

11 **Event notifications for the IPP print protocol [and JMP]**

12 Version 0.04

13 *There are several issues indicated in the document that we should cover at the upcoming*
14 *meeting, as well as review the proposal. See color highlighting.*

15 The appendix has the full specification for the 'collection' attribute syntax, as agreed on
16 our 5/6/98 telecon.

17 [Items in square brackets relate to the PWG JMP MIB trapping and will be removed
18 when this document is made into an IPP Internet-Draft.]

20 Abstract

21 In IPP/1.0, the user can determine what is happening to submitted jobs by using the Get--
22 Attributes and Get-Jobs operations to poll for results. This document describes an
23 OPTIONAL extension to the IPP/1.0 Model document for subscribing for event
24 notifications using IPP, but which are delivered over some other protocol, either by the
25 IPP Printer object or by any notification service that the IPP Printer object
26 implementation may employ. See [req] for the notification requirements.

27 Two methods are provided for subscription for notification events: (1) as part of the job
28 submission and (2) as a separate Subscribe-For-Event-Notifications operation. Both
29 methods allow the requester to specify (1) about which event(s) to be notified, (2) which
30 notification-recipient(s) are to receive the notification, (3) what content type is to be sent
31 in the notification, and (4) which notification transport method is to be used. Both
32 methods allow the requester to subscribe for job event groups, such as 'job-completion',
33 and/or printer events, such as 'printer-errors'.

34 The event notification subscription mechanism uses a new attribute syntax called a
35 'collection'. A 'collection' value is a set of attributes. See the Appendix of this document
36 for the complete specification of the 'collection' attribute syntax.

37

38 1 Introduction 5

39 1.1 Summary of the proposal for IPP Event Notification 6

40 2 Terminology 7

41 2.1 Job Submitting End User..... 7

42 2.2 Job Submitting Application..... 7

43 2.3 Security Domain..... 7

44 2.4 IPP Client 7

45 2.5 Job Recipient..... 7

46 2.6 Job Recipient Proxy 7

47 2.7 Notification Recipient Agent 8

48 2.8 Notification Recipient 8

49 2.9 Notification Events..... 8

50 2.10 Notification Subscription 8

51 2.11 Event Notification Content Attributes..... 8

52 2.12 Immediate Notification 9

53 2.13 Queued Notification 9

54 2.14 Notification with Reliable Delivery 9

55 2.15 Notification with Unreliable Delivery..... 9

56 2.16 Quality of Service..... 9

57 2.17 Human Consumable Notification..... 9

58 2.18 Machine Consumable Notification..... 10

59 2.19 Mixed Notification 10

60 3 Model for Job and Printer Event Notification 11

61 4 Subscription for notification..... 13

62 4.1 Subscription as part of job submission..... 13

63 4.2 Subscription independent of job submission..... 13

64 4.3 Semantics of Subscriptions 13

65 5 New Operation attribute for the create operations 15

66 5.1 job-notify (1setOf collection)..... 15

67 5.1.1 Notification collection value 15

68 notify-event-groups (1setOf type2 keyword) 16

69 Notification Groups..... 16

70	Notification Events.....	18
71	notify-recipients (1setOf uri)	19
72	notify-content-type (mimeType)	20
73	notify-charset (charset)	21
74	notify-natural-language (naturalLanguage)	21
75	notify-additional-attributes (1setOf keyword)	21
76	6 Operations to Subscribe and Un-subscribe for notifications.....	23
77	6.1 Subscribe-For-Event-Notifications Operation	23
78	6.1.1 Subscribe-For-Event-Notifications Request	23
79	6.1.2 Subscribe-For-Event-Notifications Response	24
80	6.2 Un-Subscribe-For-Event-Notifications Operation	25
81	6.2.1 Un-Subscribe-For-Event-Notifications Request	25
82	6.2.2 Un-Subscribe-For-Event-Notifications Response	26
83	7 Job Object Description attributes for Job Notification.....	27
84	7.1 "job-notify" (1setOf collection).....	27
85	8 Printer Object Description attributes for Notification.....	28
86	8.1 Job Notification Support Printer Description attributes.....	29
87	8.2 Printer Notification Support Printer Description attributes.....	29
88	9 Notification Content definitions.....	30
89	9.1 "time-at-event" (integer (0:MAX)).....	30
90	9.2 "event" (keyword)	30
91	9.2.1 Job event notification content	32
92	9.2.2 Printer event notification content	34
93	9.2.2.1 "device-name" (name).....	34
94	9.2.2.2 "which-alert-row" (keyword)	34
95	10 Encoding.....	35
96	11 References	35
97	12 Copyright Notice	36
98	13 Author's Address	36
99	14 Appendix - Specification for the IPP collection attribute syntax.....	38
100	14.1 Problem Statement	38
101	14.2 Summary of the attribute syntax alternative	38
102	14.3 Requirements for and properties of the suggested collection mechanism	38
103	14.4 Examples of collection usage.....	39
104	14.4.1 Example a: "printer-resolution" Job Template attribute	39
105	14.4.1.1 "printer-resolution-default" example.....	40

106	14.4.1.2	"printer-resolution-supported" example and validation of collections..	40
107	14.4.2	Example b: "job-notify" Operation attribute	40
108	14.4.3	Example c: Start page fields supplied by the end-user	41
109	14.4.4	Example d: Postal mailing address	41
110	14.5	Detailed description 'collection' attribute syntax.....	42
111	14.6	Encoding.....	43
112	14.7	Rejected alternatives for a collection mechanism.....	44
113			

114 **1 Introduction**

115 In IPP/1.0, the user can determine what is happening to submitted jobs by using the Get-
116 Attributes and Get-Jobs operations to poll for results. This document describes an
117 OPTIONAL extension to the IPP/1.0 Model document for subscribing for event
118 notifications using IPP, but which are delivered over some other protocol, either by the
119 IPP Printer object or by any notification service that the IPP Printer object
120 implementation may employ. See the IPP Notification Requirements document [req] for
121 further details. See also "General Event Notification Architecture Base [cohen] for
122 terminology and framework.

123 This document contains the definition and use of event notifications (see terminology
124 section) for two main purposes. First, when used to achieve printing over a wide area
125 network, or the Internet, the end-user experience is similar to today's FAX paradigm, so
126 we want to provide notification that the job has completed successfully (or not). This
127 notification may traverse the Internet as an e-mail message or end up on someone's pager.
128 Second, and more widely, when used as a standard LAN print submission protocol (i.e.,
129 LPR replacement), the end-user will have the desire and opportunity for a much more
130 dynamic interaction with the printer and the print job. Here, notification should consist of
131 a local area network messaging scheme that addresses unsolicited events related to the
132 printer, the job's position in the server or printer queue, start of processing, printing
133 progress and job completion, including forms of cancellation. This paper proposes
134 MANDATORY IPP attributes to be used for both purposes, and OPTIONAL attributes
135 and values that are appropriate only for one or the other.

136 [The notification events and content are also intended to apply to the PWG Job
137 Monitoring MIB (JMP). See sections 5.1.2.2 and 6.]

138 **1.1 Summary of the proposal for IPP Event Notification**

139 This paper proposes the following:

- 140 1. One OPTIONAL "job-notify" Operation attribute for use with the Print-Job, Print-
 141 URI, and Create-Job operation. The "job-notify" Operation attribute has an attribute
 142 syntax of '1setOf collection' (see Appendix) so that the client can request different
 143 events for different notification recipients for the same job. Each collection value
 144 SHALL contain the "notify-recipients" and MAY contain any of the following
 145 remaining member attributes with the indicated syntax and support by the IPP object
 146 if it supports the "job-notify" Operation attribute at all:

147 Member attribute name	syntax	in request	support
148 -----	-----	-----	-----
149 "notify-event-groups"	1setOf type2 keyword	MAY	mandatory
150 "notify-recipients"	1setOf uri	SHALL	mandatory
151 "notify-content-type"	mimeMediaType	MAY	mandatory
152 "notify-charset"	charset	MAY	mandatory
153 "notify-natural-language"	naturalLanguage	MAY	optional
154 "notify-additional-attributes"	1setOf keyword	MAY	optional

- 155
 156 2. One "job-notify" Job Description attribute which is populated with the collection
 157 value(s) supplied by the "job-notify" Operation attribute in a create operation.

158 **ISSUE 01: Would a better name be "job-notification-subscription" and the member**
 159 **attributes be named "notification-xxx"?**

- 160 3. Six "job-xxx-supported" Printer object attributes that correspond to these six member
 161 attributes. See the IPP Model for the semantics of xxx-supported Printer attributes.

- 162 4. Two new OPTIONAL Subscribe-For-Event-Notifications and Un-Subscribe-For-
 163 Event-Notifications operations on the Printer object. These operations are intended
 164 for operator/administrators and servers for long term subscription for Printer object
 165 events that are independent of job submission. The servers may be involved with (1)
 166 job submission to IPP Printer objects and/or (2) collecting accounting data using the
 167 event notification mechanism.

168 An IPP Printer SHALL support both of these operations, if it supports either one. If
 169 an IPP Printer supports these operations, it SHALL also support the "job-notify"
 170 attribute in the create operations.

- 171 5. One new "printer-notify" Printer Description attribute which is populated with the
 172 collection value supplied by the "printer-notify" Operation attribute in the Subscribe-
 173 For-Event-Notifications operation. Both attribute use the same collection as the "job-
 174 notify" Operation attribute. The "printer-notify" Printer Description attribute also has
 175 an additional "subscription-id" member attribute which is an integer id for the
 176 subscription for use with the Un-Subscribe-For-Event-Notification operation.

177 **ISSUE 02: Would a better name be "printer-notification-subscription"?**

178 **2 Terminology**

179 It is necessary to define a set of terms in order to be able to clearly express the
180 requirements for notification services in an IPP System. These terms are from the
181 requirements document [req]. Cohen [cohen] has similar terminology, with some
182 differences. **ISSUE 03: Which terminology should we use?**

183 **ISSUE 04: Some of these terms are not used in the specification. Should we delete**
184 **them?**

185 **2.1 Job Submitting End User**

186 A human end user who submits a print job to an IPP Printer. This person may or may not
187 be within the same security domain as the Printer. This person may or may not be
188 geographically near the printer.

189 **2.2 Job Submitting Application**

190 An application (for example a batch application), acting on behalf of an end user, which
191 submits a print job to an IPP Printer. The application may or may not be within the same
192 security domain as the Printer. This application may or may not be geographically near
193 the printer.

194 **2.3 Security Domain**

195 For the purposes of this discussion, the set of network components which can
196 communicate without going through a proxy or firewall. A security domain may be
197 geographically very large, for example - anyplace within IBM.COM.

198 **2.4 IPP Client**

199 The software component on the client system which implements the IPP protocol which
200 can be either a Job Submitting End User or a Job Submitting Application.

201 **2.5 Job Recipient**

202 A human who is the ultimate consumer of the print job. In many cases this will be the
203 same person as the Job Submitting End User, but this need not always be the case. For
204 example, if I use IPP to print a document on a printer in a business partner's office, I am
205 the Job Submitting End User, while the person I intend the document for in my business
206 partner's office is the Job Recipient. Since one of the goals of IPP is to be able to print
207 near the ultimate recipient of the printed output, we would normally expect the Job
208 Recipient to be in the same security domain as, and geographically near the Printer.
209 However, this may not always be the case. For example, I submit a print job across the
210 Internet to a Kinko's print shop. I am both the Submitting end User and the Job
211 Recipient, but I am neither near nor in the same security domain as the Printer.

