

1 INTERNET-DRAFT **There are 6 ISSUES highlighted like this.**
2 <draft-ietf-ipp-indp-00.txt>

Hugo Parra
Novell, Inc.
Tom Hastings
Xerox Corporation
February 29, 2000

8 Internet Printing Protocol/1.1: **IPP Notification Delivery Protocol**

9 Copyright (C) The Internet Society (2000). All Rights Reserved.

10 Status of this Memo

11 This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of [rfc2026].
12 Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its
13 working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

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

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

19 **Abstract**

20 The IPP event notification specification [ipp-ntfy] is an OPTIONAL extension to IPP/1.0 and IPP/1.1. [ipp-
21 ntfy] which enables IPP clients to request notification of printer and job events. The IPP notification
22 extension gives IPP Printers the flexibility to choose how many Subscriptions objects (individual requests for
23 notification), what delivery methods, and what natural languages to support, among others. In practice, it's
24 the working environment where an IPP Printer is deployed what ultimately dictates the notification
25 requirements for that printer. Notification Delivery Services exist to help event producers, such as IPP
26 Printers, meet the varying notification needs of disparate environments. Specifically, an IPP Notification
27 Delivery Service may extend the notification capabilities of IPP Printers and help customize the type of
28 notification required in a highly specialized environment. This documents defines the IPP Notification
29 Delivery Protocol (INDP), a protocol for IPP Printers to communicate with Notification Delivery Services
30 using "application/ipp" as the encoding mechanism and HTTP as the transport. The definition of INDP
31 lends itself nicely for use by IPP Printers and Notification Delivery Services for dispatching IPP Notifications
32 to Notification Recipients as well.

33 The full set of IPP documents includes:

34 Design Goals for an Internet Printing Protocol [RFC2567]

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

36 Internet Printing Protocol/1.1: Model and Semantics (this document)

37 Internet Printing Protocol/1.1: Encoding and Transport [ipp-pro]

38 Internet Printing Protocol/1.1: Implementer's Guide [ipp-iig]

39 Mapping between LPD and IPP Protocols [RFC2569]

40

41 The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed printing
42 functionality, and it enumerates real-life scenarios that help to clarify the features that need to be included in
43 a printing protocol for the Internet. It identifies requirements for three types of users: end users, operators,
44 and administrators. It calls out a subset of end user requirements that are satisfied in IPP/1.0. A few
45 OPTIONAL operator operations have been added to IPP/1.1.

