Creation Operation, it does not create a Subscription
605 Object. If a Printer receives this value with other values of a Subscription Creation Operation, the
606 Printer MUST treat this value as an unsupported value.

607 5.3.2.1.2 Subscribed Printer Events

608 The standard keyword values for Subscribed Printer Events are:

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

612
613 This Subscribed Event value has the following sub-values: ‘printer-restarted’ and ‘printer-
614 shutdown’. A client can listen for any of these sub-values if it doesn’t want to listen to all printer-
615 state changes:

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

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

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

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

629
630 This Event value has the following sub-values: ‘printer-media-changed’ and ‘printer-finishings-
631 changed’. A client can listen for any of these sub-values if it doesn’t want to listen to all printer-
632 configuration changes:

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

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

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

646 5.3.2.1.3 Subscribed Job Events

647 The standard keyword values for Subscribed Job Events are:

648 **‘job-state-changed’**: REQUIRED - the job has changed from any state to any other state. Specifically,
649 the Printer sends this Event whenever the value of the “job-state” attribute or “job-state-reasons”
650 attribute changes. When a Job is removed from the Job History (see [ipp-mod] 4.3.7.1), no Event is
651 generated.

652
653 This Event value has the following sub-values: ‘job-created’, ‘job-completed’ and ‘job-purged’. A
654 client can listen for any of these sub-values if it doesn’t want to listen to all ‘job-state changes’.

655 **‘job-created’**: REQUIRED - the Printer has accepted a Job Creation operation and the job’s “time-
656 at-creation” attribute value is set (see [ipp-mod] section 4.3.14.1). The Printer puts the job in
657 the ‘pending’, ‘pending-held’ or ‘processing’ states..

658 **‘job-completed’**: REQUIRED - the job has reached one of the completed states, i.e., the value of
659 the job’s “job-state” attribute has changed to: ‘completed’, ‘aborted’, or ‘canceled’. The Job’s
660 “time-at-completed” and “date-time-at-completed” (if supported) attributes are set (see [ipp-
661 mod] section 4.3.14).. The Printer also sends this Event when a Job is removed with the Purge-
662 Job operation. In this case, the Event Notification MUST report the ‘job-state’ as ‘canceled’.

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

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

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

678 **5.3.2.2 Rules for Matching of Subscribed Events**

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

682 **5.3.2.2.1 Rules for Matching of Printer Events**

683 Suppose that the Printer causes Printer Event E to occur. For each Per-Job or Per-Printer Subscription S in
684 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 S, the
685 Printer **MUST** generate an Event Notification.

686 Consider the example. There are three Subscription Objects each with the Subscribed Printer Event
687 ‘printer-state-changed’. Subscription Object A is a Per-Printer Subscription Object. Subscription Object
688 B is a Per-Job Subscription Object for Job 1, and Subscription Object C is a Per-Job Subscription
689 Object for Job 2. When the Printer enters the ‘stopped’ state, the Printer sends an Event Notification to
690 the Notification Recipients of Subscription Objects A, B, and C because this is a Printer Event. Note if
691 Job 1 has already completed, the Printer would not send an Event Notification for its Subscription
692 Object.

693 **5.3.2.2.2 Rules for Matching of Job Events**

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

- 695 3. For each Per-Printer Subscription S in the Printer, if E equals a value of this attribute in S or E is a
696 sub-value of a value of this attribute in S, the Printer **MUST** generate an Event Notification.
- 697 4. For each Per-Job Subscription S associated with Job J, if E equals a value of this attribute in S or E
698 is a sub-value of a value of this attribute in S, the Printer **MUST** generate an Event Notification.
- 699 5. For each Per-Job Subscription S that is NOT associated Job J, if E equals a value of this attribute in
700 S or E is a sub-value of a value of this attribute in, the Printer **MUST NOT** generate an Event
701 Notification from S.

702 Consider the example: There are three Subscription Objects listening for the Job Event ‘job-completed’.
703 Subscription Object A is a Per-Printer Subscription Object. Subscription Object B is a Per-Job
704 Subscription Object for Job 1, and Subscription Object C is a Per-Job Subscription Object for Job 2. In
705 addition, Per-Printer Subscription Object D is listening for the Job Event ‘job-state-changed’. When Job
706 1 completes, the Printer sends an Event Notification to the Notification Recipient of Subscription
707 Object A (because it is Per-Printer) and Subscription Object B because it is a Per-Job Subscription
708 Object associated with the Job generating the Event. The Printer also sends an Event Notification to the
709 Notification Recipient of Subscription Object D because ‘job-completed’ is a sub-value of ‘job-state-
710 changed’ – the value that Subscription Object D is listening for. The Printer does not send an Event
711 Notification to the Notification Recipients of Subscription Object C because it is a Per-Job Subscription
712 Object associated with some Job other than the Job generating the Event.

713 **5.3.2.2.3 Special Cases for Matching Rules**

714 This section contains rule for special cases.

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

721 If an Event matches two values of this “notify-events” attribute in a single Subscription object (e.g., a value
722 and its sub-value), a Printer MAY send one Event Notification for each matched value in the Subscription
723 Object or it MAY send only one Event Notification per Subscription Object. The rules in sections 5.3.2.2.1
724 and 5.3.2.2.2 are purposefully ambiguous about the number of Event Notification sent when Event E
725 matches two or more values in a Subscription Object.

726 Consider the example: There are two Per-Printer Subscription Objects when a Job completes.
727 Subscription Object A has the Subscribed Job Event ‘job-state-changed’. Subscription Object B has the
728 Subscribed Job Events ‘job-state-changed’ and ‘job-completed’. The Printer sends an Event
729 Notification to the Notification Recipient of Subscription Object A with the value of ‘job-state-
730 changed’ for the “notify-subscribing-event” attribute. The Printer sends either one or two Event
731 Notifications to the Notification Recipient of Subscription Object B, depending on implementation. If it
732 sends two Event Notifications, one has the value of ‘job-state-changed’ for the “notify-subscribing-
733 event” attribute, and the other has the value of ‘job-completed’ for the “notify-subscribing-event”
734 attribute. If it sends one Event Notification, it has the value of either ‘job-state-changed’ or ‘job-
735 completed’ for the “notify-subscribing-event” attribute, depending on implementation. The algorithm
736 for choosing such a value is implementation dependent.

737 **5.3.3 notify-attributes (1setOf type2 keyword)**

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

741 A Printer MAY support this attribute.

742 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this
743 attribute in Subscription Creation Operation or the Printer does not support this attribute, the Subscription
744 Object MUST NOT contain the “notify-attributes” attribute. There is no “notify-attributes-default”
745 attribute.

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

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

- 754 a) a Subscription attribute, the Printer MUST use the attribute N in the Subscription Object that is
755 being used to generate the Event Notification.
- 756 b) a Job attribute and the Printer is generating an Event Notification from a Per-Job Subscription
757 Object S, the Printer MUST use the attribute N in the Job object associated with S.
- 758 c) a Job attribute and the Printer is generating an Event Notification from a Per-Printer Subscription
759 Object and the Event is:
- 760 • a Job Event, the Printer MUST use the attribute N in the Job object that caused the Event.
 - 761 • a Printer Event, the Printer MUST use the attribute N in the active Job.

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

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

766 **5.3.4 notify-user-data (octetString(63))**

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

- 769 • the identity of the Subscriber
- 770 • a path or index to some Subscriber information
- 771 • a key that identifies to the Notification Recipient the ultimate recipient of the Event Notification
- 772 • the id for a Notification Recipient that had previously registered with an Instant Messaging Service

773 A Printer MUST support this attribute.

774 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this
775 attribute in Subscription Creation Operation, the Subscription Object MUST NOT contain the “notify-user-
776 data” attribute. There is no “notify-user-data-default” attribute.

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

779 5.3.5 notify-charset (charset)

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

782 A Printer MUST support this attribute.

783 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this
784 attribute in Subscription Creation Operation or supplies an unsupported value, the Printer MUST populate
785 this attribute in the Subscription Object with the value of the “attributes-charset” operation attribute, which
786 is a REQUIRED attribute in all IPP requests (see [ipp-mod]). If the value of the “attributes-charset”
787 attribute is unsupported, the Printer MUST populate this attribute in the Subscription Object with the value
788 of the Printer’s “charset-configured” attribute. There is no “notify-charset-default” attribute.

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

791 5.3.6 notify-natural-language (naturalLanguage)

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

795 A Printer MUST support this attribute.

796 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this
797 attribute in Subscription Creation Operation or supplies an unsupported value, the Printer MUST populate
798 this attribute in the Subscription Object with the value of the “attributes-natural-language” operation
799 attribute, which is a REQUIRED attribute in all IPP requests (see [ipp-mod]). If the value of the “attributes-
800 natural-language” attribute is unsupported, the Printer MUST populate this attribute in the Subscription
801 Object with the value of the Printer’s “natural-language-configured” attribute. There is no “notify-natural-
802 language-default” attribute.

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

805 5.3.7 notify-lease-duration (integer(0:67108863))

806 This attribute specifies the duration of the lease associated with the Per-Printer Subscription Object at the
807 time the Subscription Object was created or the lease was renewed. The duration of the lease is infinite if
808 the value is 0, i.e., the lease never expires.

809 This attribute is not present on a Per-Job Subscription Object because the Subscription Object lasts exactly
810 as long as the associated Job object. See section 5.4.3 on “notify-lease-expiration-time (integer(0:MAX))”
811 for more details.

812 A Printer MUST support this attribute.

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

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

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

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

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

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

833 **5.3.8 notify-time-interval (integer(0:MAX))**

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

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

838 A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this
839 attribute, the Printer MUST not populate this attribute on the Subscription Object. There is no default
840 “notify-time-interval-default” attribute.

841 There is no “notify-time-interval-supported”. The value of this attribute MAY be any nonnegative integer
842 (0,MAX).

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

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

- 847 2. This attribute is present with a nonzero value of N:
- 848 a) If the Printer has not sent an Event Notification for the ‘job-progress’ Event for the associated
849 Subscription Object within the past N seconds, the Printer MUST send an Event Notification for
850 the Event that just occurred. Note when the Printer completes the first page of a Job, this rule
851 implies that the Printer sends an Event Notification for a Per-Job Subscription Objects.
- 852 b) Otherwise, the Printer MUST NOT generate or send an Event Notification for the associated
853 Subscription Object. The Printer MUST NOT increase the value of the “notify-sequence-
854 number” Subscription Object attribute (i.e., the sequence of values of the “notify-sequence-
855 number” attribute counts the Event Notifications that the Printer sent and not the Events that do
856 not cause an Event Notification to be sent).
- 857 It is RECOMMENDED that a Subscribing Client use this attribute when it subscribes to the ‘job-progress’
858 Event, and that the value be sufficiently large to limit the frequency with which the Printer sends Event
859 Notifications.

860 This attribute MUST not effect any Events other than ‘job-progress’.

861 5.4 Subscription Description Attributes

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

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

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

868 **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))

869 5.4.1 notify-subscription-id (integer (1:MAX))

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

872 A Printer MUST support this attribute.

873 The Printer SHOULD NOT assign the value of this attribute sequentially as it creates Subscription Objects.
874 Sequential assignment makes it easy for rogue clients to guess the value of this attribute on other
875 Subscription Objects.

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

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

881 **5.4.2 notify-sequence-number (integer (0:MAX))**

882 The value of this attribute indicates the number of times that the Printer has generated and attempted to
883 send an Event Notification. When an Event Notification contains this attribute, the Notification Recipient
884 can determine whether it missed some Event Notifications (i.e., numbers skipped) or received duplicates
885 (i.e., same number twice).