212 **2.6 Job Recipient Proxy**

213 A person acting on behalf of the Job Recipient. In particular, the Job Recipient Proxy
214 physically picks up the printed document from the Printer, if the Job Recipient cannot

215 perform that function. The Proxy is **by definition** geographically near and in the same
216 security domain as the printer. For example, I submit a print job from home to be printed
217 on a printer at work. I'd like my secretary to pick up the print job and put it on my desk.
218 In this case, I am acting as both Job Submitting End User and Job Recipient. My
219 secretary is acting as a Job Recipient Proxy. An issue that needs to be considered in the
220 notification architecture is the impact of a third party receiving many unwanted
221 notifications.

222 **2.7 Notification Recipient Agent**

223 A program which receives events on behalf of the notification recipient. The agent may
224 take some action on behalf of the recipient, forward the notification to the recipient via
225 some alternative means (for example, page the recipient), or queue the notification for
226 later retrieval by the recipient.

227 **2.8 Notification Recipient**

228 Any of: Job Submitting End User, Job Submitting Application, Job Recipient, or Job
229 Recipient Proxy or Notification Recipient Agent.

230 **2.9 Notification Events**

231 There are Job events and Printer events.

232 A Job event is some change in the Job object, such as: (1) a change in the Job object's
233 "job-state" attribute, (2) the stacking of another sheet, reflected in the incrementing of the
234 job's "job-media-sheets-completed" attribute or (3) some of the changes in the value of
235 the job's "job-state-reasons" attribute. Not all changes in a job's "job-state" attribute are
236 separate events. For example, the event 'job-received' is the transition from the
237 'unknown' state to either the 'pending' or 'pending-held' state. Not all changes in a job's
238 other attributes are events.

239 A Printer event is some change in the Printer object, such as: (1) a change in the
240 Printer object's "printer-state" attribute or (2) a change in the Printer's "printer-
241 state-reasons" attribute. A Printer event corresponds one-to-one with the addition
242 or removal of a row in the Printer MIB alert table, for those implementations that
243 also implement the Printer MIB [prtmib].

244 **2.10 Notification Subscription**

245 End users may "subscribe" for notifications of Job events and/or Printer events when
246 they submit a job. These include any of those described in the preceding section.

247 **2.11 Event Notification Content Attributes**

248 When a Job or Printer event notification is delivered to the notification-recipient, it
249 contains attributes whose values reflect the state of that Job or Printer at the time of the
250 event, respectively. Examples of Job content attributes include:

251 "number-of-intervening jobs"
252 "job-impressions-completed"
253 "job-state-reasons"

254 Examples of Printer object content attributes include:

255 "printer-state-reasons"

256 "device-name"

257 "alert-code"

258 Note: when a Job event is sent, no Printer attributes, except the "printer-uri", are sent.

259 When a Printer event is sent, no Job attributes are sent.

260 **2.12 Immediate Notification**

261 Notifications sent to the notification recipient or the notification recipient's agent in such
262 a way that the notification arrives immediately, within the limits of common addressing,
263 routing, network congestion and quality of service.

264 **2.13 Queued Notification**

265 Notifications which are not necessarily sent immediately, but are queued for delivery by
266 some intermediate network application, or for later retrieval. Email with store and
267 forward is an example of queued notification.

268 **2.14 Notification with Reliable Delivery**

269 Notifications which are delivered by a reliable, sequenced delivery of packets or
270 character stream, with acknowledgment and retry, such that delivery of the notification is
271 guaranteed within some reasonable time limits. For example, if the notification recipient
272 has logged off and gone home for the day, an immediate notification cannot be
273 guaranteed to be delivered, even when sent over a reliable transport, because there is
274 nothing there to catch it. Guaranteed delivery requires both queued notification and a
275 reliable transport. If delivery of the notification requires process to process
276 communications, each session is managed in a reliable manner, assuring fully ordered,
277 end-to-end delivery.

278 **2.15 Notification with Unreliable Delivery**

279 Notifications are delivered via the fundamental transport address and routing framework,
280 but no acknowledgment or retry is required. Process to process communications, if
281 involved, are unconstrained.

282 **2.16 Quality of Service**

283 Some notification delivery methods may allow users to select quality of service
284 parameters. These will depend upon the specific delivery method chosen, and may
285 include parameters such as priority, security, number of retries, and the like.

286 **2.17 Human Consumable Notification**

287 Notifications which are intended to be consumed by human end users **only**. They contain
288 no machine readable encodings of the event. Email would be an example of a Human
289 consumable notification.

290 **2.18 Machine Consumable Notification**

291 Notifications which are intended for consumption by a program **only**, such as an IPP
292 Client. Machine Consumable notifications may not contain human readable information.

293 **2.19 Mixed Notification**

294 A mixed notification may contain both human consumable and machine consumable
295 information. Sending 'multi-part/alternative' MIME media type is mixed notification,
296 since both 'text/plain' and a machine consumable content are sent.

297

298 3 Model for Job and Printer Event Notification

299 The following pictures from the IPP/1.0 Model and Semantics [ipp-model] are enhanced
300 to show subscription for event notification (1) as part of IPP job submission and (2) using
301 the new IPP Subscribe-For-Event-Notifications operations event notifications to
302 (multiple) end-user notification-recipients and a system operator.

303 Legend:

304
305 ##### indicates a Printer object which is
306 either embedded in an output device or is
307 hosted in a server. The Printer object
308 might or might not be capable of queuing/spooling.

309 any indicates any network protocol or direct
310 connect, including IPP

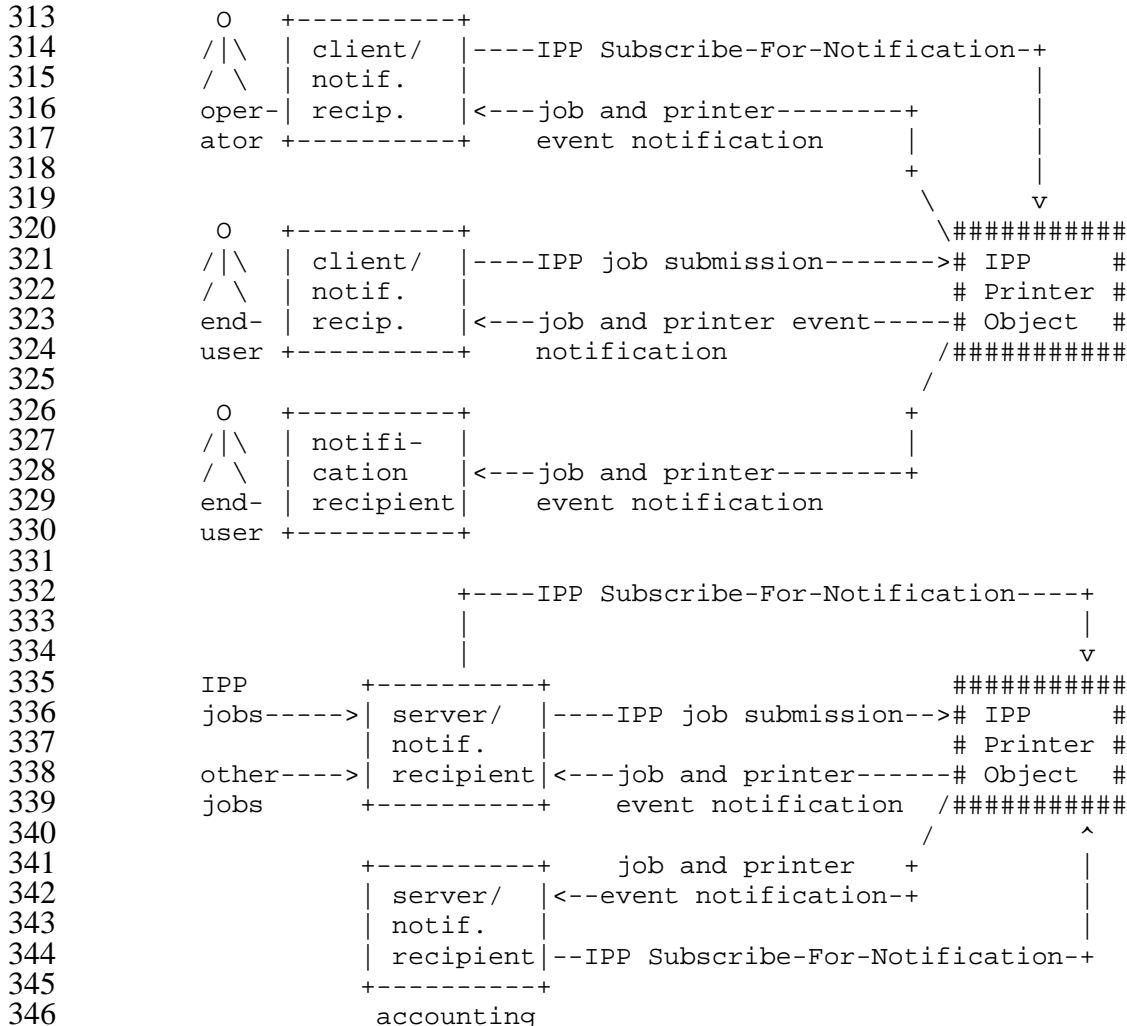


Figure 1 - Model for Job and Printer Notification

347

348 An implementation option is for the IPP Printer object to forward the subscription
 349 requests received in the job submission and with Subscribe-For-Event-Notification
 350 operations to a notification service transparently to the requester. The IPP object then
 351 passes event notifications to this notification service to distribute the event notifications
 352 to the notification recipients.