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

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

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

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

61

62

Table of Contents

63 1 Introduction..... 6

64 2 Terminology 6

65 3 Model and Operation 6

66 3.1 NOTIFICATION DELIVERY SERVICE MODEL 7

67 3.1.1 *Server Object*..... 7

68 3.1.2 *Subscription Object* 7

69 3.2 NOTIFICATION DELIVERY SERVICE OPERATION 8

70 3.2.1 *Notification without Notification Delivery Services* 9

71 3.2.2 *Delivery method support extension (INDPa)*..... 10

72 3.2.3 *Natural language support extension (INDPb)* 10

73 3.2.4 *Subscription object management outsource (INDPc)* 11

74 4 Notification Operations..... 13

75 4.1 GET-NOTIFY-SERVICE-ATTRIBUTES 13
 76 4.1.1 *Get-Notify-Service-Attributes Request* 13
 77 4.1.2 *Get-Notify-Service-Attributes Response* 14
 78 4.2 VALIDATE-NOTIFY-TARGET-URI OPERATION 14
 79 4.2.1 *Validate-Notify-Target-Uri Request* 14
 80 4.2.2 *Validate-Notify-Target-Uri Response* 15
 81 4.3 SEND-NOTIFICATIONS OPERATION 15
 82 4.3.1 *Send-Notifications Request* 15
 83 4.3.2 *Send-Notifications Response* 17
 84 4.4 REGISTER-NOTIFICATION-SOURCE OPERATION 17
 85 4.4.1 *Register-Notification-Source Request* 17
 86 4.4.2 *Register-Notification-Source Response* 18
 87 4.5 CANCEL-NOTIFICATION-SOURCE-REGISTRATION OPERATION 19
 88 4.5.1 *Cancel-Notification-Source-Registration Request* 19
 89 4.5.2 *Cancel-Notification-Source-Registration Response* 19
 90 4.6 RENEW-NOTIFICATION-SOURCE-REGISTRATION OPERATION 20
 91 4.6.1 *Renew-Notification-Source-Registration Request* 20
 92 4.6.2 *Renew-Notification-Source-Registration Response* 20
 93 4.7 CREATE-SUBSCRIPTION OPERATION 21
 94 4.7.1 *Create-Subscription Request* 21
 95 4.7.2 *Create-Subscription Response* 21
 96 4.8 VALIDATE-SUBSCRIPTION OPERATION 22
 97 4.8.1 *Validate-Subscription Request* 22
 98 4.8.2 *Validate-Subscription Response* 22
 99 4.9 CANCEL-SUBSCRIPTION OPERATION 22
 100 4.9.1 *Cancel-Subscription Request* 22
 101 4.9.2 *Cancel-Subscription Response* 23
 102 4.10 RENEW-SUBSCRIPTION OPERATION 23
 103 4.10.1 *Renew-Subscription Request* 23
 104 4.10.2 *Renew-Subscription Response* 24
 105 4.11 GET-SUBSCRIPTIONS OPERATION 24
 106 4.11.1 *Get-Subscriptions Request* 24
 107 4.11.2 *Get Subscriptions Response* 25
 108 5 Encoding of the Operation Layer 25
 109 5.1 NEW ATTRIBUTE TAG 25
 110 5.2 NEW STATUS CODES 25
 111 5.2.1 *unknown-notification-recipient (0xXXX)* 25
 112 5.2.2 *unable-to-delivery-notification-report (0xXXX)* 25
 113 5.2.3 *successful-ok-but-cancel-subscription (0xXXXX)* 25
 114 5.2.4 *unknown-registration-id (0xXXX)* 26
 115 5.2.5 *successful-ok-but-error-accessing-persistent-storage (0xXXXX)* 26
 116 5.3 ENCODING 26
 117 6 Encoding of Transport Layer 27
 118 7 IANA Considerations 28
 119 8 Internationalization Considerations 28
 120 9 Security Considerations 28

121 9.1 SECURITY CONFORMANCE..... 29

122 10 References..... 29

123 11 Author's Addresses..... 30

124 12 Full Copyright Statement..... 30

125

126

127 **1 Introduction**

128 IPP Printers that support the OPTIONAL IPP event notification extension [ipp-ntfy] either a) accept, store,
129 and use notification Subscriptions to generate notification reports and implement one or more delivery
130 methods for notifying interested parties, or b) support a subset of these tasks and farm out the remaining
131 tasks to a Notification Delivery Service. The IPP Notification Delivery Protocol (INDP) specified in this
132 document is a request/response protocol that may be used in a variety of notification scenarios. Its primary
133 intended use is for IPP Printers to engage the assistance of Notification Delivery Services for storing
134 notification Subscriptions, generating human-readable notifications in various languages, and implementing
135 additional delivery methods. Moreover, IPP Printers and Notification Delivery Services may use INDP to
136 send (push) event notifications to Notification Recipients.

137 **2 Terminology**

138 This document uses terms such as "attributes", "keywords", and "support". These terms have special
139 meaning and are defined in the model terminology [ipp-mod] section 12.2.

140 Capitalized terms, such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED
141 NOT, and OPTIONAL, have special meaning relating to conformance. These terms are defined in [ipp-
142 mod] section 12.1 on conformance terminology, most of which is taken from RFC 2119 [RFC2119].

143 This section defines the following additional terms that are used throughout this document:

144 **REQUIRED:** if an implementation supports the extensions described in this document, it **MUST** support
145 a **REQUIRED** feature.

146 **OPTIONAL:** if an implementation supports the extensions described in this document, it **MAY** support
147 an **OPTIONAL** feature.

148 Event Notification (Notification for short) - See [ip-ntfy]

149 Notification Source - See [ipp-ntfy]

150 Notification Recipient - See [ipp-ntfy]

151 Subscription object - See [ipp-ntfy]

152 Ultimate Notification Recipient - See [ipp-ntfy]

153 **3 Model and Operation**

154 In the IPP Notification Model [ipp-ntfy], print clients request an IPP Printer for event notification by causing
155 a Subscription object to be created at the printer. [ipp-ntfy] specifies a number of ways in which
156 Subscription objects can be created. Each Subscription object lists the events of interest, the delivery
157 method to be employed, and the address to which notifications should be dispatched, among others. When
158 an event occurs, the printer is responsible for notifying each Notification Recipient that has registered
159 interest in that event, using delivery method requested by that Notification Recipient. IPP Printers may
160 employ the assistance of Notification Delivery Services to accomplish some or all of these tasks.