886 A Printer MUST support this attribute.

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

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

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

898 **5.4.3 notify-lease-expiration-time (integer(0:MAX))**

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

902 A Printer MUST support this attribute.

903 When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present – the
904 Subscription Object lasts exactly as long as the associated Job object.

905 When the Printer creates a Per-Printer Subscription Object, it populates this attribute with a value that is the
906 sum of the values of the Printer's "printer-up-time" attribute and the Subscription Object's "notify-lease-
907 duration" attribute with the following exception. If the value of the Subscription Object's "notify-lease-
908 duration" attribute is 0 (i.e., no expiration time), then the value of this attribute MUST be set to 0 (i.e., no
909 expiration time).

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

912 When the "printer-up-time" equals the value of this attribute, the Printer MUST delete the Subscription
913 Object. A client can extend a lease of a Per-Printer Subscription Object with the Renew-Subscription
914 operation (see section 11.2.5).

915 Note: In order to compute the number of seconds remaining in a lease for a Per-Printer Subscription Object,
916 a client can subtract the Subscription's "notify-printer-up-time" attribute (see section 5.4.4) from the
917 Subscription's "notify-lease-expiration-time" attribute.

918 **5.4.4 notify-printer-up-time (integer(1:MAX))**

919 This attribute is an alias for the Printer's "printer-up-time" attribute " (see [ipp-mod] section 4.4.29).

920 A Printer MUST support this attribute.

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

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

929 **5.4.5 notify-printer-uri (uri)**

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

931 A Printer MUST support this attribute.

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

935 **5.4.6 notify-job-id (integer(1:MAX))**

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

938 A Printer MUST support this attribute.

939 If this attribute is not present, the Subscription Object MUST be a Per-Printer Subscription. If this attribute
940 is present, the Subscription Object MUST be a Per-Job Subscription Object and this attribute MUST
941 identify the Job with which the Subscription Object is associated.

942 Note: This attribute could be useful to a Notification Recipient that receives an Event Notification
943 generated from a Per-Job Subscription Object and caused by a Printer Event. The Event Notification gives
944 access to the Printer and the Subscription Object. The Event Notification gives access to the associated Job
945 only via this attribute..

946 **5.4.7 notify-subscriber-user-name (name(MAX))**

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

948 A Printer MUST support this attribute.

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

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

956 **6 Printer Description Attributes Related to Notification**

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

960 **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

961 **6.1 printer-state-change-time (integer(1:MAX))**

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

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

966 On power-up, the Printer MUST set the value of this attribute to be the value of its “printer-up-time”
 967 attribute, so that it always has a value. Whenever the ‘printer-state-changed’ Printer Event occurs, the
 968 Printer MUST set this attribute to the value of the Printer’s “printer-up-time” attribute.

969 **6.2 printer-state-change-date-time (dateTime)**

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

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

974 On power-up, the Printer MUST set the value of this attribute to be the value of its “printer-current-time”
 975 attribute, so that it always has a value (see [ipp-mod] section 4.4.30 on “printer-current-time”). Whenever
 976 the ‘printer-state-changed’ Printer Event occurs, the Printer MUST set this attribute to the value of the
 977 Printer’s “printer-current-time” attribute.

978 **7 New Values for Existing Printer Description Attributes**

979 **7.1 operations-supported (1setOf type2 enum)**

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

982 **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

983 **8 Attributes Only in Event Notifications**

984 This section contains those attributes that exist only in Event Notifications.

985 **8.1 notify-subscribed-event (type2 keyword)**

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

988 This attribute **MUST** contain one of the values of the “notify-events” attribute in the Subscription Object,
989 i.e., one of the Subscribed Event values. Its value is the Subscribed Event that “matches” the Event that
990 caused the Printer to send this Event Notification. This Subscribed Event value may be identical to the
991 Event or the Event may be a sub-value of the Subscribed Event. For example, the ‘job-completed’ Event
992 (which is a sub-event of the ‘job-state-changed’ event) would cause the Printer to send an Event
993 Notification for either the ‘job-completed’ or ‘job-state-changed’ Subscribed Events and to send the ‘job-
994 completed’ or ‘job-state-changed’ value for this attribute, respectively. See section 5.3.2.2 for the
995 “matching” rules of Subscribed Events and for additional examples.

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

998 **8.2 notify-text (text(MAX))**

999 This attribute contains a Human Consumable text message (see section 9.2). This message describes the
1000 Event and is encoded as plain text, i.e., ‘text/plain’ with the charset specified by Subscription Object’s
1001 “notify-charset” attribute.

1002 The Delivery Method Document specifies whether the Printer includes this attribute in an Event
1003 Notification.

1004 **9 Event Notification Content**

1005 This section defines the Event Notification content that the Printer sends when an Event occurs.

1006 When an Event occurs, the Printer **MUST** find each Subscription object whose “notify-events” attribute
1007 “matches” the Event. See section 5.3.2.2 for details on “matching”. For each matched Subscription Object,
1008 the Printer **MUST** create an Event Notification with the content and format that the Delivery Method
1009 Document specifies. The content contains the value of attributes specified by the Delivery Method
1010 Document. The Printer obtains the values immediately after the Event occurs. For example, if the “printer-
1011 state” attribute changes from ‘idle’ to ‘processing’, the Event ‘printer-state-changed’ occurs and the Printer
1012 puts various attributes into the Event Notification, including “printer-up-time” and “printer-state” with the
1013 values that they have immediately after the Event occurs, i.e., the value of “printer-state” is ‘processing’.

1014 If two different Events occur simultaneously, or nearly so (e.g., “printer-up-time” has the same value for
1015 both), the Printer **MUST** create a separate Event Notification for each Event, even if the associated
1016 Subscription Object is the same for both Events. However, the Printer **MAY** combine these distinct Event
1017 Notifications into a single Compound Event Notification if the Delivery Method supports Compound Event
1018 Notifications For example, suppose that two nearly-simultaneously Events represent two successive
1019 ‘printer-state-changed’ Events, one from ‘idle’ to ‘processing’ and another from ‘processing’ to ‘stopped’.
1020 These two Events have the same name but are different instances of the Event. Then the Printer **MUST**
1021 create a separate Event Notification for each Event and **SHOULD** accurately report the “printer-state” of the
1022 first Event as ‘processing’ and the second Event as ‘stopped’.

1023 If a Subscription Object contains more than one Subscribed Event, and several Events occur in quick
1024 succession each matching a different Subscribed Event in the Subscription Object, the Printer MUST NOT
1025 generate a single Event Notification from several of these Events, but MAY combine distinct Event
1026 Notifications into a single Compound Event Notification if the Delivery Method supports Compound Event
1027 Notifications.

1028 After the Printer has created the Event Notification, the Printer delivers it via either a:

1029 Push Delivery Method: The Printer sends the Event Notification shortly after an Event occurs. For
1030 some Push Delivery Methods, the Notification Recipient MUST send a response; for others it
1031 MUST NOT send a response.

1032 Pull Delivery Method: The Printer saves Event Notifications for some event-lease time and expects
1033 the Notification Recipient to request Event Notifications. The Printer returns the Event Notifications
1034 in a response to such a request.

1035 If an error that meets the following conditions occurs, the Printer MUST cancel the Subscription Object.

1036 a) the error occurs during the sending of an Event Notification generated from Subscription Object S
1037 AND

1038 b) the error would continue to occur every time the Printer sends an Event Notification generated from
1039 Subscription Object S in the future.

1040 From example, if the address of the “notify-recipient-uri” of Subscription Object A references a non-
1041 existent target and the Printer determines this fact, it MUST delete Subscription Object A.

1042 The next two sections describe the values that a Printer sends in the content of Machine Consumable and
1043 Human Consumable Event Notifications, respectively.

1044 The tables in the sub-sections of this section contain the following columns:

1045 a) **Source Value:** the name of the attribute that supplies the value for the Event Notification.
1046 Asterisks in this field refer to a note below the table.

1047 b) **Sends:** if the Printer supports the value (column 1) on the Source Object (column 3) the
1048 Delivery Method MUST specify:

1049 **MUST:** that the Printer MUST send the value.

1050 **SHOULD:** either that the Printer MUST send the value or that the value is incompatible
1051 with the Delivery Method.

1052 **MAY:** that the Printer MUST, SHOULD, MAY, MUST NOT, SHOULD NOT, or NEED
1053 NOT send the value. The Delivery Method specifies the level of conformance for the Printer.

1054 c) **Source Object:** the object from which the source value comes. If the object is “Event
1055 Notification”, the Printer fabricates the value when it sends the Event Notification. See section
1056 8.

1057 9.1 Content of Machine Consumable Event Notifications

1058 This section defines the attributes that a Delivery Method **MUST** mention in a Delivery Method Document
1059 when specifying the Machine Consumable Event Notification’s contents.

1060 This document does not define the order of attributes in Event Notifications. However, Delivery Method
1061 Documents **MAY** define the order of some or all of the attributes.

1062 A Delivery Method Document **MUST** specify additional attributes (if any) that a Printer implementation
1063 sends in a Machine Consumable Event Notification.

1064 Notification Recipients **MUST** be able to accept Event Notifications containing attributes they do not
1065 recognize. What a Notification Recipient does with an unrecognized attribute is implementation-
1066 dependent. Notification Recipients **MAY** attempt to display unrecognized attributes anyway or **MAY**
1067 ignore them.

1068 The next three sections define the attributes in Event Notification Contents that are:

- 1069 a) for all Events
- 1070 b) for Job Events only
- 1071 c) for Printer Events only

1072 9.1.1 Event Notification Content Common to All Events

1073 This section lists the attributes that a Delivery Method **MUST** specify for all Events.

1074 Table 5 lists potential values in each Event Notification.

1075 **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

Source Value	Sends	Source Object
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

1076 * A Printer MUST send this value only if and only if it supports the Printer’s “printer-current-time”
1077 attribute.

1078 ** If the Subscription Object does not contain a “notify-user-data” attribute and the Delivery Method
1079 document REQUIRES the Printer to send the “notify-user-data” source value in the Event Notification, the
1080 Printer MUST send an octet-string of length 0.

1081 *** The last three rows represent additional attributes that a client MAY request via the “notify-attributes”
1082 attribute. A Printer MAY support the “notify-attributes” attribute. The Delivery Method MUST say that the
1083 Printer MUST, SHOULD, MAY, MUST NOT, SHOULD NOT, or NEED NOT support the “notify-
1084 attributes” attribute and specific values of this attribute. The Delivery Method MAY say that support for the
1085 “notify-attributes” is conditioned on support of the attribute by the Printer or it MAY say that Printer
1086 MUST support the “notify-attributes” attribute if the Printer supports the Delivery Method.

1087 9.1.2 Additional Event Notification Content for Job Events

1088 This section lists the additional attributes that a Delivery Method MUST specify for Job Events. See Table
1089 6.

1090 **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

1091 * The Printer MUST send the “job-impressions-completed” attribute in an Event Notification only for the
1092 combinations of Events and Subscribed Events shown in Table 7.

1093 **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’

1094

1095 **9.1.3 Additional Event Notification Content for Printer Events**

1096 This section lists the additional attributes that a Delivery Method **MUST** specify for Printer Events. See
1097 Table 8.

1098

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

1099 **9.2 Content of Human Consumable Event Notification**

1100 This section defines the information that a Delivery Method **MUST** mention in a Delivery Method
1101 Document when specifying the Human Consumable Event Notifications contents or the value of the
1102 “notify-text” attribute.

1103 Such a Delivery Method **MUST** specify the following information and a Printer **SHOULD** send it:

- 1104 a) the Printer name (see Table 9)
- 1105 b) the time of the Event (see Table 11)
- 1106 c) for Printer Events only:
 - 1107 i) the Event (see Table 10) and/or Printer state information (see Table 14)
- 1108 d) for Job Events only:
 - 1109 i) the job identity (see Table 12)
 - 1110 ii) the Event (see Table 10) and/or Job state information (see Table 13)

1111 The subsections of this section specify the attributes that a Printer **MUST** use to obtain this information.

1112 A Delivery Method Document **MUST** specify additional information (if any) that a Printer implementation
1113 sends in a Human Consumable Event Notification or in the “notify-text” attribute.