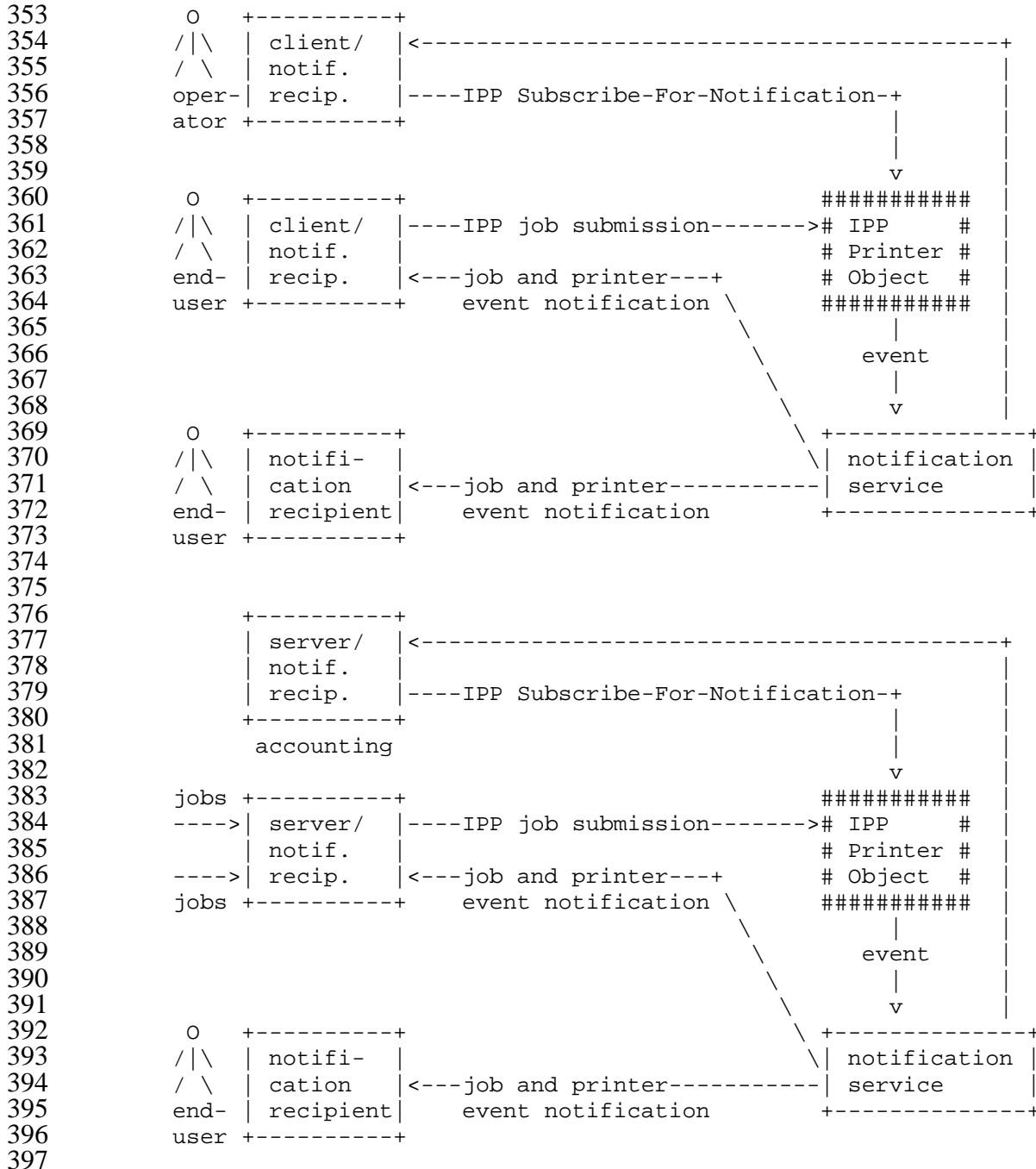


Figure 2 - Model with Notification Service

399 **4 Subscription for notification**

400 **4.1 Subscription as part of job submission**

401 Subscription for notifications is accomplished via IPP for end-user and server-to-device
402 notifications related to the jobs being submitted. This proposal includes specifics for
403 these types of subscriptions. Here the subscription information is submitted with the job
404 and an implementation SHALL store the information with the Job object so that it may be
405 queried with the Get-Job-Attributes operation.

406 As an implementation option, an implementation MAY employ an event notification
407 service to keep the event notification subscription information and to actually deliver the
408 event notifications. In this case, the IPP object passes each event as it occurs to the event
409 notification service for event notification delivery to the notification recipients for which
410 the Printer object had previously forwarded event notification subscriptions.

411 When the IPP Printer removes the job from the system, the subscription is automatically
412 removed with such an implementation. If the IPP Printer object implementation uses a
413 notification server, then the IPP object will have to un-subscribe with that notification
414 server when the job completes.

415 **4.2 Subscription independent of job submission**

416 Subscription by servers that control IPP Printers and by 3rd party accounting or job
417 monitoring applications, which are independent of job submissions, is accomplished by
418 using the Subscribe-For-Event-Notification operation. In these cases, the subscription is
419 in force, until the server or application performs an Un-Scribe-For-Event-Notifications
420 operation.

421 **4.3 Semantics of Subscriptions**

422 This sub-section summarizes the semantics of subscriptions.

423 **ISSUE 06: Ok if the semantics is duplicated here in the spec?**

- 424 1. Job Events are changes in a Job object. Printer Events are changes in the Printer
425 object.
- 426 2. Any subscription can contain either Job Events or Printer Events or both.
- 427 3. Subscriptions can be sent to the IPP Printer object either by being included in a create
428 operation when the job is submitted (called "Job Submission Subscriptions") or by
429 being sent in a separate subscription using the Subscript-For-Event-Notifications
430 operation (called "Printer Subscriptions).
- 431 4. For "Job Submission Subscriptions", the subscription is only valid while the job is
432 "on the scene". The job is on the scene from the time the IPP Job object is created
433 and enters either the 'pending' or 'held' states until the time it is "done" and enters any
434 of the 'completed', 'canceled', or 'aborted' states.
- 435 5. For "Printer Subscriptions", the subscription is valid until it is explicitly un-
436 subscribed with an Un-Subscribe-For-Event-Notifications operation.

- 437 6. Job Events in a "Job Submission Subscription" ONLY apply to "this job" (the Job
438 object created because of the job create operation).
- 439 7. Job Events in a "Printer Subscription" apply to ALL jobs contained in the IPP Printer
440 object.
- 441 8. Subscriptions indicate the delivery method and destination for each set of events
442 being subscribed to. For example, an application may submit a job with a "Job
443 Submission Subscription" indicating that some events should be sent back to it (using
444 some new HTTP based event delivery mechanism using it own address), some events
445 should be sent to a 3rd party accounting/monitoring application (using the same
446 HTTP based event delivery mechanism but with the address of the 3rd party app, not
447 its own address), and finally that some events should be sent to a 3rd party human
448 being (using email and the email address of that human being).
- 449 Implemented another way, the 3rd party accounting/management app could subscribe
450 to all job events using a persistent (until un-subscribed) "Printer Subscription"
451 indicating its own address as the address for delivery of events.
- 452 9. Any subscription (neither a "Job Submission Subscription" nor a "Printer
453 Subscription" allow for subscribing Job Events to a specific (named or otherwise
454 identified) Job.

455 **5 New Operation attribute for the create operations**

456 This section specifies the single "job-notify" Operation attribute that supplies one or more
457 Job Notification Subscriptions as part of a job create operation.

458 **5.1 job-notify (1setOf collection)**

459 The client **OPTIONALLY** supplies this Operation attribute as a *collection* attribute as
460 part of the Validate-Job, Print-Job, Print-URI, and Create-Job operations. The Printer
461 object **OPTIONALLY** supports this Operation attribute as part of the Validate-Job, Print-
462 Job, Print-URI, and Create-Job operations. If the Printer object supports this attribute for
463 any of these create operations, it **MUST** support it for all of these create operations that it
464 supports.

465 The "job-notify" Operation attribute specifies the Job Notification Subscription that starts
466 when the job is created and ends when the job completes (enters the 'completed',
467 'aborted', or 'canceled' job states). The subscription may request Job Events and/or
468 Printer Events. The Job Events **SHALL** apply only to changes in this job (the one being
469 created), while the Printer Events apply to all job. (Note: The Job Events requested with
470 the Subscribe-For-Event-Notifications operation **SHALL** apply to all jobs, just as for
471 Printer Events).

472 **5.1.1 Notification collection value**

473 The value of this attribute is one or more collection values. Each collection value
474 **SHALL** contain a "notify-recipients" member attribute and **MAY** contain any of the
475 remaining following *member* attributes with the indicated syntax:

476	Member attribute name	syntax	in request	support
477	-----	-----	-----	-----
478	"notify-event-groups"	1setOf type2 keyword	MAY	mandatory
479	"notify-recipients"	1setOf uri SHALL		mandatory
480	"notify-content-type"	mimeMediaType	MAY	mandatory
481	"notify-charset"	charset	MAY	mandatory
482	"notify-natural-language"	naturalLanguage	MAY	optional
483	"notify-additional-attributes"	1setOf keyword	MAY	optional

484 The "support" column indicates the support required by the IPP object if it supports the
485 "job-notify" Operation attribute at all.

486 If the client supplies this Operation attribute, but does not supply the "notify-recipients"
487 member attribute as one of the attributes in (each) collection value, the Printer object
488 **SHALL** reject the request and return the 'client-error-bad-request' status code, since the
489 syntax is not correct.

490 If the client supplies this Operation attribute (like the "job-k-octets", "job-impressions",
491 and "job-media-sheets" Operation attributes, see [ipp-model]), but the Printer object does
492 not support the "job-notify" Operation attribute, the Printer object **SHALL** ignore the
493 "job-notify" attribute and copy it to the Unsupported Attribute group with the out-of-band
494 value of 'not-supported'.

495 If the client supplies the "job-notify" Operation attribute and the Printer object supports
496 the "job-notify" Operation attribute, the collection value(s) of the attribute are used to
497 populate the job object's "job-notify" Job Description attribute (see section **Error!**
498 **Reference source not found.**) according to the following conditions:

499 If the values of the member attributes are within the range of the corresponding
500 Printer object's "xxx-supported" attributes (see section **Error! Reference source**
501 **not found.**), the Printer object SHALL use the collection value(s) to populate the
502 job object's "job-notify" Job Description attribute.

503 If some of the member attributes are not supported, the Printer object SHALL
504 copy such member attributes to the Unsupported Attributes response group with
505 the out-of-band value of 'not-supported', copy the remaining (supported) member
506 attributes to the job object's "job-notify" Job Description attribute, accept the
507 request, and return the 'successful-ok-ignored-or-substituted-attributes' status
508 code.

509 If some of the member attribute values are outside the range of the corresponding
510 Printer object's "xxx-supported" attributes (see section **Error! Reference source**
511 **not found.**), the Printer object SHALL copy such member attributes and their
512 values to the Unsupported Attributes response group, substitute or ignore the
513 supplied values, copy the remaining (supported) member attribute values to the
514 job object's "job-notify" Job Description attribute, accept the request, and return
515 the 'successful-ok-ignored-or-substituted-attributes' status code.

516 The following attributes are defined for use in one or more collection values of the "job-
517 notify" Operation attribute in the create operation:

518 **5.1.2 notify-event-groups (1setOf type2 keyword)**

519 The client OPTIONALLY supplies this attribute as a member of the "job-notify"
520 Operation attribute. The Printer object SHALL support this attribute if it supports the
521 "job-notify" Operation attribute. This attribute specifies one or more Job event groups
522 and/or Printer event groups for which the IPP client desires some sort of notification to be
523 sent to one or more notification recipients that the client supplies in the same "job-notify"
524 collection value in the create request for this job.

525 Each event is assigned a keyword value (see section 5.1.2.2). Each of the events is
526 assigned to one or more of the standard event groups. Each standard group is also
527 assigned a keyword (see section 5.1.2.1), in order to simplify (1) client subscription for
528 the events supplied by the client and (2) event filtering by the notification mechanism.

529 **ISSUE 07: Should a requester be able to supply either event group names and/or specific**
530 **event keywords, or is it ok to require only event group names?**

531 **5.1.2.1 Notification Groups**

532 This section defines the event groups that a client may subscribe for in the create
533 operation. These event group keywords (not the actual event keywords themselves) are
534 passed as attribute values in the "notify-event-groups" Operation attribute in the create

535 request. There are Job event groups and Printer event groups. An IPP object SHALL
536 support all event groups. Support of all of the events in a group is not required.

537 **ISSUE 08: Ok if all groups are required for conformance?**

538 Standard event group values are:

539 'none': no notifications of any events. This value is useful to prevent notifications
540 when the client has default notification attributes configured.

541 'all-job-events': any of the supported Job Event notification events occur.