161 IPP Printers with support for Notification Delivery Services must support a new printer description attribute,
162 “notification-delivery-services-uri-supported” (1SetOf uri). This attribute needs to be populated with the
163 uri’s of the Notification Delivery Services the printer is configured to use. Whether IPP Printers dynamically
164 discover Notification Delivery Services on the network or need to be configured by a system administrator it
165 implementation dependant.

166 **3.1 Notification Delivery Service Model**

167 The INDP 1.0 model defines objects of type Server and Subscription. Each object definition includes a set
168 of attributes that describe the state and workings of a Notification Delivery Service. An IPP Printer interact
169 with instances of these object types by issuing INDP operations. This section describes the attributes that
170 compose the Server and Subscription objects with their corresponding attribute syntaxes and values that are
171 part of the Notification Delivery Service Model. Each attribute is uniquely identified in this document using
172 a “keyword” as defined in the IPP/1.1: Model and Semantics document [ipp-mod]. INDP 1.0 defines The
173 Notification Delivery Service

174 **3.1.1 Server Object**

175 The Server object represents the state and capabilities of a Notification Delivery Service. It implements the
176 server-side of INDP. In version 1.0 of INDP, the Server object contains information about the capabilities
177 of a Notification Delivery Service that are of interest to an IPP Printer.

178 The following attributes comprise the Server object. Their description and intended use follow.

- 179 • notify-natural-languages-supported
- 180 • notify-uri-schemes-supported

181 **3.1.1.1 notify-natural-languages-supported (1setOf naturalLanguage)**

182 MANDATORY {To be added}

183 **3.1.1.2 notify-uri-schemes-supported (1setOf uri)**

184 MANDATORY {To be added}

185 **3.1.2 Subscription Object**

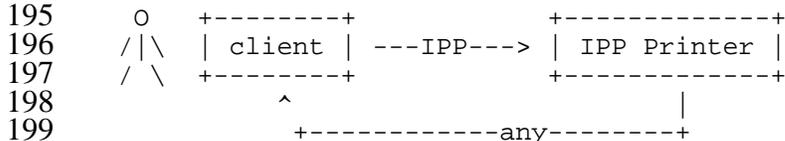
186 The Subscription object represents a request for notification. Subscription Objects are contained by a Server
187 object and are created as a result of an IPP Printer issuing a Create-Subscription operation. The syntax and
188 semantics of a Subscription object exactly mirror those of the Subscription object defined in the IPP
189 Notification spec [ipp-ntfy].

190 **3.2 Notification Delivery Service Operation**

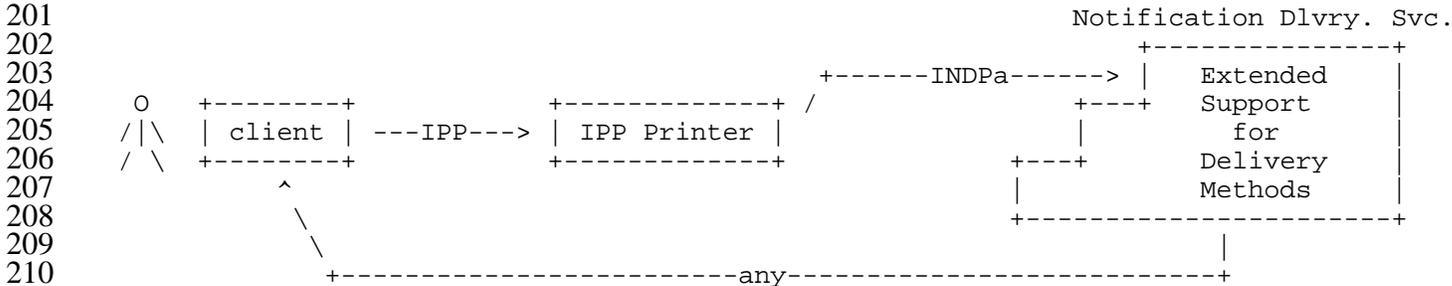
191 The figure below illustrates four different configurations through which an IPP Printer may implement
 192 support for IPP notification. Each configuration is discussed in this section.