1114 A client **MUST NOT** request additional attributes via the “notify-attributes” attribute because this attribute
1115 works only for Machine Consumable Event Notifications.

1116 Notification Recipients **MUST NOT** expect to be able to parse the Human Consumable Event Notification
1117 contents or the value of the “notify-text” attribute.

1118 The next three sections define the attributes in Event Notification Contents that are:

- 1119 a) for all Events
- 1120 b) for Job Events only

c) for Printer Events only

9.2.1 Event Notification Content Common to All Events

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

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

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

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

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

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

Table 10 – Event Name in Event Notification Content

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

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

1147

Table 11 – Event Time in Event Notification Content

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

1148

1149 9.2.2 Additional Event Notification Content for Job Events

1150 This section lists the source of the additional information that a Delivery Method MUST specify for Job
1151 Events.

1152 Table 12 lists the source of the information for the job name. The “job-name” is likely more meaningful to
1153 a user than “job-id”.

1154

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

1155

1156 Table 13 lists the source of the information for the job state. If a Printer supports the “job-state-message”
1157 and “job-detailed-state-message” attributes, it SHOULD use those attributes for the job state information,
1158 otherwise, it should fabricate such information from the “job-state” and “job-state-reasons”. For some
1159 Events, a Printer MAY combine this information with Event information.

1160

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

1161 9.2.3 Additional Event Notification Content for Printer Events

1162 This section lists the source of the additional information that a Delivery Method MUST specify for Printer
1163 Events.

1164 Table 14 lists the source of the information for the printer state. If a Printer supports the “printer-state-
1165 message”, it SHOULD use that attribute for the job state information, otherwise it SHOULD fabricate such
1166 information from the “printer-state” and “printer-state-reasons”. For some Events, a Printer MAY combine
1167 this information with Event information.

1168

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

1169

10 Delivery Methods

1170 A Delivery Method is the mechanism, i.e., protocol, by which the Printer delivers an Event Notification to a
 1171 Notification Recipient. There are several potential Delivery Methods for Event Notifications, standardized,
 1172 as well as proprietary. This document does not define any of these delivery mechanisms. Each Delivery
 1173 Method MUST be defined in a Delivery Method Document that is separate from this document. New
 1174 Delivery Methods will be created as needed using an extension to the registration procedures defined in
 1175 [ipp-mod]. Such documents are registered with IANA (see section 13).

1176 The following sorts of Delivery Methods are expected:

- 1177 – The Notification Recipient polls for Event Notifications at intervals directed by the Printer
- 1178 – The Printer sends Event Notifications to the Notification Recipient using http as the transport.
- 1179 – The Printer sends an email message.

1180 This section specifies how to define a Delivery Method Document and what to put in such a document.

1181 A Delivery Method Document MUST contain an exact copy of the following paragraph, caption and table.
 1182 In addition, column 2 of the table in the Delivery Method Document MUST contain answers to questions in
 1183 column 1 for the Delivery Method. Also, the Delivery Method document MUST contain a reference to this
 1184 document and call that reference [ipp-ntfy] because the table contains an [ipp-ntfy] reference.

1185 If a Printer supports this Delivery Method, the following are its characteristics.

1186

Table 15 – Information about the Delivery Method

Document Method Conformance Requirement	Delivery Method Realization
1. What is the URL scheme name for the Delivery Method?	
2. Is the Delivery Method REQUIRED, RECOMMEND, or OPTIONAL for an IPP Printer to support?	

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

13. What are the additional Printer Description attributes and the conformance requirements thereof?	
--	--

1187

1188 **11 Operations for Notification**

1189 This section defines all of the operations for Notification. Section 7.1 assigns of the “operation-id” for each
1190 operation. The following two sub-sections define Subscription Creation Operations, and other operations.

1191 **11.1 Subscription Creation Operations**

1192 This section defines the Subscription Creation Operations. The first section on Create-Job-Subscriptions
1193 gives most of the information. The other Subscription Creation Operations refer to the section on Create-
1194 Job-Subscriptions, even though the Create-Job-Subscriptions operation is the only OPTIONAL operation in
1195 this document (see section 12).

1196 A Printer MUST support Create-Printer-Subscriptions and the Subscription Template Attributes Group in
1197 Job Creation operations. It MAY support Create-Job-Subscriptions operations.

1198 **11.1.1 Create-Job-Subscriptions Operation**

1199 The operation creates one or more Per-Job Subscription Objects. The client supplies one or more
1200 Subscription Template Attributes Groups each containing one or more of Subscription Template Attributes
1201 (defined in section 5.3).

1202 Except for errors, the Printer MUST create exactly one Per-Job Subscription Object from each Subscription
1203 Template Attributes Group in the request, even if the newly created Subscription Object would have
1204 identical behavior to some existing Subscription Object. The Printer MUST associate each newly created
1205 Per-Job Subscription Object with the target Job, which is specified by the “notify-job-id” operation
1206 attribute.

1207 The Printer MUST accept the request in any of the target job’s ‘not-completed’ states, i.e., ‘pending’,
1208 ‘pending-held’, ‘processing’, or ‘processing-stopped’. The Printer MUST NOT change the job’s “job-state”
1209 attribute because of this operation. If the target job is in any of the ‘completed’ states, i.e., ‘completed’,
1210 ‘canceled’, or ‘aborted’, then the Printer MUST reject the request and return the ‘client-error-not-possible’
1211 status code; the response MUST NOT contain any Subscription Attribute Groups.

1212 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [IPP-MOD] section 8.3)
1213 performing this operation MUST either be the job owner or have Operator or Administrator access rights
1214 for this Printer (see [IPP-MOD] sections 1 and 8.5). Otherwise the Printer MUST reject the operation and
1215 return: the ‘client-error-forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-authorized’ status
1216 code as appropriate.

1217 **11.1.1.1 Create-Job-Subscriptions Request**

1218 The following groups of attributes are part of the Create-Job-Subscriptions Request:

1219 Group 1: Operation Attributes

1220 Natural Language and Character Set:

1221 The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod]
1222 section 3.1.4.1.

1223

1224 Target:

1225 The “printer-uri” attribute which defines the target for this operation as described in [ipp-mod]
1226 section 3.1.5.

1227

1228 Requesting User Name:

1229 The “requesting-user-name” attribute *SHOULD* be supplied by the client as described in [ipp-mod]
1230 section 8.3.

1231

1232 notify-job-id (integer(1:MAX)):

1233 The client *MUST* supply this attribute and it *MUST* specify the Job object to associate the Per-Job
1234 Subscription with. The value of “notify-job-id” *MUST* be the value of the “job-id” of the associated
1235 Job object. If the client does not supply this attribute, the Printer *MUST* reject this request with a
1236 ‘client-error-bad-request’ status code.

1237 Group 2-N: Subscription Template Attributes

1238 For each occurrence of this group:

1239 The client *MUST* supply one or more Subscription Template Attributes in any order. See section
1240 5.3 for a description of each such attribute. See section 5.2 for details on processing these
1241 attributes.

1242 **11.1.1.2 Create-Job-Subscriptions Response**

1243 The Printer *MUST* return to the client the following sets of attributes as part of a Create-Job-Subscriptions
1244 response:

1245 Group 1: Operation Attributes

1246 Status Message:

1247 As defined in [ipp-mod].

1248

1249 The Printer can return any status codes defined in [ipp-mod] and section 16. The following is a
1250 description of the important status codes:

1251

1252 **successful-ok:** the Printer created all Subscription Objects requested.

- 1253 **successful-ok-ignored-subscriptions:** the Printer created some Subscription Objects requested
1254 but some failed. The Subscription Attributes Groups with a “notify-status-code” attribute are
1255 the ones that failed.
- 1256 **client-error-ignored-all-subscriptions:** the Printer created no Subscription Objects requested
1257 and all failed. The Subscription Attributes Groups with a “notify-status-code” attribute are
1258 the ones that failed
- 1259 **client-error-not-possible:** For this operation and other Per-Job Subscription operations, this
1260 error can occur because the specified Job has already completed.

1261

1262 Natural Language and Character Set:

1263 The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod]
1264 section 3.1.4.2.

1265

1266 Group 2: Unsupported Attributes

1267 See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes. This group does not
1268 contain any unsupported Subscription Template Attributes; they are returned in the Subscription
1269 Attributes Group (see below).

1270

1271 Group 3-N: Subscription Attributes

1272 These groups MUST be returned if and only if the “status-code” parameter returned in Group 1 has
1273 the values: ‘successful-ok’, ‘successful-ok-ignored-subscriptions’, or ‘client-error-ignored-all-
1274 subscriptions’.

1275

1276 See section 5.2 for details on the contents of each occurrence of this group.

1277 **11.1.2 Create-Printer-Subscriptions operation**

1278 The operation is identical to Create-Job-Subscriptions with exceptions noted in this section.

1279 The operation creates Per-Printer Subscription Objects instead of Per-Job Subscription Objects, and
1280 associates each newly created Per-Printer Subscription Object with the Printer specified by the operation
1281 target rather than with a specific Job.

1282 The Printer MUST accept the request in any of its states, i.e., ‘idle’, ‘processing’, or ‘stopped’. The Printer
1283 MUST NOT change its “printer-state” attribute because of this operation.

1284 Access Rights: To create Per-Printer Subscription Objects, the authenticated user (see [IPP-MOD] section
1285 8.3) performing this operation MUST have Operator or Administrator access rights for this Printer (see
1286 [IPP-MOD] sections 1 and 8.5). Otherwise, the Printer MUST reject the operation and return: the ‘client-
1287 error-forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

1288 **11.1.2.1 Create-Printer-Subscriptions Request**

1289 The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.1) except that the Operation
1290 Attributes group **MUST NOT** contain the “notify-job-id” attribute. If the client does supply the “notify-
1291 job-id” attribute, then the Printer **MUST** treat it as any other unsupported Operation attribute and **MUST**
1292 return it in the Unsupported Attributes group.

1293 **11.1.2.2 Create-Printer-Subscriptions Response**

1294 The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.2).
1295

1296 **11.1.3 Job Creation Operation – Extensions for Notification**

1297 This document extends the Job Creation operations to create Subscription Objects as a part of the operation.

1298 The operation is identical to Create-Job-Subscriptions with exceptions noted in this section.

1299 Unlike the Create-Job-Subscriptions operation, this operation associates the newly created Subscription
1300 Objects with the Job object created by this operation. The operation succeeds if and only if the Job creation
1301 succeeds. If the Printer does not create some or all of the requested Subscription Objects, the Printer **MUST**
1302 return a ‘successful-ok-ignored-subscriptions’ status-code instead of a ‘successful-ok’ status-code, but the
1303 Printer **MUST NOT** reject the operation because of a failure to create Subscription Objects.

1304 If the operation includes a Job Template group, the client **MUST** supply it after the Operation Attributes
1305 group and before the first Subscription Template Attributes Group.

1306 If a Printer does not support this Notification specification, then it **MUST** treat the Subscription Attributes
1307 Group like an unknown group and ignore it (see [ipp-mod] section 5.2.2). Because the Printer ignores the
1308 Subscription Attributes Group, it doesn’t return them in the response either, thus indicating to the client that
1309 the Printer doesn’t support Notification.

1310 Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [IPP-MOD] section 8.3)
1311 performing this operation **MUST** either have permission to create Jobs on the Printer. Otherwise the Printer
1312 **MUST** reject the operation and return: the ‘client-error-forbidden’, ‘client-error-not-authenticated’, or
1313 ‘client-error-not-authorized’ status code as appropriate.