542 **ISSUE 09: Ok if I split 'all' into two, now that we have both kinds?**

543 'job-delivery': any of the following events which, in general, pertain to the progress
544 of delivering the job to the Printer:

545 'job-received', 'job-started-processing'

546 'job-progress': any of the following events which, in general, pertain to the progress
547 of pending or actually interpreting, marking, finishing or otherwise processing the
548 job by the Printer object:

549 'job-held', 'job-released', 'sheet-completed', 'collated-copy-completed'

550 'job-completion': any of the following events which, in general, pertain to ways that a
551 job can end:

552 'job-completed', 'job-aborted', 'job-canceled'

553 'all-printer-events': any of the supported Printer Events occurs.

554 'printer-reports': any Printer object or device event that are informational, as opposed
555 to warnings or errors. Printer MIB events that fall in this group included the
556 alertRemovalOfBinaryChangeEntry(1801) alert that indicates that a binary
557 change event entry row has been removed from the Alert Table and any event
558 with the prtAlertSeverityLevel value set to noInterventionRequired(7) [draft-
559 prtmib].

560 'printer-warnings': any Printer object or device event that are warnings, i.e., non-
561 critical alert where the Printer object's "printer-state" attribute remains in the
562 'processing' state and the device(s) continue to operate. However, if there is not
563 human intervention soon, the device will stop.

564 Examples include: paper-low and toner-low. Warning events may be either
565 binary or unary [see draft-prt-mib]. A binary event is one in which a second event
566 terminates the warning. Examples include: paper low and toner low. A unary
567 event is one in which there is not a second event that terminates the warning.

568 **ISSUE 10: What if a Printer object controls several devices and one of them stops. The**
569 **"printer-state" remains in 'processing', but it should be a Printer error, since some**
570 **device stopped.**

571 'printer-errors': any Printer object or device event that is an errors, i.e., critical alert
572 where the Printer object's "printer-state" attribute changes to 'stopped' or (at least
573 one of) the devices stop (even though other devices that the Printer object
574 controls, continue to operate).

575 Examples include: jammed(8) and markerTonerEmpty(1101).

576 Implementers MAY add additional events to a group. Therefore, notification recipients
577 SHOULD check the event that is sent in the notification content (see section 6) to make

578 sure that it is an event that is wanted. Implementors SHOULD NOT add new groups, lest
579 interoperability will be lessened.

580 In a create request, if the client supplies 'none' along with any other combination of
581 values, it is the same as if only that other set of values had been supplied (i.e., the 'none'
582 value has no affect). If the client supplies 'all' along with any other combination of
583 values, it is the same as if only 'all' had been supplied (i.e., the 'all' value subsumes all
584 other values).

585 Note: the group 'job-progress' is intended for those who wish to receive more frequent,
586 "real-time" progress notifications on a page and copy boundary basis. This is why 'job-
587 started-printing' is in the 'delivery' group, rather than the 'progress' group, for example.
588 An application which was interested in less granular milestones of print job progress
589 would likely subscribe for 'job-completion' and 'printer-errors' event groups (only).

590 5.1.2.2 Notification Events

591 This section defines the notification events. Each event is a member of one or more
592 event groups. When an event occurs, the event keyword, not the event group, is included
593 in the notification content (see section 6).

594 The standard event values are:

- 595 'job-received': when the Printer object accepts the job (i.e., when the job is created
596 entering the 'pending' or 'pending-held' [JMP 'pendingHeld' states] [JMP: issued
597 by the agent when the agent creates a row in the MIB for that job.]
- 598 'job-started-processing': the Printer starts processing the Job (i.e., when the job leaves
599 the 'pending' state and enters the 'processing' state).
- 600 'sheet-completed': when each sheet in the job is completed (i.e., stacked in the output
601 bin).
- 602 'collated-copy-completed': when each document copy in the job is completed (i.e.,
603 last sheet of a collated copy is stacked in an output bin)
- 604 'job-held': when the job enters the 'pending-held' (JMP pendingHeld) state (using
605 some protocol operation not defined in IPP/1.0, but perhaps in another protocol or
606 added as an extension), or the system or device holds the job because of some
607 requirement that cannot be met and other jobs could be processed, if there are any.
- 608 'job-released': when the job leaves the 'pending-held' (JMP pendingHeld) state
609 entering the 'pending' or 'processing' states due to the user, operator, or system
610 releasing the held job (using some protocol operation not defined in IPP/1.0, but
611 perhaps in another protocol or added as an extension).
- 612 'job-warning': when the job encounters a warning. See the definition of the 'job-
613 warnings' event group.
- 614 'job-error': when the job encounters a problem (i.e., when the job leaves the
615 'processing' state and enters the 'processing-stopped' state)
- 616 'job-completed': when the job completes processing (with or without errors or
617 warnings) and enters the 'completed' state.
- 618 'job-aborted': when the job was aborted by the system while in the 'processing' or
619 'processing-stopped' state, due to some encountered problem that cannot be
620 remedied by human intervention.

621 `job-canceled`: when the job was canceled by the user or operator using the Cancel-
622 Job operation while the job was in any state .
623 `printer-report`: when the Printer issues a non-warning and non-error.
624 `printer-warning`: when the Printer issues a non-critical event and continues in the
625 `processing` state.
626 `printer-error`: when the Printer issues a critical event and enters the `stopped` state.
627

628 **5.1.3 notify-recipients (1setOf uri)**

629 The client OPTIONALLY supplies this attribute as a member of the "job-notify"
630 Operation attribute. The Printer object SHALL support this attribute if it supports the
631 "job-notify" Operation attribute and SHALL support the `mailto` scheme at least .

632 **ISSUE 11: Is it too hard to require an embedded device to include sending e-mail?**

633 This attribute describes both where (the address) and how (the mechanism for delivery)
634 events are to be delivered. The Printer object SHALL use this attribute as the set of
635 addresses and methods for sending notifications when one of the events occurs that the
636 client supplied in the "notify-event-groups" member attribute in the same "job-notify"
637 collection value in the create request for this job. The Printer object MAY achieve the
638 subscription and event notification delivery either (1) itself or (2) by using some
639 (unspecified) notification service that supports the requested mechanism of notifying the
640 notification recipients. Either implementation choice SHALL be transparent to clients
641 and notification-recipients.

642 Standard uriScheme values are:

643 `mailto`: a text message via email to the specified email address
644 `http`: an HTML formatted message via an HTTP POST method to the specified URI
645 `ftp`: a text message via an FTP `append` command to the specified remote file.
646

647 The following values are not yet standardized or registered. Some of them represent
648 work in progress. They will be registered following the procedures [url-reg]. See
649 also [cohen] for HTTP URL schemes for notification.

650 **ISSUE 12: Which schemes do we want to progress?**

651
652 `page`: a pager phone number to call as specified by the /phone-number parameter in
653 the URL.

654 `ipp-tcp-ip-socket`: an IPP notification via a TCP/IP socket that is opened by the
655 Printer object on the IP address specified in the URI (using IP address dot
656 notation) using the port on that host specified using the /port=nnn keyword. For
657 example:

658 ipp-tcp-ip-socket:13.240.120.138/port=6000
659 would cause the Printer object to open the TCP/IP port 6000 at IP address
660 13.240.120.138.
661

662 **ISSUE 13: Ok that I removed this note, since the printer-uri is being returned in**
663 **all event notifications?**

664 `snmpv1`: a notification as an SNMPv1 trap to the host specified as the address in the
665 URI.
666 `snmpv2`: a notification as an SNMPv2 inform to the host specified as the address in
667 the URI.
668 `snmpv3`: a notification as an SNMPv3 inform to the host specified as the address in
669 the URI.
670 `sense`: a notification as a SENSE UDP data gram that is opened by the Printer object
671 on the IP address specified in the URI (using IP address dot notation) using the
672 port on that host specified using the /port=nnn keyword. See the `ipp-tcp-ip-
673 socket` example.
674

675 The Printer object SHALL validate that the schemes supplied in the "notify-recipients" is
676 supported by comparing with the Printer object's "notify-schemes-supported".

677 **5.1.4 notify-content-type (mimeMediaType)**

678 The client OPTIONALLY supplies this attribute as a member of the "job-notify"
679 Operation attribute. The Printer object SHALL support this attribute if it supports the
680 "job-notify" Operation attribute and SHALL support the `multi-part/alternative`,
681 `application/ipp`, and the `text/plain` values for all event groups.

682 **ISSUE 14: Ok to require supporting all three values? Ok for all event groups?**

683 This attribute specifies the type of content that is sent in the notification. Thus the client
684 can control whether the event notification content is human readable, machine readable,
685 or both.

686 If the MIME media type registration permits a charset parameter, than such a
687 specification SHALL be used (instead of the "notify-charset" member attribute) in order
688 to indicate the charset to be used in the notification content.

689 Standard values are:

690 `multi-part/alternative` - contains both human consumable notification content
691 using the `text/plain` MIME media type and machine consumable
692 notification content using the `application/ipp` MIME media type with the
693 Get-Job-Attributes response encoding of the attributes listed in Table 3 or
694 the Get-Printer-Attributes response encoding of the attributed listed in
695 Table 4. This value SHALL be supported and is the default, if the client
696 does not supply the "notify-content-type" member attribute.

697 **ISSUE 15: Should we make this attribute 1setOf so that the additional values
698 could specify which alternatives are to be used with `multi-part/alternative`?**

699
700 `application/ipp` - the machine consumable notification content using the
701 `application/ipp` MIME media type [ipp-model] with the Get-Job-
702 Attributes response encoding of the attributes listed in Table 3 or the Get-
703 Printer-Attributes response encoding of the attributed listed in Table 4.
704

705 `text/plain` - the human consumable notification content. If the charset is other
706 than US-ASCII, the /charset parameter SHALL be included in the value of

707 this attribute and in the event notification content. RFC 2046 indicates
708 that the absence of the charset parameter SHALL mean US-ASCII rather
709 than simply unspecified [RFC2046]. Examples:
710 `text/plain`: A plain text document in US-ASCII [US-ASCII]
711 `text/plain; charset=US-ASCII`: A plain text document in US-ASCII.
712 `text/plain; charset=ISO-8859-1`: A plain text document in ISO 8859-
713 1 (Latin 1) [ISO8859-1].
714 `text/plain; charset=utf-8`: A plain text document in ISO 10646
715 represented as UTF-8 [RFC-2044]
716 `text/plain, charset=iso-10646-ucs-2`: A plain text document in ISO
717 10646 represented in two octets (UCS-2) [ISO10646-1]
718

719 **5.1.5 notify-charset (charset)**

720 The client OPTIONALLY supplies this attribute as a member of the "job-notify"
721 Operation attribute. The Printer object SHALL support this attribute if it supports the
722 "job-notify" Operation attribute.

723 This attribute specifies the charset to be used in the human readable part of the
724 notification content that is sent to the notification recipients that the client supplied in this
725 same collection value. This attribute SHALL NOT be used when the "notify-content-
726 type" attribute value specifies the charset parameter in its MIME media type value.

727 If the "notify-charset" attribute is not supplied, the charset supplied in the "attributes-
728 charset" Operation attribute SHALL be used, if the charset value is supported by the
729 Printer, else the Printer object shall use the Printer's "charset-configured" value.

730 **5.1.6 notify-natural-language (naturalLanguage)**

731 The client OPTIONALLY supplies this attribute as a member of the "job-notify"
732 Operation attribute. The Printer object OPTIONALLY supports this attribute if it
733 supports the "job-notify" Operation attribute.