193

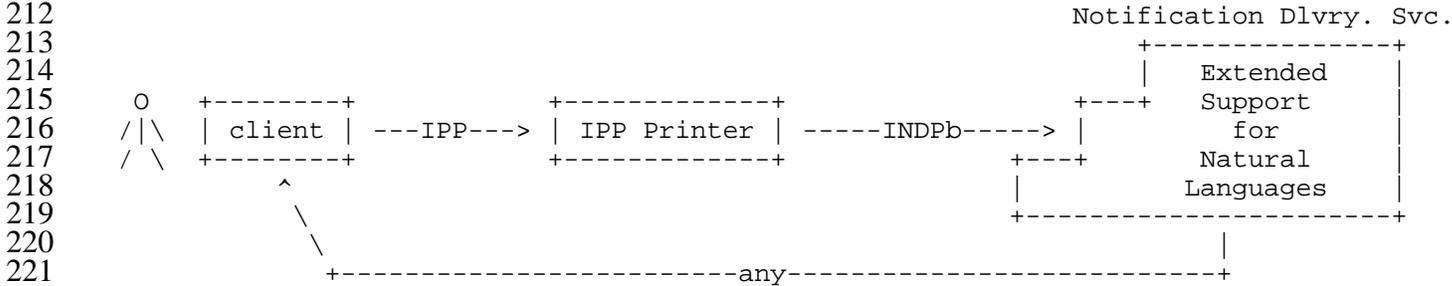
194



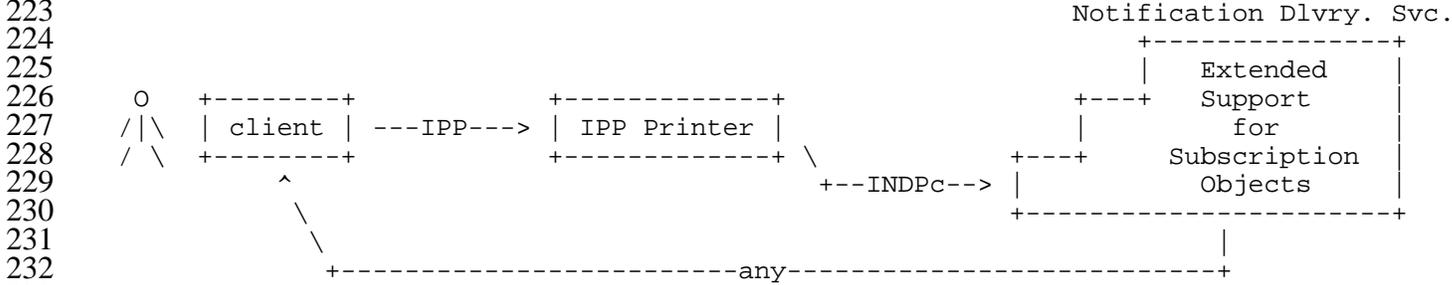
200



211



222



233

234 Legend:

235

236 INDPx represents three different subsets of INDP operations the IPP
 237 Printer uses to communicate with the Notification Delivery Service
 238 to realize three different levels of support.

239

240 any represents any protocol, including INDP, that the IPP Printer
 241 or the Notification Delivery Service may support for notifying interested notification
 242 Recipients.

243

244 **3.2.1 Notification without Notification Delivery Services**

245 An IPP Printer working without the assistance of a Notification Delivery Service must implement on its own
246 at least the minimum set of functionality required by the IPP Notification spec. This section gives a
247 summary of the process a typical IPP Printer may employ to support IPP notifications on its own. The IPP
248 Notification spec [ipp-ntfy] provides a detailed description of this process. Subsequent sections will describe
249 how an IPP Printer may use INDP to indirectly accomplish some of the following tasks.