1314 **11.1.3.1 Job Creation Request**

1315 The groups for this operation are sufficiently different from the Create-Job-Subscriptions operation that
1316 they are all presented here. The following groups of attributes are supplied as part of a Job Creation
1317 Request:

1318 Group 1: Operation Attributes

1319 Same as defined in [ipp-mod] for Print-Job, Print-URI, and Create-Job requests.

- 1320 Group 2: Job Template Attributes
- 1321 The client **OPTIONALLY** supplies a set of Job Template attributes as defined in [ipp-mod] section
1322 4.2.
- 1323 Group 3 to N: Subscription Template Attributes
- 1324 The same as Group 2-N in Create-Job-Subscriptions. See section 11.1.1.1.
- 1325 Group N+1: Document Content (Print-Job only)
- 1326 The client **MUST** supply the document data to be processed.
- 1327 **11.1.3.2 Job Creation Response**
- 1328 The Printer **MUST** return to the client the following sets of attributes as part of a Print-Job, Print-URI, and
1329 Create-Job Response:
- 1330 Group 1: Operation Attributes
- 1331
- 1332 Status Message:
- 1333
- 1334 As defined in [ipp-mod] for Print-Job, Print-URI, and Create-Job requests.
- 1335
- 1336 The Printer can return any status codes defined in [ipp-mod] and section 16. The following is a
1337 description of the important status codes:
- 1338
- 1339 **successful-ok:** the Printer created the Job and all Subscription Objects requested.
- 1340 **successful-ok-ignored-subscriptions:** the Printer created the Job and not all of the Subscription
1341 Objects requested. This status-code hides ‘successful-ok-xxx’ status-codes that could reveal
1342 problems in Job creation. The Printer **MUST** not return the ‘client-error-ignored-all-
1343 subscriptions’ status code for Job Creation operations because the Printer returns an error
1344 status-code only when it fails to create a Job.
- 1345
- 1346 Natural Language and Character Set:
- 1347 The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod]
1348 section 3.1.4.2.
- 1349
- 1350 Group 2: Unsupported Attributes
- 1351 See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes. This group does not
1352 contain any unsupported Subscription Template Attributes; they are returned in the Subscription
1353 Attributes Group (see below).
- 1354
- 1355 Group 3: Job Object Attributes
- 1356 As defined in [ipp-mod] for Print-Job, Print-URI, and Create-Job requests.
- 1357

1358 Group 4 to N: Subscription Attributes

1359 These groups **MUST** be returned if and only if the client supplied Subscription Template Attributes
1360 and the operation was accepted.

1361

1362 See section 5.2 for details on the contents of each occurrence of this group.

1363

1364 **11.2 Other Operations**

1365 This section defines other operations on Subscription objects.

1366 **11.2.1 Validate-Job Operation - Extensions for Notification**

1367 A client can test whether one or more Subscription Objects could be created using the Validate-Job
1368 operation. The client supplies one or more Subscription Template Attributes Groups (defined in section
1369 5.3), just as in a Job Creation request.

1370 A Printer **MUST** support this extension to this operation.

1371 The Printer **MUST** accept requests that are identical to the Job Creation request defined in section 11.1.3.1,
1372 except that the request **MUST** not contain document data.

1373 The Printer **MUST** return the same groups and attributes as the Print-Job operation (section 11.1.3.1) with
1374 the following exceptions. The Printer **MUST NOT** return a Job Object Attributes Group because no Job is
1375 created. The Printer **MUST NOT** return the “notify-subscription-id” attribute in any Subscription Attribute
1376 Group because no Subscription Object is created.

1377 If the Printer would succeed in creating a Subscription Object, the corresponding Subscription Attributes
1378 Group either has no ‘status-code’ attribute or a ‘status-code’ attribute with a value of ‘successful-ok-too-
1379 many-events’ or ‘successful-ok-ignored-or-substituted-attributes’ (see sections 5.2 and 17). The status-
1380 codes have the same meaning as in Job Creation except the results state what “would happen”.

1381 The Printer **MUST** validate Subscription Template Attributes Groups in the same manner as the Job
1382 Creation operations.

1383 **11.2.2 Get-Printer-Attributes - Extensions for Notification**

1384 This operation is extended so that it returns Printer attributes defined in this document.

1385 A Printer **MUST** support this extension to this operation.

1386 In addition to the requirements of [ipp-mod] section 3.2.5, a Printer **MUST** support the following additional
1387 values for the “requested-attributes” Operation attribute in this operation and return such attributes in the
1388 Printer Object Attributes group of its response.

1389 1. **Subscription Template Attributes:** Each supported attribute in column 2 of Table 1.

- 1390 2. **New Printer Description Attributes:** Each supported attribute in section 6.
- 1391 3. **New Group Name:** The ‘subscription-template’ group name, which names all supported
1392 Subscription Template Attribute in column 2 of Table 1. This group name is also used in the Get-
1393 Subscription-Attributes and Get-Subscriptions operation with an analogous meaning.
- 1394 4. **Extended Group Name:** The ‘all’ group name, which names all Printer attributes according to
1395 [ipp-mod] section 3.2.5. In this extension ‘all’ names all attributes specified in [ipp-mod] plus those
1396 named in items 1 and 2 of this list.

1397

1398 **11.2.3 Get-Subscription-Attributes operation**

1399 This operation allows a client to request the values of the attributes of a Subscription Object.

1400 A Printer MUST support this operation.

1401 This operation is almost identical to the Get-Job-Attributes operation (see [ipp-mod] section 3.3.4). The
1402 only differences are that the operation is directed at a Subscription Object rather than a Job object, and the
1403 returned attribute group contains Subscription Object attributes rather than Job object attributes.

1404 **11.2.3.1 Get-Subscription-Attributes Request**

1405 The following groups of attributes are part of the Get-Subscription-Attributes request:

1406 Group 1: Operation Attributes

1407 Natural Language and Character Set:

1408 The “attributes-charset” and “attributes-natural-language” attributes as described in section [ipp-
1409 mod] 3.1.4.1.

1410

1411 Target:

1412 The “printer-uri” attribute which defines the target for this operation as described in [ipp-mod]
1413 section 3.1.5.

1414

1415 “notify-subscription-id” (integer (1:MAX)):

1416 The client MUST supply this attribute. The Printer MUST support this attribute. This attribute
1417 specifies the Subscription Object from which the client is requesting attributes. If the client omits
1418 this attribute, the Printer MUST reject this request with the ‘client-error-bad-request’ status code.

1419

1420 Requesting User Name:

1421 The “requesting-user-name” attribute SHOULD be supplied by the client as described in [ipp-mod]
1422 section 8.3.

1423

1424 “requested-attributes” (1setOf keyword):
1425 The client **OPTIONALLY** supplies this attribute. The Printer **MUST** support this attribute. This
1426 attribute specifies the attributes of the specified Subscription Object that the Printer **MUST** return in
1427 the response. Each value of this attribute is either an attribute name (defined in sections 5.3 and 5.4)
1428 or an attribute group name. The attribute group names are:
1429
1430 - ‘subscription-template’: all attributes that are both defined in section 5.3 and present on the
1431 specified Subscription Object (column 1 of Table 1).
1432 - ‘subscription-description’: all attributes that are both defined in section 5.4 and present on the
1433 specified Subscription Object (Table 2).
1434 - ‘all’: all attributes that are present on the specified Subscription Object.
1435
1436 A Printer **MUST** support all these group names.
1437
1438 If the client omits this attribute, the Printer **MUST** respond as if this attribute had been supplied with
1439 a value of ‘all’.

1438 **11.2.3.2 Get-Subscription-Attributes Response**

1439 The Printer returns the following sets of attributes as part of the Get-Subscription-Attributes Response:

1440 Group 1: Operation Attributes

1441 Status Message:

1442 Same as [ipp-mod].
1443

1444 Natural Language and Character Set:

1445 The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod]
1446 section 3.1.4.2. The “attributes-natural-language” **MAY** be the natural language of the Subscription
1447 Object, rather than the one requested.
1448

1449 Group 2: Unsupported Attributes

1450 See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes.
1451

1452 The response **NEED NOT** contain the “requested-attributes” operation attribute with any supplied
1453 values (attribute keywords) that were requested by the client but are not supported by the Printer. If
1454 the Printer does return unsupported attributes referenced in the “requested-attributes” operation
1455 attribute and that attribute included group names, such as ‘all’, the unsupported attributes **MUST**
1456 **NOT** include attributes described in the standard but not supported by the implementation.
1457

1458 Group 3: Subscription Attributes

1459 This group contains a set of attributes with their current values. Each attribute in this group:

1460 a) **MUST** be specified by the “requested-attributes” attribute in the request, **AND**

1461 b) **MUST** be present on the specified Subscription Object **AND**

1462 c) MUST NOT be restricted by the security policy in force. For example, a Printer MAY prohibit
1463 a client who is not the creator of a Subscription Object from seeing some or all of its attributes.
1464 See [ipp-mod] section 8.

1465 The Printer can return the attributes of the Subscription Object in any order. The client MUST
1466 accept the attributes in any order.

1467 **11.2.4 Get-Subscriptions operation**

1468 This operation allows a client to retrieve the values of attributes of all Subscription Objects belonging to a
1469 Job or Printer.

1470 A Printer MUST supported this operation.

1471 This operation is similar to the Get-Subscription-Attributes operation, except that this Get-Subscriptions
1472 operation returns attributes from possibly more than one object.

1473 This operation is similar to the Get-Jobs operation (see [ipp-mod] section 3.2.6), except that the operation
1474 returns Subscription Objects rather than Job objects.

1475 **11.2.4.1 Get-Subscriptions Request**

1476 The following groups of attributes are part of the Get-Subscriptions request:

1477 Group 1: Operation Attributes

1478 Natural Language and Character Set:

1479 The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod]
1480 section 3.1.4.1.

1481

1482 Target:

1483 The “printer-uri” attribute which defines the target for this operation as described in [ipp-mod]
1484 section 3.1.5.

1485

1486 Requesting User Name:

1487 The “requesting-user-name” attribute SHOULD be supplied by the client as described in [ipp-mod]
1488 section 8.3.

1489

1490 “notify-job-id” (integer(1:MAX)):

1491 If the client specifies this attribute, the Printer returns the specified attributes of all Per-Job
1492 Subscription Objects associated with the Job whose “job-id” attribute value equals the value of this
1493 attribute. If the client does not specify this attribute, the Printer returns the specified attributes of all
1494 Per-Printer Subscription Objects. Note: there is no way to get all Per-Job Subscriptions.

1495

1496 “limit” (integer(1:MAX)):

1497 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. It is an
1498 integer value that determines the maximum number of Subscription Objects that a client will receive

1499 from the Printer even if the “my-subscriptions” attribute constrains which Subscription Objects are
1500 returned. The limit is a “stateless limit” in that if the value supplied by the client is ‘N’, then only
1501 the first ‘N’ Subscription Objects are returned in the Get-Subscriptions Response. There is no
1502 mechanism to allow for the next ‘M’ Subscription Objects after the first ‘N’ Subscription Objects.
1503 If the client does not supply this attribute, the Printer responds with all applicable Subscription
1504 Objects.

1505
1506 “requested-attributes” (1setOf type2 keyword):

1507 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. This
1508 attribute specifies the attributes of the specified Subscription Objects that the Printer MUST return
1509 in the response. Each value of this attribute is either an attribute name (defined in sections 5.3 and
1510 5.4) or an attribute group name (defined in section 11.2.3.1). If the client omits this attribute, the
1511 Printer MUST respond as if the client had supplied this attribute with the one value: ‘notify-
1512 subscription-id’.