734 This attribute specifies the natural language for the IPP object to use in the human
735 readable part of the notification content is sent to the notification recipients that the client
736 supplied in this same collection value. If this attribute is not supported or the supplied
737 value is not supported, the IPP Printer SHALL return the attribute in the Unsupported
738 Attributes Group but still accept the operation, as with all create operations. If this
739 attribute is not supplied or the attribute or value is not supported by the Printer object, the
740 natural language supplied in the "attributes-natural-language" create operation attribute
741 SHALL be used, if that natural language value is supported by the Printer, else the Printer
742 object SHALL use the Printer's "natural-language-configured" value. See the Print-Job
743 operation in [ipp-model].

744 **5.1.7 notify-additional-attributes (1setOf keyword)**

745 The client OPTIONALLY supplies this attribute as a member of the "job-notify"
746 Operation attribute. The Printer object OPTIONALLY supports this attribute if it
747 supports the "job-notify" Operation attribute.

748 This attribute specifies the additional attributes that the requester wishes to be included in
749 the notification content, in addition to the fixed set that depends on the event as shown in
750 the table in section 6. If this attribute is not supported or not supplied by the client, the
751 Printer object SHALL supply the fixed set of attributes indicated in section 6 depending
752 on the event being requested.

753 **6 Operations to Subscribe and Un-subscribe for notifications**

754 There are two new OPTIONAL operations to allow a client or server to subscribe for
755 Printer object events without submitting a job. An IPP Printer SHALL support both of
756 these operations, if it supports either one. If an IPP Printer supports these operations, it
757 SHALL also support the "job-notify" attribute in the create operations as described in
758 section 5.

759 These new operations are intended for use by servers that control printers, by clients used
760 by operators/administrators that manage printers, and by applications that collect
761 accounting data.

762 **6.1 Subscribe-For-Event-Notifications Operation**

763 This OPTIONAL operation allows a client to subscribe with the Printer object to be
764 notified when identified events happen to the device(s) that the Printer object is
765 representing without requiring that the client submit jobs. In the request, the client
766 supplies the set of Job event group names and/or Printer event group names in which the
767 notification-recipient(s) are interested. In the response, the Printer object returns a list of
768 the current subscriptions, including the new one requested by this operation.

769 This operation is intended for use by system operators and administrators that have a long
770 term interest in the events without submitting jobs. It is also intended to be used by
771 servers that control IPP Printers. Finally, it is also intended to be used by accounting
772 applications that need to be notified when jobs complete.

773 The possible names of Job and Printer event groups are the same as for use in the "job-
774 notify" Operation attribute in create requests. See section 5.1.2. An IPP object SHALL
775 support all event groups. Support of all of the events in a group is not required.

776 **ISSUE 16: Ok if all groups are required for conformance?**

777 **6.1.1 Subscribe-For-Event-Notifications Request**

778 The following sets of attributes are part of the Subscribe-For-Event-Notifications
779 Request:

780 Group 1: Operation Attributes

781 Target:

782 The "printer-uri" operation attribute which is the target for this operation as
783 described in section 3.1.3.

784

785 Natural Language and Character Set:

786 The "attributes-charset" and "attributes-natural-language" attributes as described
787 in section 4.3.23 and 4.3.24.

788

789 Requesting User Name:

790 The "requesting-user-name" attribute SHOULD be supplied by the client as
791 described in section 8.3.

792

793 "printer-notify" (collection) :
794 The client SHALL supply a "printer-notify" Operation attribute that MUST
795 specify the notification-recipient(s), and MAY specify additional information
796 about the subscription. The Printer object SHALL support this Operation
797 attribute (if it supports this OPTIONAL operation). The value of this attribute is
798 one collection value. The collection value SHALL contain a "notify-recipients"
799 member attribute and MAY contain any of the other *member* attributes defined for
800 use with the "job-notify" Operation attribute in create operations (see section 5.1).
801 If the client omits this attribute, the Printer SHALL reject the operation and return
802 the 'client-error-bad-request' status code.

803
804 Note: only one collection value is permitted, so that each collection value will
805 have its own "subscription-id".
806

807 The Printer object SHALL validate that this client is permitted to subscribe for Printer
808 notifications. The means for configuring the permissions is outside the scope of this
809 specification. If a requester is not permitted to subscribe for Printer notifications, the IPP
810 Printer SHALL reject the request and return the 'client-error-authenticated' or 'client'-
811 error-not-authorized' status code.

812 If the same subscription (same client and same collection values) has already been made
813 as indicated in one of the collection values of the Printer object's "printer-notify"
814 Description attribute, the IPP Printer SHALL reject the request and return the 'client-
815 error-not-possible' status code.

816 **ISSUE 17: Or should we add a new status code that is more specific, such as 'client-**
817 **error-already-subscribed'.**

818 If the IPP Printer object accepts the request, it SHALL add the subscription collection
819 value to the Printer object's "printer-notify" attribute. The Printer object SHALL add a
820 "notify-subscription-id" member attribute with a unique integer id. This id is used to un-
821 subscribe using the Un-Subscribe-For-Event-Notifications operations.

822 **6.1.2 Subscribe-For-Event-Notifications Response**

823 The Printer object returns the following sets of attributes as part of Subscribe-For-Event-
824 Notifications Response:

825 Group 1: Operation Attributes

826 Status Code and Message:

827 The response includes the MANDATORY status code and an OPTIONAL
828 "status-message" (text) operation attribute as described in section 3.1.5.

829

830 Natural Language and Character Set:

831 The "attributes-charset" and "attributes-natural-language" attributes as described
832 in section 3.1.4.2.
833

834 "subscription-id" (integer(1:MAX)):

835 The unique integer id for the accepted subscription to be used to un-
836 scribe using the Un-Scribe-For-Event-Notifications operation. This value SHOULD NOT be
837 re-used too soon after subscription in order to avoid confusion in subsequent Un-
838 Scribe-For-Event-Notification operations.

839

840 Group 2: Unsupported Attributes

841 This is a set of Operation (member) attributes supplied by the client (in the
842 request) that are not supported by the Printer object or that conflict with one
843 another (see sections 15.3 and 15.4).

844

845 Group 3: Printer Object Attributes

846 The updated "printer-notify" attribute that contains the requested subscription
847 supplied in this operation request, along with any that have been previously
848 subscribed by any client.

849

850 **6.2 Un-Subscribe-For-Event-Notifications Operation**

851 This OPTIONAL operation allows a client to un-subscribe with the Printer object for
852 event notifications that had been subscribed to previously using the Subscribe-For-Event-
853 Notification operation. In the request, the client supplies the notify-subscription-id
854 attribute that the Printer object created and returned in the Subscribe-For-Event-
855 Notifications operation. In the response, the Printer object returns a list of the current
856 subscriptions which SHALL NOT include the one removed by this operation.

857 This operation is intended for use by system operators and administrators that have a long
858 term interest in the events without submitting jobs. It is also intended to be used by
859 servers that control IPP Printers. Finally, it is also intended to be used by accounting
860 applications that need to be notified when jobs complete.

861 **6.2.1 Un-Subscribe-For-Event-Notifications Request**

862 The following sets of attributes are part of the Un-Subscribe-For-Event-Notifications
863 Request:

864 Group 1: Operation Attributes

865 Target:

866 The "printer-uri" operation attribute which is the target for this operation as
867 described in section 3.1.3.

868

869 Natural Language and Character Set:

870 The "attributes-charset" and "attributes-natural-language" attributes as described
871 in section 3.1.4.1.

872

873 Requesting User Name:

874 The "requesting-user-name" attribute SHOULD be supplied by the client as
875 described in section 8.3.

876

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

878 The client SHALL supply a "notify-subscription-id" Operation attribute that
879 specifies a subscription id assigned by the Printer object in a previous Subscribe-
880 For-Event-Notifications. The Printer object MUST support this Operation
881 attribute (if it supports this OPTIONAL operation). If the client omits this
882 attribute, the Printer SHALL reject the operation and return the 'client-error-bad-
883 request' status code.

884

885 The Printer object SHALL validate that this client is permitted to un-subscribe
886 notifications in general and this notification subscription in particular. The means for
887 configuring the permissions is outside the scope of this specification.

888 If a requester is not permitted to un-subscribe for notifications in general or for the
889 requested subscription, the IPP Printer SHALL reject the request and return the 'client-
890 error-authenticated' or 'client-error-not-authorized' status code. The means for keeping
891 track of which clients requested each subscription is not specified by this document and is
892 implementation dependent. For example, an implementation might add an additional
893 "client-id" member attribute to each subscription value of the Printer object's "printer-
894 notify" Description attribute, that is not returned to non-privileged users.

895 If the value of the "notify-subscription-id" is not found, the IPP Printer SHALL reject the
896 request and return the 'client-error-not-found' status code.

897 If the IPP Printer object accepts the request, it SHALL remove the requested event
898 notification subscription from the Printer object's "printer-notify" attribute. Clients
899 SHOULD remove subscriptions that are no longer wanted using this operation.

900 **6.2.2 Un-Subscribe-For-Event-Notifications Response**

901 The Printer object returns the following sets of attributes as part of the Un-Subscribe-For-
902 Event-Notifications Response:

903 Group 1: Operation Attributes

904 Status Code and Message:

905 The response includes the MANDATORY status code and an OPTIONAL
906 "status-message" (text) operation attribute as described in section .

907

908 Natural Language and Character Set:

909 The "attributes-charset" and "attributes-natural-language" attributes as described
910 in section 3.1.4.2.

911

912 Group 2: Unsupported Attributes

913 This is a set of Operation (member) attributes supplied by the client (in the
914 request) that are not supported by the Printer object or that conflict with one
915 another (see sections 15.3 and 15.4).

916

917 Group 3: Printer Object Attributes

918 The updated "printer-notify" attribute that no longer contains the event
919 notification subscription that was requested to be removed.

920 **7 Job Object Description attributes for Job Notification**

921 This section specifies the Job Description attributes for notification.

922 **7.1 "job-notify" (1setOf collection)**

923 This attribute specifies one or more collections of events, notification-recipients, and
924 other member attributes that the client supplied in the "job-notify" Operation attribute of
925 the create request. The Printer object SHALL support this Job attribute if it supports the
926 "job-notify" Operation attribute.

927 The IPP Printer object SHALL populate the value(s) of this attribute with the collection
928 value(s) supplied by the "job-notify" Operation attribute in the create operation that
929 created this job. See the description of the "job-notify" Operation attribute for the
930 complete specification of the semantics of this Job Description attribute.

931 **8 Printer Object Description attributes for Notification**

932 This section specifies the Printer object Description attributes for Job and Printer
 933 Notifications. If the Printer object supports the "job-notify" Operation attribute for the
 934 Print-Job, Print-URI, and Create-Job operations, then the Printer object SHALL support
 935 the following supported Printer object Description attributes in the second column in
 936 Table 1 that correspond to the "job-notify" member attributes supported.

937 If the Printer object supports the Subscribe-For-Event-Notifications operations, then the
 938 Printer object SHALL support the following Printer object Description attributes in the
 939 third column in Table 1 that correspond to the "printer-notify" member attributes
 940 supported.

941 Note: These Printer attributes are specified as separate Printer object attributes, rather
 942 than as member attributes of a Printer object's collection attribute, since any combination
 943 of values may be used for any of the attributes.