- 250 a) Creating a Subscription object. The IPP notification spec [ipp-ntfy] describes a number of mechanisms
251 for IPP clients to request notification of an IPP printer. The end result, however, is that a Subscription
252 object is instantiated at the IPP printer containing the information needed by the printer to know who to
253 notify, how, and of what events.
- 254 b) Validating the Subscription object. At Subscription object instantiation time, the IPP printer validates its
255 contents to make sure the requested events and delivery methods are supported. The IPP printer may
256 also perform some validation on the recipient uri, requested natural language, and other information
257 contained in the Subscription object.
- 258 c) Storing the Subscription object. The IPP printer provides persistent and non-persistent storage for
259 Subscription objects until the object's lease expires (in the case of per-printer subscriptions) or their
260 associated print job is removed (in the case of per-job subscriptions). The IPP notification spec [ipp-
261 ntfy] outlines the minimum number of Subscription objects a printer MUST be able to store. In practice,
262 this requirement will vary widely depending on the administrative practices and usage patterns of the
263 printer's users.
- 264 d) Event condition. Normal printer operation as well as printer exception circumstances will cause event
265 conditions to be raised.
- 266 e) Matching event with subscriptions. For each raised event condition the printer finds all the Subscription
267 objects that request notification of that event. Rather than inspecting each Subscription object each time
268 an event condition is raised, an IPP Printer may keep a list of the events the combined Subscription
269 objects have requested to quickly discard event conditions no one is interested in.
- 270 f) Generating human-readable notification data. The IPP Printer examines each Subscription object found
271 in step (e) to determine if it needs to generate human-readable notification information for it. IPP
272 Printers with users of different language preferences may need to provide translation for multiple natural
273 languages.
- 274 g) Dispatching the notification via the specified delivery method. The IPP Printer may need to generate
275 slightly different Notification payloads for different delivery methods. With Notifications generated for
276 each target Recipient, the IPP Printer uses its implementation of the delivery method specified in the
277 corresponding Subscription object to dispatch the notification to its intended Recipient.

278 Though in this scenario the IPP Printer does not need to interact with a Notification Delivery Service, it may
279 use INDP to dispatch Notifications encoded in “application/ipp” and transported over HTTP to interested
280 notification Recipients. IPP Printers may use the Send-Notifications operation to accomplish this task.

281 **3.2.2 Delivery method support extension (INDPa)**

282 An IPP Printer may use a Notification Delivery Service simply to extend the list of delivery methods it
283 supports. Doing so offloads a printer from having to implement all the common delivery methods its
284 potential clients might require. It also enables a generic printer to support very specialized delivery methods
285 implemented by a site’s Notification Delivery Service. Moreover, by using existing Notification Delivery
286 Methods, an IPP Printer can take advantage of present, widely deployed notification infrastructure,
287 standards-based or proprietary.

288 Using a Notification Delivery Service for the sole purpose of extending the notification delivering
289 capabilities on and IPP Printer results in very small changes to the notification process described in the
290 previous section. Specifically, the following changes apply.

- 291 1) Before accepting requests to create Subscription objects, step (a) above, the IPP Printer gets a list of the
292 uri schemes the Notification Delivery Service supports and adds the values to its “notify-schemes-
293 supported” attribute. To obtain this list, the IPP Printer uses the Get-Notify-Service-Attributes
294 operation requesting the “notify-schemes-supported” attribute from the Notification Delivery Service.
295 To an IPP client reading the printer’s “notify-schemes-supported” attribute, the entries with internal
296 support and those supported via a Notification Delivery Service are indistinguishable.
- 297 2) During Subscription object validation, step (b) above, the IPP Printer may communicate with the
298 Notification Delivery Service to validate a target uri requesting a delivery method implemented in the
299 Notification Delivery Service. This IPP Printer accomplishes through the Validate-Notification-Uri
300 operation.
- 301 3) For dispatching notifications that require a delivery method implemented in the Notification Delivery
302 Service, step (g) above, the IPP Printer forwards the Notification on to the Notification Delivery Service
303 through the Send-Notifications operation. The IPP Printer must provide the target uri and human-
304 readable data, when the case requires it. The Notification Delivery Service is then responsible for
305 creating a Notification payload suitable for the requested delivery method and for dispatching the
306 notification to the specified Recipient.

307 **3.2.3 Natural language support extension (INDPb)**

308 An IPP Printer may use a Notification Delivery Service to generate human-readable notification data in
309 addition to extending its delivery methods support. By using a Notification Delivery Service in this manner,
310 an IPP Printer can dynamically support notifications in any number of natural languages, as long as the
311 Notification Delivery Service being used supports them.