1513
1514 “my-subscriptions” (boolean):

1515 The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. If the
1516 value is ‘false’, the Printer MUST consider the Subscription Objects from all users as candidates. If
1517 the value is ‘true’, the Printer MUST return the Subscription Objects created by the requesting user
1518 of this request. If the client does not supply this attribute, the Printer MUST respond as if the client
1519 had supplied the attribute with a value of ‘false’. The means for authenticating the requesting user
1520 and matching the Subscription Objects is similar to that for Jobs which is described in [ipp-mod]
1521 section 8.

1522 **11.2.4.2 Get-Subscriptions Response**

1523 The Printer returns the following sets of attributes as part of the Get-Subscriptions Response:

1524 Group 1: Operation Attributes

1525 Status Message:

1526 Same as [ipp-mod].
1527

1528 Natural Language and Character Set:

1529 The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod]
1530 section 3.1.4.2.
1531

1532 Group 2: Unsupported Attributes

1533 Same as for Get-Subscription-Attributes.
1534

1535 Groups 3 to N: Subscription Attributes

1536 The Printer responds with one Subscription Attributes Group for each requested Subscription Object
1537 (see the “notify-job-id” attribute in the Operation Attributes Group of this operation).
1538

1539 The Printer returns Subscription Objects in any order.

1540
1541 If the “limit” attribute is present in the Operation Attributes group of the request, the number of
1542 Subscription Attributes Groups in the response MUST NOT exceed the value of the “limit”
1543 attribute.

1544
1545 If there are no Subscription Objects associated with the specified Job or Printer, the Printer MUST
1546 return zero Subscription Attributes Groups and it MUST NOT treat this case as an error, i.e., the
1547 status-code MUST be ‘successful-ok’ unless something else causes the status code to have some
1548 other value.

1549
1550 See the Group 3 response (Subscription Attributes Group) of the Get-Subscription-Attributes
1551 operation (section 11.2.3.2) for the attributes that a Printer returns in this group.
1552

1553 **11.2.5 Renew-Subscription operation**

1554 This operation allows a client to request the Printer to extend the lease on a Per-Printer Subscription Object.

1555 The Printer MUST support this operation.

1556 The Printer MUST accept this request for a Per-Printer Subscription Object in any of the target Printer’s
1557 states, i.e., ‘idle’, ‘processing’, or ‘stopped’, but MUST NOT change the Printer’s “printer-state” attribute.

1558 The Printer MUST reject this request for a Per-Job Subscription Object because it has no lease (see section
1559 5.4.3). The status code returned MUST be ‘client-error-not-possible’.

1560 *Access Rights:* The authenticated user (see [IPP-MOD] section 8.3) performing this operation MUST either
1561 be the owner of the Per-Printer Subscription Object or have Operator or Administrator access rights for the
1562 Printer (see [IPP-MOD] sections 1 and 8.5). Otherwise, the Printer MUST reject the operation and return:
1563 the ‘client-error-forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-authorized’ status code as
1564 appropriate.

1565 **11.2.5.1 Renew-Subscription Request**

1566 The following groups of attributes are part of the Renew-Subscription Request:

1567 Group 1: Operation Attributes

1568 Natural Language and Character Set:

1569 The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod]
1570 section 3.1.4.1.

1571 Target:

1572 The “printer-uri” attribute which defines the target for this operation as described in [ipp-mod]
1573 section 3.1.5.
1574
1575

1576 “notify-subscription-id” (integer (1:MAX)):
1577 The client MUST supply this attribute. The Printer MUST support this attribute. This attribute
1578 specifies the Per-Printer Subscription Object whose lease the Printer MUST renew. If the client
1579 omits this attribute, the Printer MUST reject this request with the ‘client-error-bad-request’ status
1580 code.

1581
1582 Requesting User Name:
1583 The “requesting-user-name” (name(MAX)) attribute SHOULD be supplied by the client as
1584 described in [ipp-mod] section 8.3.
1585

1586 Group 2: Subscription Template Attributes

1587
1588 “notify-lease-duration” (integer(0:MAX)):
1589 The client MAY supply this attribute. It indicates the number of seconds to renew the lease for the
1590 specified Subscription Object. A value of 0 requests an infinite lease (which MAY require Operator
1591 access rights). If the client omits this attribute, the Printer MUST use the value of the Printer’s
1592 “notify-lease-duration-default” attribute. See section 5.3.7 for more details.

1593 11.2.5.2 Renew-Subscription Response

1594 The Printer returns the following sets of attributes as part of the Renew-Subscription Response:

1595 Group 1: Operation Attributes

1596 Status Message:
1597 Same as [ipp-mod].
1598

1599 The following are some of the status codes returned:

1600
1601 **successful-ok:** The operation successfully renewed the lease on the Subscription Object for the
1602 requested duration..

1603 **successful-ok-ignored-or-substituted-attributes:** The operation successfully renewed the lease on
1604 the Subscription Object for some duration other than the amount requested.

1605 **client-error-not-possible:** The operation failed because the “notify-subscription-id” Operation
1606 attribute identified a Per-Job Subscription Object.

1607 **client-error-not-found:** The operation failed because the “notify-subscription-id” Operation
1608 attribute identified a non-existent Subscription Object.
1609

1610 Natural Language and Character Set:

1611 The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod]
1612 section 3.1.4.2. The “attributes-natural-language” MAY be the natural language of the Subscription
1613 Object, rather than the one requested.
1614

1615 Group 2: Unsupported Attributes

1616 See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes.

1617
1618 Group 3: Subscription Attributes
1619 The Printer MUST return the following Subscription Attribute:

1620 “notify-lease-duration” (integer(0:MAX)):
1621 The value of this attribute MUST be the number of seconds that the Printer has granted for the lease
1622 of the Subscription Object (see section 5.3.7 for details, such as the value of this attribute when the
1623 Printer doesn’t support the requested value).
1624
1625

1626 **11.2.6 Cancel-Subscription operation**

1627 This operation allows a client to delete a Subscription Object and stop the Printer from sending more Event
1628 Notifications. Once performed, there is no way to reference the Subscription Object.

1629 A Printer MUST supported this operation.

1630 The Printer MUST accept this request in any of the target Printer’s states, i.e., ‘idle’, ‘processing’, or
1631 ‘stopped’, but MUST NOT change the Printer’s “printer-state” attribute.

1632 If the specified Subscription Object is a Per-Job Subscription Object, the Printer MUST accept this request
1633 in any of the target Job’s states, but MUST NOT change the Job’s “job-state” attribute or affect the Job.

1634 *Access Rights:* The authenticated user (see [IPP-MOD] section 8.3) performing this operation MUST either
1635 be the owner of the Subscription Object or have Operator or Administrator access rights for the Printer (see
1636 [IPP-MOD] sections 1 and 8.5). Otherwise, the Printer MUST reject the operation and return: the ‘client-
1637 error-forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

1638 Note: There is no way to change any attributes on a Subscription Object, except the “notify-lease-
1639 duration” attribute (using the Renew-Subscription operation). In order to change other attributes, a client
1640 performs a Subscription Creation Operation and Cancel-Subscription operation on the old Subscription
1641 Object. If the client wants to avoid missing Event Notifications, it performs the Subscription Creation
1642 Operation first. If this order would create too many Subscription Objects on the Printer, the client reverses
1643 the order.

1644 **11.2.6.1 Cancel-Subscription Request**

1645 The following groups of attributes are part of the Cancel-Subscription Request:

1646 Group 1: Operation Attributes

1647 Natural Language and Character Set:
1648 The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod]
1649 section 3.1.4.1.
1650

1651 Target:
1652 The “printer-uri” attribute which defines the target for this operation as described in [ipp-mod]
1653 section 3.1.5.
1654
1655 “notify-subscription-id” (integer (1:MAX)):
1656 The client **MUST** supply this attribute. The Printer **MUST** support this attribute. This attribute
1657 specifies the Subscription Object that the Printer **MUST** cancel. If the client omits this attribute, the
1658 Printer **MUST** reject this request with the ‘client-error-bad-request’ status code.
1659
1660 Requesting User Name:
1661 The “requesting-user-name” attribute **SHOULD** be supplied by the client as described in [ipp-mod]
1662 section 8.3.
1663

1664 11.2.6.2 Cancel-Subscription Response

1665 The Printer returns the following sets of attributes as part of the Cancel-Subscription Response:

1666 Group 1: Operation Attributes

1667 Status Message:
1668 Same as [ipp-mod].
1669

1670 The following are some of the status codes returned:

1671
1672 **successful-ok:** The operation successfully canceled (deleted) the Subscription Object..
1673 **client-error-not-found:** The operation failed because the “notify-subscription-id” Operation
1674 attribute identified a non-existent Subscription Object.
1675

1676 Natural Language and Character Set:

1677 The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod]
1678 section 3.1.4.2. The “attributes-natural-language” **MAY** be the natural language of the Subscription
1679 Object, rather than the one requested.
1680

1681 Group 2: Unsupported Attributes

1682 See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes.
1683

1684 12 Conformance Requirements

1685 It is **OPTIONAL** to implement this Event Notification specification.

1686 If this Event Notification specification is implemented, Printers **MUST**:

1687 1. meet the Conformance Requirements detailed in section 5 of [ipp-mod].

- 1688 2. support all of the following attributes:
- 1689 a. REQUIRED Subscription Object attributes in section 5.
- 1690 b. REQUIRED Printer Description object attributes in section 6.
- 1691 c. REQUIRED attributes in Event Notification content in section 8.
- 1692 3. send Event Notifications that conform to the requirements of the Delivery Method Document for each
- 1693 supported Delivery Method (the conformance requirements for Delivery Method Documents is
- 1694 specified in section 10).
- 1695 4. support all operations as described in Table 16:

1696 **Table 16 – Conformance Requirements for Operations**

Attribute	Conformance requirements
Subscription Attributes Group	REQUIRED
Create-Printer-Subscriptions (section 11.1.2)	REQUIRED
Create-Job-Subscriptions (section 11.1.1)	OPTIONAL
Get-Subscription-Attributes (section 11.2.2)	REQUIRED
Get-Subscriptions (section 11.2.4)	REQUIRED
Renew-Subscription (section 11.2.5)	REQUIRED
Cancel-Subscription (section 11.2.6)	REQUIRED

1697

1698 13 IANA Considerations

1699 This section describes the procedures for registering Event Notification Delivery Method proposals with
 1700 IANA to be used with this document. Such Delivery Method proposals can be IETF standards track
 1701 documents or vendor-defined documents. In either case, they will be registered with IANA using
 1702 procedures that extend those defined in [ipp-mod] section 6 and 11.

1703 These extension procedures are aligned with the guidelines as set forth by the IESG [IANA-CON]. Section
 1704 13.1 defines the format and content for new registrations for consideration. IANA will reject registration
 1705 proposals that leave out required information or do not follow the appropriate format described in Section
 1706 13.1.

1707 Implementers can, at any time, define new Event Notification Delivery Methods by proposing the complete
 1708 specification to IANA:

1709 iana@iana.org

1710 or by filling out the appropriate form on the IANA web pages (<http://www.iana.org>).

1711 IANA will forward the registration proposal to the IPP Designated Expert who will review the proposal
1712 with a mailing list that the Designated Expert keeps for this purpose. Initially, that list will be the mailing
1713 list used by the IPP WG:

1714 ipp@pwg.org

1715 even after the IPP WG is disbanded as permitted by [IANA-CON]. The IPP Designated Expert is appointed
1716 by the IESG Area Director responsible for IPP, according to [IANA-CON].

1717 When a Delivery Method Document is approved, the IPP Designated Expert becomes the point of contact
1718 for any future maintenance that might be required for that registration.