944 **Table 1 - Printer Description Attributes for Job and Printer Notifications**

945	+-----+-----+-----+			
946		Collection member	Job Notification	Printer Notification
947		attribute	support Attributes	support Attributes
948		+-----+-----+-----+		
949		notify-event-	job-notify-event-	printer-notify-event-
950		groups	groups-supported	groups-supported
951		(1setOf type2	(1setOf type2 keyword	(1setOf type2 keyword
952		keyword)		
953		+-----+-----+-----+		
954		notify-	job-notify-schemes	printer-notify-schemes
955		recipients	-supported	-supported
956		(1setOf uri)	(1setOf uriScheme)	(1setOf uriScheme)
957		+-----+-----+-----+		
958		notify-	job-notify-content-	printer-notify-content-
959		content-type	type-supported	type-supported
960		(mimeType)	(1setOf mimeType)	(1setOf mimeType)
961		+-----+-----+-----+		
962		notify-charset	job-notify-charset-	printer-notify-charset-
963		(1setOf charset)	supported	supported
964			(1setOf charset)	(1setOf charset)
965		+-----+-----+-----+		
966		notify-	job-notify-natural-	printer-notify-natural-
967		natural-language	language-supported	language-supported
968		(naturalLanguage)	(1setOf	(1setOf
969			naturalLanguage)	naturalLanguage)
970		+-----+-----+-----+		
971		notify-	job-notify-	printer-notify-
972		additional-	additional-	additional-
973		attributes	attributes-supported	attributes-supported
974		(1setOf keyword)	(1setOf keywords)	attributes-supported
975		+-----+-----+-----+		

976

977 **8.1 Job Notification Support Printer Description attributes**

978 The Job Notification Support Printer object Description attributes (column 2 in Table 1)
979 specify the supported values for the corresponding member attributes of the "job-notify"
980 Operation collection attribute used in the job create operations. The value of the Printer
981 object's "job-notify-recipients-supported" attribute is a 'uriScheme'. The Printer object
982 SHALL use the values of this attribute to validate the scheme supplied by the client in the
983 "notify-recipients" member attribute.

984 For example, if a Printer object supports:

- 985 1) 'mailto:' method for the 'job-completion' event groups using English, French, U.S.
986 English, and German and supporting additional attributes: "job-uri", "job-name",
987 "job-originating-user-name", "number-of-documents", "job-state", "sides",
988 "finishing"
- 989 2) 'sense' and 'ipp-tcp-ip-socket' methods for the 'job-delivery', 'job-progress', and
990 'job-completion' event groups in English only

991 a system administrator could configure the following Printer attributes":

992 "job-notify-schemes-supported" = 'mailto', 'sense', 'ipp-tcp-ip-socket'

993 "job-notify-event-groups-supported" = 'job-delivery', 'job-progress', 'job-
994 completion'

995 "job-notify-natural-language-supported" = 'en', 'fr', 'en-us', 'de'

996 "job-notify-additional-attributes-supported" = 'job-uri', 'job-name',
997 'job-originating-user-name', 'number-of-documents',
998 'job-state', 'sides', 'finishing'

999 **ISSUE 18: Should an administrator be able to configure so that the groups supported is**
1000 **less than all of them. All of them are required for conformance?**

1001

1002 Note: the fact that not all events are supported for the mailto scheme, or that not all
1003 languages are supported for the 'sense' and 'ipp-tcp-ip-socket' methods is not represented,
1004 since the collection mechanism is not used to represent the supported attributes. If the
1005 client supplies a combination that is not supported, the Printer object SHALL accept the
1006 create request (independent of the value of the "ipp-attribute-fidelity" attribute supplied
1007 by the client), make suitable substitutions, and return the attributes that are ignored or
1008 substituted in the create operation response.

1009 **ISSUE 19: Are we still ok with not making these "xxx-supported" attributes member**
1010 **attributes of one collection "notifications-supported" Printer Description attribute?**
1011 **Or maybe two collections: "job-notifications-supported" and "printer-notifications-**
1012 **supported" Printer Description attributes?**

1013 **8.2 Printer Notification Support Printer Description attributes**

1014 The Printer Notification Support Printer object Description attributes (column 3 in Table
1015 1) specify the supported values for the corresponding member attributes of the "printer-
1016 notify" Operation collection attribute used in the Subscribe-For-Event-Notifications
1017 operation. The value of the Printer object's "printer-notify-recipients-supported" attribute

1018 is a 'uriScheme'. The Printer object SHALL use the values of this attribute to validate the
1019 scheme supplied by the client in the "notify-recipients" member attribute. See section 8.1
1020 for an example, except change all "job-xxx" attributes to "printer-xxx" attributes.

1021 **9 Notification Content definitions**

1022 Just as applications need a defined (extendable) set of notifications, they also need a fixed
1023 structure and reliable notification content. The notification content depends on the event.
1024 Job events in a Job Submission Subscription via a create operation ONLY apply to the
1025 job created. Job events in a Printer Subscription apply to ALL jobs.

1026 An IPP Printer object MAY also implement the "notify-additional-attributes" Operation
1027 member attribute in order to allow a client to request additional attributes over and above
1028 the fixed set shown in Table 3.

1029 [Some delivery methods, such as SNMP, do not support the requester requesting
1030 additional attributes; the notification recipient will have to explicitly use a Get-Job-
1031 Attributes or Get-Printer-Attributes operation to get additional attributes about the job or
1032 device.]

1033 [IPP does not have some of the job progress attributes that the PWG Job Monitoring MIB
1034 has. These are indicated with "-" in the IPP attribute column.]

1035 **ISSUE 20: Should we add the job progress attributes to IPP that the PWG Job**
1036 **Monitoring MIB returns in an SNMP trap so that accounting programs can get the same**
1037 **attributes with IPP?**

1038 The following sub-sections specify those content attributes that are not Job or Printer
1039 attributes:

1040 9.1 "time-at-event" (integer (0:MAX))

1041 This notification content attribute indicates the point in time at which the event occurred.
1042 In order to populate this attribute, the Printer object uses the value in its "printer-up-time"
1043 attribute at the time the job or printer event occurred. This notification content attribute
1044 SHALL be part of all notification contents for all events.

1045 NOTE: The "time-at-event" and "printer-up-time" are in units of seconds, not one
1046 hundreds of a second (like prtAlertTime and sysUpTime). Thus the attribute name is
1047 "time-at-event", rather than "prt-att-18-9-r" (where "r" is the row in the alert table of this
1048 alert), since the value has different semantics.

1049 9.2 "event" (keyword)

1050 This notification content attribute indicates the event (not the event group) that occurred.
1051 This notification content attribute SHALL be part of all notification contents for all
1052 events, so that a notification recipient can determine which event occurred, even though
1053 implementors add their own events and/or other MIBs may use their MIB-specific alert
1054 codes in the "alert-code" notification content attribute. For example, for any Printer
1055 errors, the value of the "event" notification content attribute SHALL be the 'printer-error'
1056 keyword.

1057 ISSUE 21: Ok, that the "event" attribute always occurs in the notification content, even
1058 when there is also the prtAlertCode from the Printer MIB, so that we can add other MIB
1059 alerts in the future, too?

1060 **9.2.1 Job event notification content**

1061 Table 3 shows the notification content attributes that SHALL be included in any
 1062 notification content for a Job event.

1063 **Table 3 - Mandatory attributes for notification content depending on the Job event**

IPP attribute (content)	JMP VarBind object/attribute (content)	Job Event (not Event Group)			
		job-received	job-started-processing, job-held, job-released	job-warning, job-error	sheet-completed, collated-copy-completed, job-completed, job-aborted, job-canceled
Common to Job and Printer events:					
printer-uri	hrDeviceIndex	Yes	Yes	Yes	Yes
time-at-event	jmAlertTime (new)	Yes	Yes	Yes	Yes
event	event	Yes	Yes	Yes	Yes
Specific to Job events:					
job-id	jmJobIndex	Yes	Yes	Yes	Yes
number-of-intervening-jobs	jmNumberOfInterveningJobs	Yes	Yes	Yes	-
job-k-octets	jmJobKOctetsPerCopyRequested	-	Yes	Yes	Yes
job-k-octets-processed	jmJobKOctetsProcessed	-	-	Yes	Yes
job-impressions	jmJobImpressionsPerCopyRequested	-	Yes*	Yes*	Yes*
-	impressionsInterpreted	-	-	Yes	Yes
job-impressions-completed	jmJobImpressionsCompleted	-	-	Yes	Yes
copies	jobCopiesRequested	-	-	Yes	Yes
-	impressionsCompletedCurrentCopy	-	-	Yes	Yes
-	sheetCompletedCopyNumber	-	-	Yes	Yes
-	sheetCompletedDocumentNumber	-	-	Yes	Yes

IPP attribute (content)	JMP VarBind object/attribute (content)	Job Event (not Event Group)			
		job-received	job-started-processing, job-held, job-released	job-warning, job-error	sheet-completed, collated-copy-completed, job-completed, job-aborted, job-canceled
-	jobCollationType	-	-	Yes	Yes
-	outputBin	-	-	-	Yes**
job-state	jmJobState	-	-	Yes	-
job-state-reasons	jmJobStateReasons1	Yes	Yes		Yes

1064

1065 '-' indicates that the attribute SHALL NOT be included in the notification content.

1066

1067 * The IPP Printer object will treat jmJobImpressionsPerCopyRequested in the following
 1068 manner. If explicitly *passed in on submission*, this will be the value used. If there is no
 1069 value passed in on submission, then the *implicit value, derived from the final number of*
 1070 *impressionsInterpreted for the first copy will be used.*

1071

1072 ** **outputBin** may be multi-valued

1073 Note: the 'job-delivery' group has different patterns of attributes sent in the notification
 1074 content, so that the IPP Printer object would have to subscribe with the SNMP agent
 1075 using several different SNMP trap OIDs because the VarBind lists must be different.

1076 NOTE: The following objects and attributes have not been included in the fixed set of
 1077 attributes that SHALL be returned for the indicated reasons (they MAY be requested in
 1078 implementations that support the "ipp-notify-additional-attributes" attribute):

1079 1) "job-state" (JMP jmJobState) - the event indicates the job's new state.

1080 **ISSUE 22:** But "job-state" does appear in the table for certain events?1081 **ISSUE 23:** What about "job-state-reasons"?

1082 2) "job-owner" (JMP jobOwner) - the notification recipient should know who the
 1083 owner is. Also the owner is a string, so it can be long. The total size of the
 1084 content must fit in the maximum size of a PDU for any transport, which is
 1085 about 500 octets or so (for IPX).

1086 3) For an IPP device, the jmJobSubmissionID is "job-uri", at least the last 47
 1087 octets of it.

1088 **9.2.2 Printer event notification content**

1089 Table 4 shows the notification content attributes that SHALL be included in any
 1090 notification content for a Job event. The following sub-sections specify those attributes
 1091 that are neither Printer attributes not Printer MIB alert objects:

1092 **9.2.2.1 "device-name" (name)**

1093 This Printer attribute specifies the device name of the device generating the event. This
 1094 attribute is needed for those IPP Printer objects that support more than one device (so-
 1095 called fan-out). See [ipp-model]. This attribute is being added as a Printer attribute as
 1096 well (see [mib-access]).

1097 The other Printer attributes that are contained in a notification-content are the attributes
 1098 that would be returned in a Get-Printer-Attributes Response, when the "which-device"
 1099 Operation attributes were supplied with the value equal to that of the "device-name"
 1100 attribute. For example, the "printer-state" attribute is returned as if the device identified
 1101 by "device-name" were the only device that the IPP Printer controlled. In other words,
 1102 the Printer attributes returned in a notification are specialized to the device that generated
 1103 the event (see [mib-access] for more explanation of this specialization).