312 In addition to the modifications to the notification process listed in section 3.2, the following changes result
313 from using a Notification Delivery Service to generate human-readable notification data.

314 1) Before accepting requests to create Subscription objects, step (a) above, the IPP Printer must
315 communicate with the Notification Delivery Service to get a list of the natural languages it supports for
316 human-readable message generation and add these values to its own “notify-natural-languages-
317 supported” attribute.

318 **ISSUE 01: Do we have any use for the printer description attribute “notify-natural-languages-supported”?**

319 2) The IPP Printer no longer needs to perform steps (f) and (g) above. Instead it uses the Send-
320 Notifications operation to send the Notification to the Notification Delivery Service along with the
321 language specified in the corresponding Subscription object.

322 **3.2.4 Subscription object management outsource (INDPc)**

323 Through INDP an IPP Printer can employ the full services of a Notification Delivery Service, which includes
324 storing and managing Subscription objects on behalf of the printer. Outsourcing this type of functionality
325 greatly reduces the logic and resources requirements for an IPP Printer to support notification. Suitably
326 hosted Notification Delivery Services can meet the notification needs of an environment without having to
327 increase the capabilities of each printer in that environment. This section describes how an IPP Printer
328 interacts with a Notification Delivery Service to accomplish this level of interaction.

329 This notification configuration requires the IPP Printer to establish a temporary registration with the
330 Notification Delivery Service. Through a lease-based relationship, the Notification Delivery Service can
331 keep track of what Subscription objects belong to what IPP Printer and generate the appropriate
332 notifications when events are reported. This mechanism also enables the Notification Delivery Service to
333 clean up orphaned Subscription objects. The IPP Printer uses the Register-Event-Producer operation to
334 establish this type of relationship with the Notification Delivery Service. The model requires that an IPP
335 Printer renew its lease periodically using the Renew-Registration operation.

336 When registering, an IPP Printer can specify a location for a Notification Delivery Service to store
337 Subscription objects persistently. Subscription objects stored persistently in previous registrations are
338 automatically re-instantiated when an IPP Printer registers with a Notification Delivery Service. The printer
339 instructs the Notification Delivery Service what Subscription objects should be stored persistently and which
340 one should be automatically disposed when the registration expires.

341 Once registered, the IPP Printer may forward requests to create Subscription objects on to the Notification
342 Delivery Service. The IPP Printer uses the Create-Subscription operation to accomplish this task.

343 In this notification configuration an IPP Printer only needs to keep track of the superset of events requested
344 by all the Subscription objects combined. The Notification Delivery Service assists the IPP Printer
345 accomplish this task. First, in the response of a successful registration request, the Notification Delivery
346 Service returns to the printer the list of events that it must generate to satisfy any Subscription objects that
347 might have been reinstated from persistent storage. Then, in the response to every successful request to add

348 or delete Subscription objects, the Notification Delivery Service returns to the printer a list of the new events
349 needed and those to be discontinued as a result of the operation.

350 The following summarizes an IPP Printer's process for handling notification when making full use of a
351 Notification Delivery Service's capabilities. For simplification, the description assumes that the IPP Printer
352 supports these capabilities only via a Notification Delivery Service and not directly. However, for printers
353 that implement some delivery methods internally and support others through a Notification Delivery Service,
354 the notification process is a combination of the process outlined below and the one summarized in section
355 3.1.1.