1719 **13.1 Format and Requirements for IPP Delivery Method Registration Proposals**

1720 This section defines the format and requirements for an IPP Event Notification Delivery Method
1721 Registration Proposal. A Delivery Method Registration Proposal:

1722 1. MUST contain the following information:

1723 Type of registration: IPP Event Notification Delivery Method

1724 Name of this delivery method:

1725 Proposed URL scheme name of this delivery method:

1726 Name of proposer:

1727 Address of proposer:

1728 Email address of proposer:

1729 Is this delivery method REQUIRED or OPTIONAL for conformance to the IPP Event Notification
1730 Specification document:

1731 Is this delivery method defining Machine Consumable and/or Human Consumable content:

1732 2. MUST meet the conformance requirements for Delivery Method Documents specified in section 10.

1733

1734 **14 Internationalization Considerations**

1735 This IPP Notification specification continues support for the internationalization of [ipp-mod] of attributes
1736 containing text strings and names. Allowing a Subscribing Client to specify a different natural language
1737 and charset for each Subscription Object increases the internationalization support.

1738 The Printer MUST be able to localize the content of Human Consumable Event Notifications and to
1739 localize the value of “notify-text” attribute in Machine Consumable Event Notifications that it sends to
1740 Notification Recipients. For localization, the Printer MUST use the value of the “notify-charset” attribute
1741 and the “notify-natural-language” attribute in the Subscription Object supplied by the Subscribing Client.

15 Security Considerations

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

Clients submitting Notification requests to the IPP Printer has the same security issues as submitting an IPP/1.1 print job request. The same mechanisms used by IPP/1.1 can therefore be used by the client Notification submission. Operations that require authentication can use the HTTP authentication. Operations that require privacy can use the HTTP/TLS privacy.

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

The standard security concerns (delivery to the right user, privacy of content, tamper proof content) apply to the Delivery Method. IPP should use the security mechanism of the Delivery Method used. Some delivery mechanisms are more secure than others. Therefore, sensitive Event Notifications should use the Delivery Method that has the strongest security.

16 Status Codes

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

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

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

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

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

1776 subscriptions' and either 'successful-ok-ignored-or-substituted-attributes' and/or 'successful-ok-conflicting-
1777 attributes', it MUST return 'successful-ok-ignored-subscriptions'.

1778 **16.2 client-error-ignored-all-subscriptions (0x0414)**

1779 This status code is the same as 'successful-ok-ignored-subscriptions' except that only the Create-Job-
1780 Subscriptions and Create-Printer-Subscriptions operation return it. They return this status code only when
1781 the Printer creates zero Subscription Objects.

1782 **17 Status Codes in Subscription Attributes Groups**

1783 This section contains values of the "notify-status-code" attribute that the Printer returns in a Subscription
1784 Attributes Group in a response when the corresponding Subscription Object:

- 1785 1. is not created or
- 1786 2. is created and some of the client-supplied attributes are not supported.

1787 The following sections are ordered in decreasing order of importance of the status-codes.

1788 **17.1 client-error-uri-scheme-not-supported (0x040C)**

1789 This status code is defined in [ipp-mod]. This document extends its meaning and allows it to be in a
1790 Subscription Attributes Group of a response.

1791 The scheme of the client-supplied URI in a "notify-recipient-uri" Subscription Template Attribute in a
1792 Subscription Creation Operation is not supported. See section 5.3.1.

1793 **17.2 client-error-too-many-subscriptions (0x0415)**

1794 The number of Subscription Objects supported by the Printer would be exceeded if this Subscription Object
1795 were created (see section 5.2).

1796 **17.3 successful-ok-too-many-events (0x0005)**

1797 The client supplied more Events in the "notify-events" operation attribute of a Subscription Creation
1798 Operation than the Printer supports, as indicated in its "notify-max-events-supported" Printer attribute (see
1799 section 5.3.2).

1800 **17.4 successful-ok-ignored-or-substituted-attributes (0x0001)**

1801 This status code is defined in [ipp-mod]. This document extends its meaning to include unsupported
1802 Subscription Template Attributes and it can appear in a Subscription Attributes Group.

18 Encodings of Additional Attribute Tags

This section assigns values to two attributes tags as extensions to the encoding defined in [ipp-pro]).

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

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

The following table specifies the values for the delimiter tags:

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

19 References

[IANA-CON]

Narte, T. and Alvestrand, H.T.: Guidelines for Writing an IANA Considerations Section in RFCs, Work in Progress, draft-iesg-iana-considerations-04.txt, May 21, 1998.

[ipp-mod]

deBry, R., , Hastings, T., Herriot, R., Isaacson, S., Powell, P., “Internet Printing Protocol/1.1: Model and Semantics”, <draft-ietf-ipp-model-v11-07.txt>, work in progress, May 22, 2000.

[ipp-not-req]

deBry, R., Lewis, H., Hastings, T., “Internet Printing Protocol/1.1: Requirements for IPP Notifications”, <draft-ietf-ipp-not-04.txt>, work in progress, July 6, 2000.

[ipp-pro]

Herriot, R., Butler, S., Moore, P., Tuner, R., “Internet Printing Protocol/1.1: Encoding and Transport”, <draft-ietf-ipp-protocol-v11-06.txt>, work in progress, May 30, 2000.

[ipp-prog]

Hastings, T., Bergman, R., Lewis, H., “IPP Job Progress Attributes”, <draft-ietf-ipp-job-prog-00.txt> work in progress, July 6, 2000.

[ipp-set]

Kugler, C., , Hastings, T., Herriot, R., Lewis, H., “Internet Printing Protocol (IPP): Job and Printer Set Operations”, <draft-ietf-ipp-job-printer-set-ops-02.txt>, work in progress, March 23, 2000.

[RFC2026]

S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

- 1831 [RFC2119]
1832 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March 1997
- 1833 [RFC2566]
1834 deBry, R., , Hastings, T., Herriot, R., Isaacson, S., Powell, P., "Internet Printing Protocol/1.0: Model
1835 and Semantics", RFC 2566, April 1999.
- 1836 [RFC2567]
1837 Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.
- 1838 [RFC2568]
1839 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",
1840 RFC 2568, April 1999.
- 1841 [RFC2569]
1842 Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC
1843 2569, April 1999.

1844 **20 Author's Addresses**

- 1845 Scott A. Isaacson (Editor)
1846 Novell, Inc.
1847 122 E 1700 S
1848 Provo, UT 84606
1849
1850 Phone: 801-861-7366
1851 Fax: 801-861-2517
1852 e-mail: sisaacson@novell.com
1853
- 1854 Tom Hastings
1855 Xerox Corporation
1856 737 Hawaii St. ESAE 231
1857 El Segundo, CA 90245
1858
1859 Phone: 310-333-6413
1860 Fax: 310-333-5514
1861 e-mail: hastings@cp10.es.xerox.com
1862
- 1863 Robert Herriot
1864 Xerox Corporation
1865 3400 Hillview Ave., Bldg #1
1866 Palo Alto, CA 94304
1867
1868 Phone: 650-813-7696

1869 Fax: 650-813-6860
1870 Email: robert.herriot@pahv.xerox.com

1871
1872 Roger deBry
1873 Utah Valley State College
1874 Orem, UT 84058

1875
1876 Phone: (801) 222-8000
1877 EMail: debryro@uvsc.edu

1878
1879 Jay Martin
1880 e-mail: jk@underscore.com

1881
1882 Michael Shepherd
1883 Xerox Corporation
1884 800 Phillips Road MS 128-51E
1885 Webster, NY 14450

1886
1887 Phone: 716-422-2338
1888 Fax: 716-265-8871
1889 e-mail: mshepherd@crt.xerox.com

1890
1891 Ron Bergman (Editor)
1892 Hitachi Koki Imaging Solutions
1893 1757 Tapo Canyon Road
1894 Simi Valley, CA 93063-3394
1895
1896 Phone: 805-578-4421
1897 Fax: 805-578-4001
1898 Email: rbergma@hitachi-hkis.com

1899 **A. Appendix - Model for Notification with Cascading Printers**

1900 With this model (see Figure 2), there is an intervening Print server between the human user and the output-
1901 device. So the system effectively has two Printers. There are two cases to consider.

- 1902 1. When the Printer 1 (in the server) generates Events, the system behaves like the client and Printer in
1903 Figure 1. In this case, Printer 1 sends Event Notifications that are shown as Event Notifications (A)
1904 of Figure 2,.
- 1905 2. When the Printer 2 (in the output-device) generates Events, there are two possible system
1906 configurations:
 - 1907 a) Printer 1 forwards the client-supplied Subscription Creation Operations to the downstream
1908 Printer 2 and lets Printer 2 send the Event Notifications directly to the Notification Recipients
1909 supplied by the Client (Event Notifications(C) in the diagram).

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

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

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

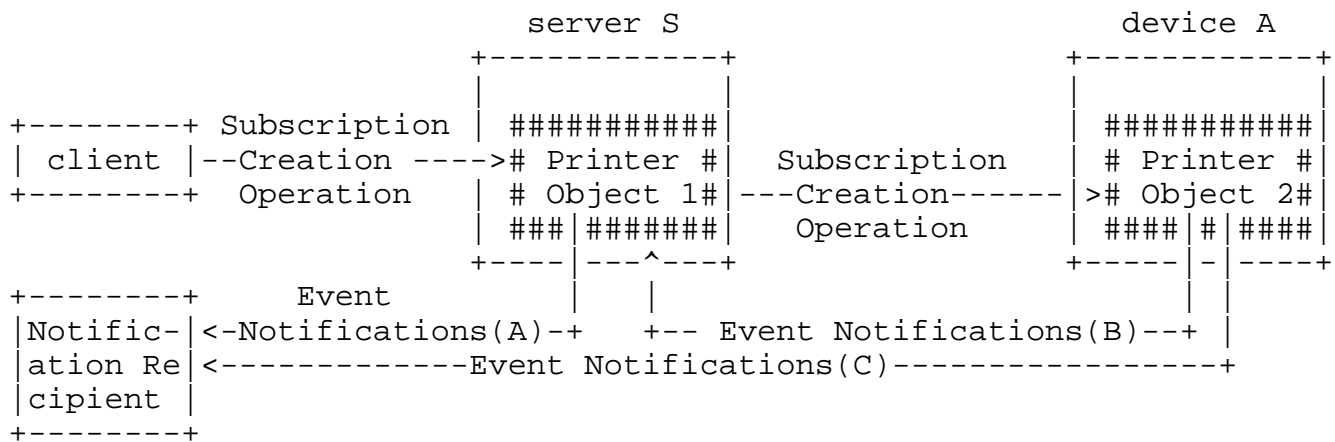


Figure 2 – Model for Notification with Cascading Printers

B. Appendix - Distributed Model for Notification

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

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

1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972

PDA, desktop, or server

```
+-----+
| client |---IPP Subscription---
+-----+      Creation operation
```

```
+-----+
|Notification|
|Recipient   |
+-----+
```

```
*****
*
* Printer (including
* the distributed
* Notification Service)
*
* output device or server
* +-----+
* | ##### |
* | # partial # |
* |># Printer # |
* | # Object # |
* | ##### |##### |
* +-----+-----+
* | Subscriptions
* | OR Event
* | Notifications
* +-----v-----+
* | Notification |
* | Service      |
* +-----+
*
*****
```

*** = Implementation configuration opaque boundary

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

1973

1974

C. Appendix - Extended Notification Recipient

1975
1976
1977
1978

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.