1104 **9.2.2.2 "which-alert-row" (keyword)**

1105 This notification content attribute identifies the row in the Printer MIB alert table. The
 1106 value is a keyword of the form: "prt-row-18-*r*" where "*r*" is the decimal digits
 1107 representing the alert row number in the prtAlertTable that was added to generate this
 1108 alert. The value is a keyword that the client MAY supply directly in a Get-Printer-
 1109 Attributes operation to get the entire alert group row that causes this alert.

1110 **Table 4 - Mandatory attributes for notification content depending on the Printer**
 1111 **event**

IPP attribute (content)	Printer MIB VarBind object (content)	Printer Event (not Event Group)		
		printer- report	printer- warning	printer- error
Common to Job and Printer events:				
printer-uri (uri)	hrDeviceIndex	Yes	Yes	Yes
time-at-event (integer(0:MAX))	prtAlertTime	Yes	Yes	Yes
event (enum)	-	Yes	Yes	Yes
Specific to Printer events:				
device-name	-	Yes	Yes	Yes
which-alert-row (keyword)	prtAlertIndex	Yes	Yes	Yes
prt-att-18-2- <i>r</i> (enum)	prtAlertSeverityLevel	Yes	Yes	Yes

IPP attribute (content)	Printer MIB VarBind object (content)	Printer Event (not Event Group)		
		printer- report	printer- warning	printer- error
prt-att-18-3-r (enum)	prtAlertTrainingLevel	Yes	Yes	Yes
prt-att-18-4-r (enum)	prtAlertGroup	Yes	Yes	Yes
prt-att-18-5-r (integer(1:MAX))	prtAlertGroupIndex	Yes	Yes	Yes
prt-att-18-6-r (integer(- MAX:MAX))	prtAlertLocation	Yes	Yes	Yes
prt-att-18-7-r (enum)	prtAlertCode	Yes	Yes	Yes
prt-att-18-8-r (text(255))	prtAlertDescription	Yes	Yes	Yes
printer-state (type1 enum)	-	Yes	Yes	Yes
printer-state-reasons (1setOf type2 keyword)	-	Yes	Yes	Yes

1112

1113 ' ' indicates that the attribute SHALL NOT be included in the notification content.

1114

1115 **ISSUE 24:** Ok that I changed the data types that go with prtAlertGroup and
 1116 prtAlertGroupIndex from keyword back to the ones in the Printer MIB (except time), so
 1117 that we could use the values returned from the Printer MIB directly.

1118

10 Encoding

1119 The new 'collection' attribute syntax will use the 0x34 tag value that has been reserved in
 1120 the IPP/1.0: Protocol Specification for this purpose.

1121

11 References

1122 [cohen]

1123 J. Cohen, et al, General Event Notification Architecture Base. April 23, 1998,
 1124 work-in-progress, <draft-cohen-gena-p-base-00.txt>.

1125 [draft-prtmb]

1126 R. Turner, Printer MIB, work-in-progress, <draft-ietf-printmib-mib-info-03.txt>,
 1127 October 1997.

1128 [ipp-model]

1129 Isaacson, S, deBry, R, Hastings, T, Herriot, R, Powell, P, "Internet Printing
 1130 Protocol/1.0: Model and Semantics".

1131 [ISO10646-1]

1132 ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-Octet
 1133 Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane,
 1134 JTC1/SC2."

- 1136 [ISO8859-1]
1137 ISO/IEC 8859-1:1987, "Information technology -- 8-bit One-Byte Coded
1138 Character Set - Part 1: Latin Alphabet Nr 1", 1987, JTC1/SC2.
- 1139 [mib-access]
1140 S. Isaacson, T. Hastings, R. Herriot, K. Schoff, IPP Device and MIB access,
1141 Version 0.03, May 5, 1998, work-in-progress, <ipp-mib-access-980505.pdf>.
- 1142 [req]
1143 R. deBry, Requirements for IPP Notifications, March 11, 1998, work-in-progress,
1144 <draft-ietf-ipp-not-01.txt>.
- 1145 [RFC-1759]
1146 R. Smith, F. Wright, T. Hastings, S. Zilles, J. Gyllenskog, Printer MIB, March
1147 1995.
- 1148 [RFC-2044]
1149 F. Yergeau, "UTF-8, a transformation format of Unicode and ISO 10646", RFC
1150 2044, October 1996.
- 1151 [RFC-2046]
1152 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed
1153 & N. Borenstein. November 1996. (Obsoletes RFC1521, RFC1522, RFC1590),
1154 RFC 2046.
- 1155 [url-reg]
1156 R. Petke, Registration Procedures for URL Scheme Names, April 30, 1998, work-
1157 in-progress, <draft-ietf-urlreg-procedures-01.txt>
- 1158 [US-ASCII]
1159 Coded Character Set - 7-bit American Standard Code for Information Interchange
1160 (ASCII), ANSI X3.4-1986. This standard is the specification of the US-ASCII
1161 charset.

1162 **12 Copyright Notice**

1163 None,

1164 **13 Author's Address**

1165 Tom Hastings
1166 Xerox Corporation
1167 701 S. Aviation Blvd.
1168 El Segundo, CA 90245
1169
1170 Phone: 310-333-6413
1171 Fax: 310-333-5514
1172 e-mail: hastings@cp10.es.xerox.com
1173

1174 Scott A. Isaacson
1175 Novell, Inc.
1176 122 E 1700 S
1177 Provo, UT 84606
1178
1179 Phone: 801-861-7366
1180 Fax: 801-861-2517
1181 e-mail: sisaacson@novell.com
1182
1183 Harry Lewis
1184 IBM Corporation
1185 6300 Diagonal Hwy
1186 Boulder, CO 80301
1187
1188 Phone: (303) 924-5337
1189 Fax: (303) 924-4662
1190 Email: harryl@us.ibm.com

1191 **14 Appendix - Specification for the IPP collection attribute syntax**

1192 This appendix is the complete specification for the new 'collection' attribute syntax that
1193 the notification specification uses. Other future extensions, both registered and private,
1194 will make use of this new attribute syntax.

1195 This mechanism had originally been named 'dictionary', but we agreed to change it since
1196 the member attributes are not ordered, typically.

1197 **There are two issues highlighted in yellow.**

1198 **14.1 Problem Statement**

1199 There is no good way to add attributes that contain several fields, whether the fields are
1200 mandatory or optional. Instead of each new attribute that needs more than one field
1201 (struct), requiring an ad hoc attribute syntax, such as we have done for the 'resolution'
1202 attribute syntax for use in the "printer-resolution" attribute, it would be desirable to have
1203 a simple, general mechanism for representing multi-field values. It would also be
1204 desirable to allow fields to be omitted, when the attribute specification allows that. This
1205 mechanism would be useful for both new attributes that we might register as extensions
1206 to be used with the IPP standard, or that implementers might implement as private
1207 extensions.

1208 **14.2 Summary of the attribute syntax alternative**

1209 A number of alternatives were considered. See the last section for a list and the reasons
1210 for their rejection.

1211 The proposal is to add a new attribute syntax, called 'collection'. Any attribute of type
1212 'collection' shall have a value that is a set of unordered attributes, where each attribute
1213 MAY be single-valued or multi-valued as specified for the collection attribute. Since the
1214 attribute value has a length, like any other attribute value, IPP objects not supporting the
1215 attribute can easily skip over the entire attribute value, i.e., skip over the entire set of
1216 attributes that make up the collection value.

1217 **14.3 Requirements for and properties of the suggested collection** 1218 **mechanism**

1219 The collection mechanism for use with IPP needs to have the following semantic
1220 properties:

- 1221 1. The collection mechanism provides a way to supply and query a set of attributes as a
1222 logical unit. Then each 'field' that is present in the collection would be self-
1223 identifying by its attribute name.
- 1224 2. The attributes in a collection are unordered. Therefore, an IPP object **MUST** be able
1225 to accept attributes in a collection in any order.
- 1226 3. The semantics of a collection attribute specifies which attributes in a collection
1227 instance are **MANDATORY** for the IPP object to support and which are **OPTIONAL**
1228 for the IPP object to support when the IPP object supports that collection attribute.

- 1229 4. The semantics of a collection attribute specifies which attributes in a collection
1230 instance are required for the requester to supply and which the requester may omit.
- 1231 5. A collection attribute could be single valued, i.e., with one collection value consisting
1232 of a set of attributes, or could be multi-valued, i.e., with multiple collection values,
1233 each consisting of a set of attributes.
- 1234 6. An attribute in a collection value can be single valued or multi-valued as well
1235 according to the specification of the collection attribute.
- 1236 7. As with all attribute values, if an IPP object does not support a collection attribute, it
1237 must be easy for the IPP object to ignore each collection attribute value.
- 1238 8. The syntax of each collection value is the same as a group of attributes in a request or
1239 response, so each attribute in a collection value instance has its keyword name, its
1240 attribute syntax code, and its value.
- 1241 9. An implementer MAY support additional registered or private attributes in a
1242 collection. In other words, a collection is extensible, just like an attribute group in an
1243 operation or response.
- 1244 10. Since support of all possible combinations of values for all attributes in a collection
1245 value may not be supported by some implementations, there should be a way for the
1246 IPP object to indicate which combinations of values are supported. For example,
1247 300x300, 600x300, and 600x600, but not 300x600 dpi.
- 1248 11. Finally, an attribute in a collection value can be itself a collection, so that nesting
1249 could be allowed, if the specification of a collection attribute allowed a collection
1250 attribute to be contained in its collection.

1251 **14.4 Examples of collection usage**

1252 This section describes four collection Job Template examples: "printer-resolution", "job-
1253 notify", "job-start-page-contents", and "postal-mail-disposition" attributes. The "printer-
1254 resolution" attribute only contains single-valued attributes, while the "printer-resolution-
1255 supported" and "job-notify" attribute contains multi-valued collection attributes, i.e.,
1256 contain more than one collection as a value of an attribute.

1257 **14.4.1 Example a: "printer-resolution" Job Template attribute**

1258 For example, the new "printer-resolution" attribute was defined using a very ad hoc
1259 'resolution' attribute syntax. Had we had the collection attribute syntax, we might have
1260 chosen to use it here, though we wouldn't have had to either. If we did use the 'collection'
1261 attribute syntax for the "resolution", the attribute value would contain the following
1262 attributes: "resolution", "cross-feed-resolution", and "resolution-units". We could have
1263 also specified that the "cross-feed-resolution" attribute is OPTIONAL and when omitted,
1264 the cross-feed resolution is the same as the "resolution" attribute, since most resolutions
1265 are the same in both directions. We could have also specified that the "resolution-units"
1266 attribute is OPTIONAL and when omitted, the resolution units are dots per inch.

1267 **ISSUE 25: Should we also allow the member attributes of a collection to be supplied by**
 1268 **themselves when the client does not want to group them or is that just an unnecessary**
 1269 **alternative form?**

1270 The specification for the "printer-resolution" collection attribute is that its collection
 1271 value is made up of the following attributes:

1272	Attribute name	syntax	in request
1273	-----	-----	-----
1274	"resolution"	integer	required
1275	"cross-feed-resolution"	integer	optional
1276	"resolution-units"	enum	optional

1277 For a simplified collection attribute notation, lets use:

1278 `"collection attribute" = { set of attributes and values }`

1279 where a set of { } is used to group a single collection value.