- 356 a) Register with Notification Delivery Service. Early in its initialization process the IPP Printer should use
357 the Register-Event-Producer operation to register with a Notification Delivery Service if configured to
358 do so. It must indicate to the Notification Delivery Service the location of its persistent Subscription
359 object storage, if applicable. The IPP Printer must store away the registration Id returned by the
360 operation and remember any events listed in the response so it can start generating them.
- 361 b) Get Notification Delivery Service information. Right after registering with a Notification Delivery
362 Service, the IPP Printer should query the Notification Delivery Service's "notify-uri-schemes-supported"
363 and "notify-natural-languages-supported" attributes. The printer must populate its "notify-uri-schemes-
364 supported" and "notify-natural-languages-supported" attributes with the information obtained.
- 365 c) Create Subscription objects. When the IPP Printer receives a client request to create a new Subscription
366 object, it must forward the request to the Notification Delivery Service using the Create-Subscription
367 operation. This results in the Notification Delivery Service instantiating and validating a Subscription
368 object. If the operation to create a new Subscription object succeeds, its response portion will tell the
369 IPP Printer what, if any, new events it must generate to satisfy the new request. As with print jobs
370 Subscription objects do not become active while the job is in "job-pending" state, the IPP Printer would
371 not send a request to create a new Subscription object to the Notification Delivery Service until just
372 before the job changes states from "job-pending". For these types of notification requests, the IPP
373 Printer may instead issue the Validate-Subscription operation to request that the Notification Delivery
374 Service simply validate the request, thus allowing the printer to return an accurate status code to IPP
375 operations requesting per-job notifications.
- 376 d) Event condition. The IPP Printer uses the consolidated list of events it maintains with the help of the
377 Notification Delivery Service to know what events are of interest.
- 378 e) Send event report. When the IPP Printer raises an event condition, it reports the event to the
379 Notification Delivery Service using the Send-Notification operation. At that point the IPP Printer is
380 finished processing the event condition. The Notification Delivery Service is responsible for matching
381 the event with the Subscription objects that requested it, generating any human-consumable data in the
382 natural language specified in the Subscription object, and dispatching the appropriately formatted
383 Notification using the requested delivery method.

384 4 Notification Operations

385 INDP makes extensive use of the operations model defined by IPP [rfc2566]. This includes, the use of a URI
 386 as the identifier for the target of each operation, the inclusion of a version number, operation-id, and
 387 request-id in each request, and the definition of attribute groups. INDP operations use the Operation
 388 Attributes group, but currently have no need for the Unsupported Attributes, Printer Object Attributes, and
 389 Job-Object Attributes groups. However, it uses a new attribute group, the Notification Attributes group.

390 The following operations form version 1.0 of INDP. All operations are targeted at the Server object. This
 391 section formally defines each INDP 1.0 operation.

- 392 • Get-Notify-Service-Attributes
- 393 • Validate-Notify-Target-Uri,
- 394 • Send-Notifications
- 395 • Register-Notification-Source
- 396 • Cancel-Notification-Source-Registration
- 397 • Renew-Notification-Source-Registration
- 398 • Create-Subscription
- 399 • Validate-Subscription
- 400 • Cancel-Subscription
- 401 • Renew-Subscription
- 402 • Get-Subscriptions
- 403

404 4.1 Get-Notify-Service-Attributes

405 This REQUIRED operation allows an IPP Printer to request the values of attributes of a Server object. In
 406 the request, the IPP Printer supplies the set of Server attribute names it's interested in. In the response, the
 407 Service object returns a corresponding attribute set with the appropriate attribute values filled in.

408 4.1.1 Get-Notify-Service-Attributes Request

409 The following sets of attributes are part of the Get-Service-Attributes Request:

410 Group 1: Operation Attributes

411

412 Natural Language and Character Set:

413 The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566]
 414 section 3.1.4.1.

415

416 "server-uri":

417 The URI of the Notification Delivery Service.

418

419 "requested attributes" (1setOf keyword):

420 The IPP Printer OPTIONALLY supplies a set of attribute names in whose values the
 421 requester is interested. The Service object MUST support this attribute. If the IPP Printer
 422 omits this attribute, the Notification Delivery Service MUST respond with a list of all the
 423 attributes it supports and its respective values.

424 **4.1.2 Get-Notify-Service-Attributes Response**

425 The Server object returns the following sets of attributes as part of the Get-Notify-Service-Attributes
 426 Response:

427 Group 1: Operation Attributes

428 Natural Language and Character Set:

429 The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566]
 430 section 3.1.4.1.

431

432 Group 2: Unsupported Attributes

433 A list of the attribute names requested by the IPP Printer but not supported by the Service object.
 434 See [rfc 2566] section 3.1.7 for details on returning Unsupported Attributes. As in version 1.0 of
 435 INDP all defined Service object attributes are mandatory, this group is a forward-looking feature
 436 when new OPTIONAL attributes may be defined.