1979

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

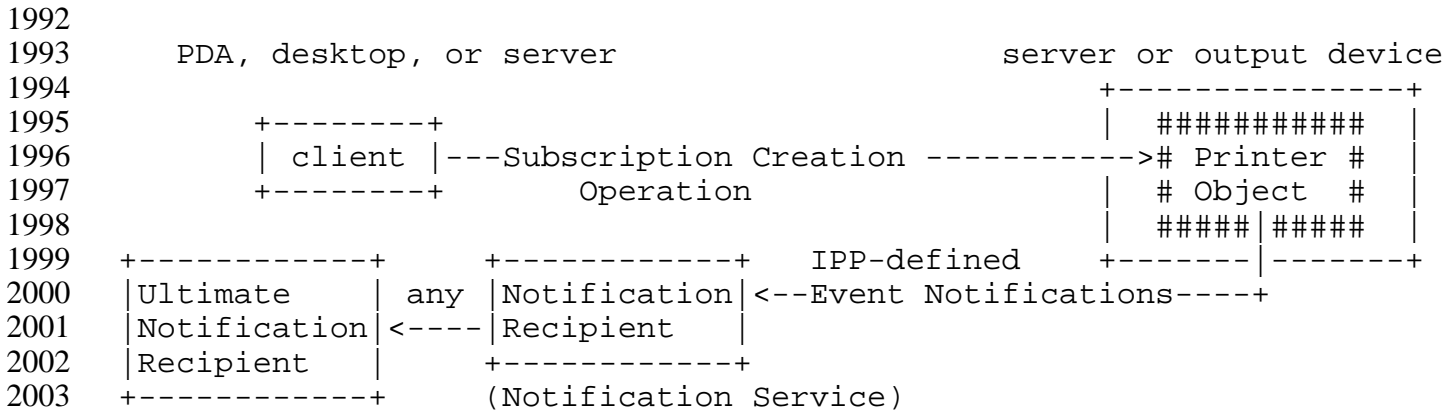
1980
1981
1982
1983
1984
1985

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.

1986
1987
1988
1989

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.

1990 Examples of this extended Notification Recipient are paging, immediate messaging services, general
1991 notification services, and NOS vendors' infrastructure. Figure 4 shows this approach.



2004 **Figure 4 – Use of an Extended Notification Recipient transparent to the Printer**

2005 **D. Appendix - Details about Conformance Terminology**

2006 The following paragraph provide more details about conformance terminology.

2007 **REQUIRED** - an adjective used to indicate that a conforming IPP Printer implementation **MUST**
2008 support the indicated operation, object, attribute, attribute value, status code, or out-of-band value in
2009 requests and responses. See [ipp-mod] "Appendix A - Terminology for a definition of "support".
2010 *Since support of this entire Notification specification is OPTIONAL for conformance to IPP/1.0*
2011 *or IPP/1.1, the use of the term REQUIRED in this document means "REQUIRED if this*
2012 *OPTIONAL Notification specification is implemented".*

2013 **RECOMMENDED** - an adjective used to indicate that a conforming IPP Printer implementation is
2014 recommended to support the indicated operation, object, attribute, attribute value, status code, or
2015 out-of-band value in requests and responses. *Since support of this entire Notification specification*
2016 *is OPTIONAL for conformance to IPP/1.0 or IPP/1.1, the use of the term RECOMMENDED in*
2017 *this document means "RECOMMENDED if this OPTIONAL Notification specification is*
2018 *implemented".*

2019 **OPTIONAL** - an adjective used to indicate that a conforming IPP Printer implementation **MAY**, but is
2020 **NOT REQUIRED** to, support the indicated operation, object, attribute, attribute value, status code,
2021 or out-of-band value in requests and responses.

2022 **E. Appendix - Object Model for Notification**

2023 This section describes the Notification object model that adds a Subscription Object which together with
2024 the Job and Printer object provide the complete Notification semantics.

The object relationships can be seen pictorially as:

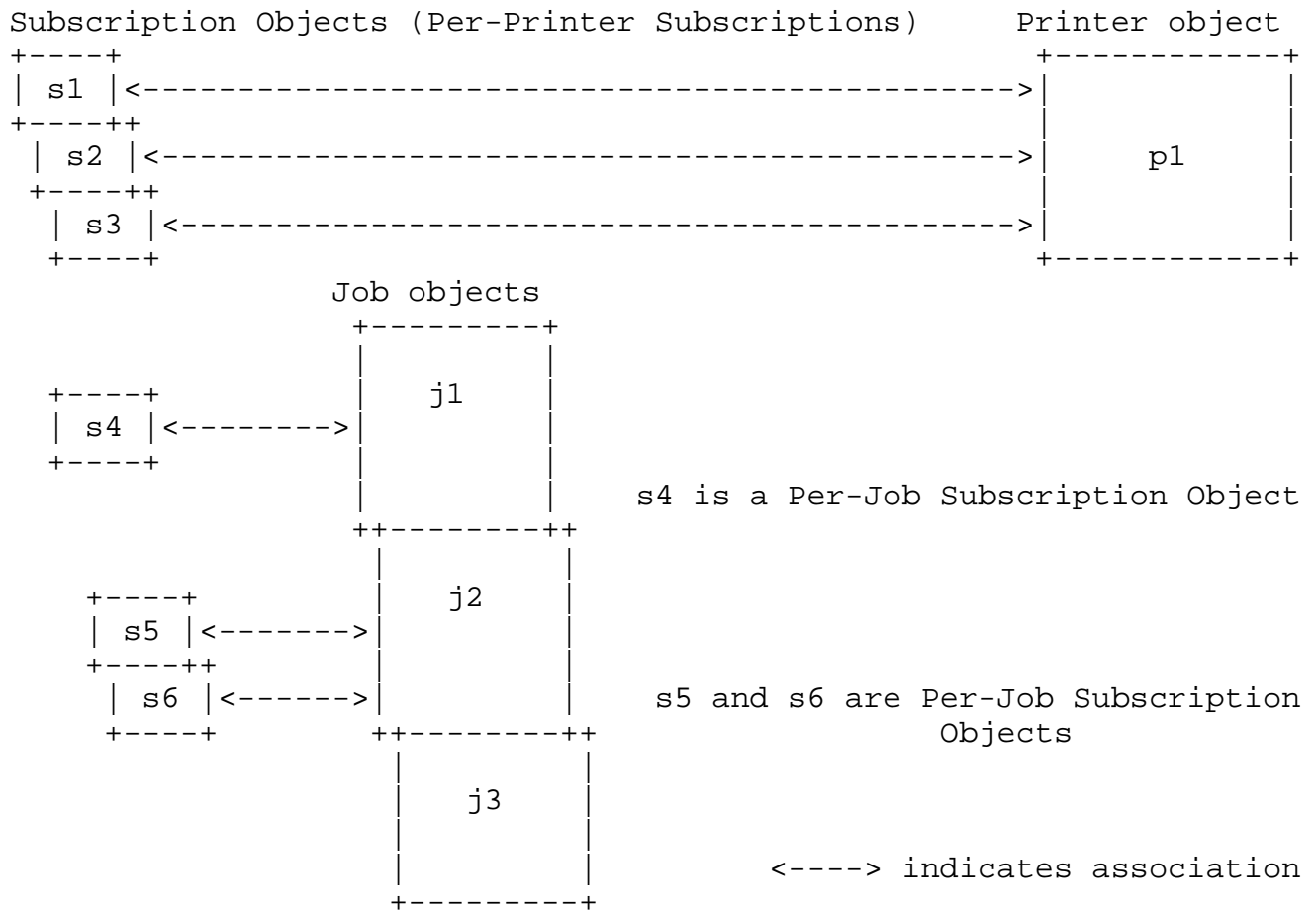


Figure 5 – Object Model for Notification

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

E.1 Appendix - Object relationships

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

E.2 Printer Object and Per-Printer Subscription Objects

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

- 2067 2. Each Per-Printer Subscription Object (s1, s2, and s3) is contained in (or is associated with) exactly
2068 one Printer object (p1).

2069 **E.3 Job Object and Per-Job Subscription Objects**

- 2070 1. A Job object (j1, j2, j3) is associated with zero or more Per-Job Subscription Objects (s4-s6). Job j1
2071 is associated with Per-Job Subscription Object s4, Job j2 is associated with Per-Job Subscription
2072 Objects s5 and s6, and Job j3 is not associated with any Per-Job Subscription Object.
- 2073 2. Each Per-Job Subscription Object is associated with exactly one Job object.

2074 **F. Appendix - Per-Job versus Per-Printer Subscription Objects**

2075 Per-Job and Per-Printer Subscription Objects are quite similar. Either type of Subscription Object can
2076 subscribe to Job Events, Printer Events, or both. Both types of Subscription Objects can be queried using
2077 the Get-Subscriptions and Get-Subscription-Attributes operations and canceled using the Cancel-
2078 Subscription operation. Both types of Subscription Objects create Subscription Objects which have the
2079 same Subscription Object attributes defined. However, there are some semantic differences between Per-
2080 Job Subscription Objects and Per-Printer Subscription Objects. A Per-Job Subscription Object is
2081 established by the client when submitting a job and after creating the job using the Create-Job-
2082 Subscriptions operation by specifying the “job-id” of the Job with the “notify-job-id” attribute. A Per-
2083 Printer Subscription Object is established between a client and a Printer using the Create-Printer-
2084 Subscriptions operation. Some specific differences are:

- 2085 1. A client usually creates one or more Per-Job Subscription Objects as part of the Job Creation operations
2086 (Create-Job, Print-Job, and Print-URI), rather than using the OPTIONAL Create-Job-Subscriptions
2087 operation, especially since Printer implementations NEED NOT support the Create-Job-Subscriptions
2088 operation, since it is OPTIONAL.
- 2089 2. For Per-Job Subscription Objects, the Subscription Object is only valid while the job is “not-complete”
2090 (see sections 5.4.3) while for the Per-Printer Subscription Objects, the Subscription Object is valid until
2091 the time (in seconds) that the Printer returned in the “notify-lease-expiration-time” operation attribute.
- 2092 3. Job Events in a Per-Job Subscription Object apply only to “one job” (the Job created by the Job
2093 Creation operation or references by the Create-Job-Subscriptions operation) while Job Events in a Per-
2094 Printer Subscription Object apply to ALL jobs contained in the IPP Printer.

2095 **G. Appendix: Change History (to be removed for Internet-Draft)**

2096 This section summarizes the changes to the document. Each sub-section is in *reverse* chronological order.
2097 Adding or removing ISSUES that don't change the document are not listed here.

2098 **G.1 Changes to the June 30, 2000 version to create the July 13, 2000 version**

2099 The following changes were made to the June 30, 2000 version to create the July 13, 2000 version based on
2100 the agreements reached at the July IPP WG meeting:

- 2101 1. Deleted the “notify-max-job-subscriptions” and “notify-max-printer-subscriptions” Printer Description
2102 attributes, since the maximum cannot be guaranteed.
- 2103 2. Added the “notify-time-interval (integer(0:MAX)) Subscription Template attribute to give Subscribing
2104 Client control over moderation of ‘job-progress’ Event Notifications that MUST be implemented if and
2105 only if the ‘job-progress’ event is implemented. There are no default or supported Printer attributes.
- 2106 3. Removed the idea that a Delivery Method MAY allow the Printer to moderate certain high frequency
2107 events.
- 2108 4. Clarified that the Printer MUST treat the address part of the “notify-recipient-uri” attribute value as
2109 opaque.
- 2110 5. Added the REQUIRED ‘printer-stopped’ event and the OPTIONAL ‘job-stopped’ event.
- 2111 6. Deleted the ‘job-purged’ event.
- 2112 7. Deleted the “notify-persistence” Subscription Template attribute.
- 2113 8. Clarified the concept of Compound Event Notifications used by both ‘mailto’ and ‘indp’.
- 2114 9. Clarified that a Printer MUST cancel a Subscription if it gets hard errors when sending that will never
2115 change.
- 2116 10. Clarified Figure 3 – Opaque Use of a Notification Service Transparent to the Client to indicate that the
2117 Printer includes the Notification Service.