1280 For example, a client supplying a resolution of 600 x 300 would be indicated in examples
 1281 using the following notation:

1282 `"printer-resolution" = { "resolution" = '600', "cross-feed-resolution" = '300' }`

1283 **14.4.1.1 "printer-resolution-default" example**

1284 The Printer object could represent the "printer-resolution-default" default values as a
 1285 single collection value. For example, a system administrator (or the printer vendor) could
 1286 specify the default as:

1287 `"printer-resolution-default" = { "resolution" = '300' }`

1288 **14.4.1.2 "printer-resolution-supported" example and validation of** 1289 **collections**

1290 The Printer object could indicate the combinations of resolutions that are supported by
 1291 three sets of collection values which represent 300x300, 600x300, and 600x600 dpi,
 1292 respectively (300x600, say, is not supported). Such a configured situation could be
 1293 represented in examples as:

1294 `"printer-resolution-supported" = {`
 1295 `{ "resolution" = '300' },`
 1296 `{ "resolution" = '600', "cross-feed-resolution" = '300' },`
 1297 `{ "resolution" = '600' } }`

1298 **14.4.2 Example b: "job-notify" Operation attribute**

1299 In order to meet the IPP notification requirements, the requester must be able to supply
 1300 one or more notification profile values, where each profile value consists of a set of "job-
 1301 notify-events", one "job-notify-method", multiple "job-notify-recipients", one "job-
 1302 notify-natural-language", one "job-notify-charset", and possibly multiple "job-notify-
 1303 additional-requested-attributes". There might be a similar multi-valued "printer-notify"
 1304 Printer object collection attribute that is set by means outside of the IPP/1.0 protocol, but

1305 is independent of jobs, so that they would specify notification to operators. Both the
 1306 "job-notify" and the "printer-notify" collection attributes are MULTI-VALUED and
 1307 contain attributes that themselves are MULTI-VALUED.

1308 The "job-notify" Operation collection attribute would have collection values with the
 1309 following syntax:

1310	Attribute name	syntax	in request
1311	-----	-----	-----
1312	"notify-event-groups"	1setOf enum	optional
1313	"notify-recipients"	1setOf uri	required

1314

1315 A Print-Job request could supply the collection attribute values in order to send
 1316 immediate 'job-aborted' and 'job-canceled' events to Smith (himself) and e-mail 'job-
 1317 completion' to Jones and White. A notation for this example could be to use a set of {} to
 1318 indicate each

```

1319     "job-notify" = { { "notify-event-groups" = 'job-errors'
1320                       "notify-recipients" = 'Smith' },
1321                   { "notify-event-groups" = 'job-completion'
1322                     "notify-recipients" = 'Jones', 'White' } }
1323
  
```

1323

1324 **14.4.3 Example c: Start page fields supplied by the end-user**

1325 As a third example of a collection, an attribute could represent the fields that the
 1326 submitter wishes to be printed on the job-start page. The name of the attribute might be:
 1327 "job-start-page-contents". The collection value might include: "job-name", "user-name",
 1328 "job-comment", "account-name", "job-disposition", "job-delivery", etc. where the values
 1329 of the attributes in the collection are printed after each attribute name on the job-start-
 1330 page.

1331	Attribute name	syntax	in request
1332	-----	-----	-----
1333	"job-name"	name	required
1334	"user-name"	name	required
1335	"job-comment"	text	optional
1336	"account-name"	name	optional
1337	"job-disposition"	keyword	optional
1338	"job-delivery"	1setOf keyword	optional

1339 **14.4.4 Example d: Postal mailing address**

1340 As a final example of a collection, an attribute could represent a postal mailing address
 1341 for the output. The name of the attribute might be "postal-mail-disposition" and it would
 1342 be multi-valued, i.e., 1setOf collection. The collection attribute might have the following
 1343 specification and support requirements if the "postal-mail-disposition" collection attribute
 1344 is supported at all:

1345	Attribute name	syntax	in request	IPP object support
1346	"addressee-name"	text	required	MANDATORY
1347	"company-name"	text	optional	OPTIONAL
1348	"internal-mail-stop"	text	optional	OPTIONAL

1349	"apartment-number	text	optional	MANDATORY
1350	"street-address"	text	required	MANDATORY
1351	"city-or-town	text	required	MANDATORY
1352	"state"	text	required	MANDATORY
1353	"postal-zone	text	required	MANDATORY
1354	"country"	text	optional	OPTIONAL
1355	"phone-numbers	1setOf text	optional	OPTIONAL
1356				

1357 **14.5 Detailed description 'collection' attribute syntax**

1358 Register the following attribute syntax, written in the style of section 4.1 Attribute
1359 Syntaxes of the IPP Model specification:

1360 4.1.n 'collection'

1361 A set of unordered attributes, where each attribute MAY be single-valued or multi-valued
1362 as specified for the collection attribute. As in the attribute sets that are passed in
1363 operations, an IPP object SHALL accept the attributes in a collection value in any order
1364 and no attribute SHALL occur more than once in a collection. However, if the same
1365 attribute does occur more than once in a collection by error, the IPP object SHALL reject
1366 the operation and SHALL return the 'client-error - bad syntax' error code.

1367 The specification of the attribute that uses the 'collection' attribute syntax SHALL
1368 specify:

- 1369 1. as with any attribute, whether the attribute is single-valued (attribute syntax =
1370 'collection') or multi-valued (attribute-syntax = '1setOf collection').
- 1371 2. For each attribute in the collection value, whether the IPP object MUST implement
1372 the attribute (MANDATORY) or MAY implement the attribute (OPTIONAL).
- 1373 3. for each attribute in the collection value, whether the attribute's presence is required
1374 or optional.
- 1375 4. for each attribute permitted in the collection value, the completed specification of that
1376 attribute shall be included or inferred by reference to the specification of that attribute
1377 elsewhere, including its keyword name, its attribute syntax, including '1setOf', if it is
1378 multi-valued, and the semantics of the values.
- 1379 5. for each attribute defined in the collection, whether that attribute may also be used
1380 separately by itself. For example, in the "job-notify" example, could the "job-notify-
1381 events" and "job-notify-recipients" attributes occur by themselves in a create
1382 operation, say, when the client is only specifying a single collection or must they
1383 always occur within a collection value.

1384 A collection may contain another collection, i.e., may include an attribute whose attribute
1385 syntax is, itself, a 'collection', if the specification of the (outer) collection attribute allows.

1386 Additional attributes may be registered for use in a collection attribute.

1387 Implementers may support additional private attributes in a collection value.

1388 **ISSUE 26:** What should the maximum size of a collection value be? If it is much bigger
 1389 than the current maximum of 1023 octets, it may not be safely ignored by existing
 1390 parsers. Is 2047 octets sufficiently big, without being a problem to existing parsers?

1391 **14.6 Encoding**

1392 This section shows the encoding for the alternative of representing a collection as a new
 1393 attribute syntax. The following example is written in the style of the IPP/1.0 "Encoding
 1394 and Transport" (nee "Protocol") document.

Octets	Symbolic Value	Protocol field	comments
0x34	collection type	value-tag	"job-notify" attribute
0x000a		name-length	
job-notify	job-notify	name	
0x0062		value-length	98 octets in 1st dict value
0x45	uri type	value-tag	"job-notify-recipients" attribute
0x0011		name-length	
notify-recipients	notify-recipients	name	
0x0020		value-length	
ipp-tcp-ip-socket:port=700	ipp-tcp-ip-socket:port=700	value	
0x44	keyword type	value-tag	"job-notify-events" attribute
0x0013		name-length	
notify-event-groups	notify-event-groups	name	
0x0b		value-length	
job-errors	job-errors group	value	
0x44	keyword type	value-tag	start of 2nd job-notify-events value
0x0000		name-length	0 length means next multiple value
0x000e		value-length	
job-completion	job-completion	value	
0x34	collection-type	value-tag	start of 2nd collection value
0x0000		name-length	0 length mean next multiple value
0xn timer	0xn timer	value-length	n timer octets in 2nd dict value
0x45	uri type	value-tag	"job-notify-recipients" attribute

Octets	Symbolic Value	Protocol field	comments
0x0015		name-length	
job-notify- recipients	job-notify-recipients	name	
0x000c		value-length	
mailto:smit h	mailto:smith	value	
...			nnnn octets of the next dict value

1395

1396 **14.7 Rejected alternatives for a collection mechanism**

1397 This section lists the alternatives we considered for adding a new attribute syntax to
1398 represent a collection value.

- 1399 1. No maximum length for the new attribute syntax: 'collection'. If an IPP object
1400 supports collection it has to read a piece at a time. If it doesn't it has to be able to
1401 ignore an arbitrarily long data value. See the encoding example in the next section.

1402 Reason for rejection: Not completely compatible with current parsers that have a fixed
1403 buffer size for entities of around 1023 octets, the current IPP data type maximum.

- 1404 2. Have a 2047 octet max length, continueCollection as a second attribute syntax and
1405 endCollection so that dictionaries can nest.

1406 Reason for rejection: More complexity.

- 1407 3. Have a 2047 octet max length but allow repeated instances of an attribute to append
1408 additional collection values.

1409 Reason for rejection: Not the current procedure for duplicate attributes; the IPP Object is
1410 to return an error.

- 1411 4. Add a new group tag to represent a collection value somehow. Groups do NOT have
1412 lengths and existing parsers are supposed to ignore group tags they don't understand.

1413 Reason for rejection: Not completely compatible with existing parsers.

- 1414 5. Add an out-of-band value that indicates that this attribute was the beginning of a
1415 collection and add an attribute that marked the end of the collection value.

1416 Reason for rejection: Not completely compatible with existing parsers. Existing parser
1417 would try to interpret the contents of the collection as regular attributes.

- 1418 6. Extend the attribute naming mechanism to include a collection name and a collection
1419 index for use with multi-valued dictionaries. Use the colon (":") to separate
1420 component names. Thus if foo is a set of dictionaries, then "foo:1:x" is the name that
1421 accesses field x of the 2nd collection of attribute foo (indexing is 0 based). Leaving
1422 off the syntax after either colon, is interpreted as a wild card meaning all values with
1423 the prefix up to the colon.

- 1424 Reason for rejection: Changing the naming more of a change than is necessary with the
1425 current 1setOf 1setOf proposal, which does not change the naming and does not add an
1426 attribute syntax.
- 1427 7. Add a numeric instance number to the end of parallel attributes, i.e., "job-notify-
1428 method-supported-1".
- 1429 Reason for rejection: Not needed to be able to address a particular instance of a parallel
1430 attribute value.
- 1431 8. Use the semantics of parallel multi-valued attributes that we have in IPP/1.0, such as
1432 we already have for the "printer-uri-supported" and "uri-security-supported" Printer
1433 attributes, in order to achieve the effect of multi-valued dictionaries containing single
1434 values attributes. In order to represent the effect of a collection which contains
1435 attributes that are multi-valued, we only need to introduce the model semantics of:
1436 1setOf 1setOf X as an attribute syntax.
- 1437 Reason for rejection: Implementation with DPA parallel attributes has shown that it is
1438 too difficult for clients and servers to deal with parallel values. Its much better if the
1439 values in a collection value are all bound together. Also what if the number of values
1440 isn't the same?
- 1441 9. Calling the new data type a 'dictionary'. Instead, we chose 'collection', since the name
1442 dictionary implies some sort of sorting or ordering.
- 1443