2118 **G.2 Changes to the May 10, 2000 version to create the June 30, 2000 version**

2119 The following changes were made to the May 10, 2000 version to create the June 30, 2000 version based on
2120 the agreements reached at the May IPP WG meetings and subsequent teleconferences:

- 2121 1. Editorially reorganized and revised the document so that information is stated only once. Moved
2122 supplementary material to appendices.
- 2123 2. Cleaned up the terminology so that it is used consistently throughout the document; capitalized such
2124 terms. Simplified the descriptions of each term.
- 2125 3. Recast the Subscription attributes to be Subscription Template and Subscription Description attributes
2126 following the IPP/1.1 model for Jobs. Therefore, a few attribute names were changed to make them
2127 consistent.
- 2128 4. Reworked the operation descriptions to align with the style in [ipp-mod].
- 2129 5. Made the validation and processing of Subscription Template attributes be the same for Job Creation
2130 Operations, Create-Job-Subscriptions, and Create-Printer-Subscriptions operations (and defined in one
2131 place) and as similar to validation of jobs as possible (though there are some differences since one
2132 request can generate multiple Subscription objects).

- 2133 6. Clarified the error handling for all operations.
- 2134 7. Removed the “notify-text-format” and “notify-additional-formats” Subscription Template attributes and
2135 added the “notify-job-id” Subscription Description attribute.
- 2136 8. The client can supply one or more Subscription Template Attribute Groups in all Subscription Creation
2137 requests and the printer returns Subscription Object Attributes groups for each Subscription object
2138 created. Consequently, an “s” was added to Create-Job-Subscriptions and Create-Printer-Subscriptions
2139 operations.
- 2140 9. Reorganized the Events, so that some of the Events represent a group of events and the rest are sub-
2141 events. This reduces the number of Subscribed Events that a Printer needs to support in one
2142 Subscription from 5 to 2. It also means that the event that is delivered is one of the Subscribed events,
2143 not necessarily the trigger event, so “notify-trigger-event” was renamed to “notify-subscribed-event” in
2144 the Event Notification.
- 2145 10. Added the ‘printer-full’ and ‘printer-not-almost-idle’ Events to go along with the ‘printer-no-longer-
2146 full’ and ‘printer-almost-idle’ Events. Renamed the ‘printer-queue-changed’ Event to ‘printer-queue-
2147 order-changed’.
- 2148 11. Clarified what MUST be in a Delivery Method Document.
- 2149 12. Removed “persistent-jobs-supported” Printer Description attribute, since it has nothing to do with
2150 Notifications and is not needed to describe Subscription object persistence.
- 2151 13. Changed notify-max-printer-subscriptions-supported (integer(0:MAX)) and notify-max-job-
2152 subscriptions-supported (integer(0:MAX)) so that MAX means no limit and 0 means no subscriptions
2153 are (currently) allowed, so as to give a way to turn off accepting new subscriptions.

2154 **G.3 Changes to the March 8, 2000 version to create the May 10, 2000 version**

2155 The following changes were made to the March 8, 2000 version to create the May 10, 2000 version based
2156 on the agreements reached at the April IPP WG meetings and subsequent teleconferences:

- 2157 1. Change “notify-format” to “notify-text-format” and made it apply only to the format of the “notify-
2158 text” (formerly called “human-readable-report”) and Human Consumable form. A new attribute “notify-
2159 additional-formats” specifies the formats for the Machine Consumable contents of Delivery Methods
2160 that support multiple formats.
- 2161 2. Change the “job-notify” collection attribute in Job Creation operations to be multiple “notify-xxx”
2162 attributes. This change eliminates the need for collection values. It also means that a Job Creation
2163 operation can create only one Subscription Object.
- 2164 3. Change the Machine Consumable form to be transport independent.
- 2165 4. Reduce the set of REQUIRED attributes in the Machine Consumable form and add the OPTIONAL
2166 “notify-attributes” attribute that allows a client to request additional attributes.

2167 5. Specify the information that SHOULD be in the Human Consumable form

2168 **G.4 Changes to the March 6, 2000 version to create the March 8, 2000 version**

2169 The following changes were made to the March 6, 2000 version to create the March 8, 2000 version based
2170 on the agreements reached on the mailing list:

- 2171 1. Changed the name of the SNMP Delivery Method from ‘snmp’ to ‘snmpnotify’, since the Notification
2172 Recipient isn’t an SNMP agent.
- 2173 2. Clarified that an implementation with only a single value for persistent-jobs-supported (boolean) or
2174 persistent-subscriptions-supported (boolean) MAY make it settable to the single value or make it not-
2175 settable.

2176 **G.5 Changes to the February 2, 2000 version to create the March 6, 2000 version**

2177 The following changes were made to the February 2, 2000 version to create the March 6, 2000 version
2178 based on the agreements reached on the mailing list, at the February IPP WG meetings, and reflected in the
2179 minutes:

- 2180 1. Clarified that this extension is intended as an extension to IPP/1.0, IPP/1.1, and future versions.
- 2181 2. Allocated the operation-id 0x0016 to 0x001B values for the Notification operations defined in the
2182 document.
- 2183 3. Pre-pended the word “subscription-” on the front of the “request-id” Subscription Object attribute to
2184 distinguish it from the “request-id” parameter that is sent in every request and response.
- 2185 4. Added the term “settable” for describing attributes that are not READ-ONLY.
- 2186 5. Added the term “Subscription Creation Operation” to stand for any operation that can create a
2187 Subscription Object: Job Creation operations (Create-Job, Print-Job, and Print-URI), Create-Job-
2188 Subscriptions, and Create-Printer-Subscriptions.
- 2189 6. Changed the “subscriber-user-name” (name(MAX)) Subscription Object attribute from OPTIONAL to
2190 REQUIRED.
- 2191 7. Changed the name and semantics of “notify-printer-up-time(integer(1:MAX)) to notify-server-up-time
2192 so that it can be either the Printer’s uptime or a Notification Delivery Service uptime.
- 2193 8. Added the ‘ipp:’, ‘indp:’, ‘mailto:’, and ‘snmp:’ notification delivery schemes to the definition of the
2194 “notify-recipients” to indicate possible schemes.
- 2195 9. Changed the name and semantics of “notify-text-format” (mimeMediaType) to “notify-format” so that it
2196 can be used to specify either Human Consumable or Machine Consumable formats where the
2197 implementation supports both. Clarified that this attribute controls whatever variable Notification
2198 Content that the implementation supports, which may be an attachment to the fixed content format or
2199 the contents of the “human-readable-report” (text(MAX)) attribute. Clarified that an implementation

- 2200 NEED NOT support all of its supported Notification Content formats with all of its supported Delivery
2201 Methods.
- 2202 10. Added 'text/xml', 'application/ipp', 'application/postscript', and 'image/tiff' and additional example
2203 MIME media types for "notify-format" (mimeMediaType).
- 2204 11. Clarified that the recommend way for a client to determine whether or not a Printer supports Per-Job
2205 Subscriptions is to query the Printer's "notify-max-job-subscriptions-supported" attribute, since Create-
2206 Job-Subscriptions is an OPTIONAL operation.
- 2207 12. Clarified that the recommend way for a client to determine whether or not a Printer supports Per-Printer
2208 Subscriptions is to query the Printer's "operations-supported" attribute to see if the Create-Printer-
2209 Subscriptions operations is supported, since this is the usual way to determine a Printer's capabilities.
- 2210 13. Clarified that if "persistent-jobs-supported" (boolean) and "persistent-subscriptions-supported"
2211 (boolean) are settable, then setting them must affect whether or not jobs and subscriptions are persistent.
- 2212 14. Allowed Delivery Methods to send operations with or without a response, depending on the definition
2213 of the Delivery Method.
- 2214 15. Indicated that a deliver method definition is free to REQUIRE that the client supply the "notify-user-
2215 data" attribute.
- 2216 16. Required that the Printer support the "job-uri" operation attribute as a target, in addition to "printer-uri"
2217 & "job-id", i.e., keep consistent with all Job operations.
- 2218 17. Changed the 'none' out-of-band value to be a reference to the collection document [ipp-coll], since the
2219 use for it in this document is with the 'collection' attribute syntax.
- 2220 18. Clarified that a conforming implementation MUST support the 'collection' attribute syntax, since that is
2221 required in Job Creation operations.
- 2222 19. Allocated the values to the new status codes defined in this document.
- 2223 20. Allocated the [ipp-pro] subscription-attributes-tag and notification-attributes-tag delimiter tags to
2224 delimit Subscription attributes and Notification Content attributes in requests and responses.
- 2225 21. Changed the 'server-error-too-many-subscriptions' and 'server-error-too-many-events' to be client
2226 errors, i.e., 'client-error-too-many-subscriptions' and 'client-error-too-many-events', since other errors
2227 of this type are client errors.

2228 **G.6 Changes to the October 14, 1999 version to create the February 2, 2000 version**

2229 The following changes were made to the October 14, 1999 version to create the February 2, 2000 version
2230 based on the agreements reached at the October and December IPP WG meetings and reflected in the
2231 minutes:

- 2232 1. Added a Java Listener as an example of a Notification Recipient.
- 2233 2. Clarified the object relationships.
- 2234 3. Clarified how job Events differ for Per-Job versus Per-Printer Subscriptions.
- 2235 4. Added the ability for the Machine Consumable form to contain a Human Readable “human-readable-report” (text) attribute so that both forms could be sent in the same Notification.
2236
- 2237 5. Clarified that the ‘none’ value for notify-text-format (mimeType) has to be out-of-band, not the
2238 text string ‘none’ as a mimeType.
- 2239 6. Clarified that ‘none’ means send the Machine Consumable form without the “human-readable-report”
2240 (text) attribute, if it is defined.
- 2241 7. Clarified that Notification Recipients MUST be able to accept unrecognized attributes.
- 2242 8. Allowed the notification Delivery Method definition to be modeled as (1) a request with an operation
2243 code without a response, (2) a request with a operation code with a response or (3) a response with a
2244 status code.
- 2245 9. Added “notify-text-format” (mimeType) and “human-readable-report” (text(MAX)) to be able to
2246 be sent in a Notification content, if the notification Delivery Method Document permits it.
- 2247 10. Added “job-k-octets” (integer(0:MAX)), “job-impressions” (integer(0:MAX)), and “job-media-sheets”
2248 (integer(0:MAX)) as OPTIONAL for Notification content for use in job-progress Events to show the
2249 target values so that the Notification Recipient can show a thermometer.
- 2250 11. Added a Subscription Attributes Group (and subscription-attributes tag) the Create-Job-Subscriptions
2251 and Create-Printer-Subscriptions requests and responses.
- 2252 12. Added the ‘none’ out-of-band value for use with “notify-text-format” (mimeType) attribute.
- 2253 13. Changed the job progress attributes from using -2 to mean ‘unknown’ as in the PWG Job Monitoring
2254 MIB, to use the ‘unknown’ out-of-band value.

2255

2256 **H. Appendix: Full Copyright Statement**

2257 Copyright (C) The Internet Society (1998,1999,2000). All Rights Reserved

2258 This document and translations of it may be copied and furnished to others, and derivative works that
2259 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
2260 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
2261 this paragraph are included on all such copies and derivative works. However, this document itself may not
2262 be modified in any way, such as by removing the copyright notice or references to the Internet Society or
2263 other Internet organizations, except as needed for the purpose of developing Internet standards in which
2264 case the procedures for copyrights defined in the Internet Standards process must be followed, or as
2265 required to translate it into languages other than English.

2266 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its
2267 successors or assigns.

2268 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET
2269 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,
2270 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE
2271 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
2272 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

2273