437

438 Group 3: Server Object Attributes

439 This is the set of requested attributes and their current values. The Server object ignores any
 440 requested attribute that is not supported. The Service object MAY respond with a subset of the
 441 supported attribute and value, depending on the security policy in force. However, the Service
 442 object MUST respond with the 'unknown' value for any supported attribute for which the Service
 443 object does not know the value. For a description of "out-of-band" values see [rfc 2566] section
 444 4.1.

445 **4.2 Validate-Notify-Target-Uri Operation**

446 This REQUIRED operation allows an IPP Printer to request that the Notification Delivery Service validate a
 447 notification target uri. The Service object successfully validates the uri if the Notification Delivery Service
 448 implements the delivery method implied by the uri scheme or the target uri. The Service object is free to
 449 perform extended analysis on the validity of the recipient's address provided in the uri is the semantics of the
 450 delivery method so allow.

451 **4.2.1 Validate-Notify-Target-Uri Request**

452 The following sets of attributes are part of the Validate-Notify-Target-Uri Request:

453 Group 1: Operation Attributes

454

455 Natural Language and Character Set:

456 The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566]
457 section 3.1.4.1.

458
459 “server-uri”:
460 The URI of the Notification Delivery Service.

461
462 “notify-target-uri” (uri):
463 The IPP Printer MUST supply this attribute. The Notification Delivery Service MUST
464 support this attribute. It is the uri to be validated by the Server object.

465 4.2.2 Validate-Notify-Target-Uri Response

466 The Server object returns the following set of attributes as part of the Validate-Notify-Target-Uri Response:

467 Group 1: Operation Attributes

468 Natural Language and Character Set:
469 The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566]
470 section 3.1.4.1.

471
472 “validation-code” (boolean):
473 The Server object MUST return this attribute with a value of TRUE if the notify-target-uri
474 was validates successfully; FALSE otherwise.

475 4.3 Send-Notifications Operation

476 This REQUIRED operation allows an IPP Printer to send one or more Notifications to a Notification
477 Delivery Service. The Send-Notification operation can be used to transport Notification data in all four
478 notification configurations described in section 3.2. Different attributes will be required depending on
479 whether the operation is being used a) by an IPP Printer or Notification Delivery Service to send
480 Notifications directly to a notification Recipient, b) by an IPP Printer to Send a localized Notification to a
481 Notification Delivery Service (INDPa), c) by an IPP Printer to Send a Notification to be localized and
482 dispatched by the Notification Delivery Service (INDPb), or d) by an IPP Printer to send a target-less
483 notification using an established registration to a Notification Delivery Service (INDPc).

484 Both Machine-Consumable and Human-Consumable notifications may be included in the Send-Notification
485 operation.

486 4.3.1 Send-Notifications Request

487 The following groups of attributes are part of the Send-Notifications Request:

488 Group 1: Operation Attributes

489
490 Natural Language and Character Set:

491 The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566]
 492 section 3.1.4.1.

493

494

Target:

495

496

497

498

499

The Target can a) The URI of the Notification Delivery Service if an IPP Printer is using Send-Notifications to dispatch notifications, or b) the URI of the Notification Recipient if the IPP Printer or the Notification Delivery Service are using the operation to dispatch notifications directly to a Notification Recipient.

500

“ultimate-target-uri”:

501

502

503

504

This attribute MUST be supplied by the IPP Printer when it uses the Send-Notifications operation to send notifications to a Notification Delivery Service without having registered as a Notification Source, i.e., configurations INDPa and INDPb above.

505

“registration-id”:

506

507

508

509

This attribute MUST be supplied by the IPP Printer when it uses the Send-Notifications operation to send notifications to a Notification Delivery Service after having registered a as a Notification Source, i.e., configuration INDPc above.

510

Group 2 to N: Notification Attributes

511

"human-readable-report" (text)

512

513

514

515

516

517

518

519

520

521

522

The Notification Source OPTIONALLY supports this attribute. This attribute is a text string generated by the IPP printer or Notification Delivery Service from the contents of the IPP Notification suitable for human consumption. If the Notification Source supports this attribute, it MUST supply this attribute if the Subscription object contains the "notify-text-format" (mimeType) attribute. The text value of this attribute MUST be localized in the charset identified by the "notify-charset" (charset) attribute and the natural language identified by the "notify-natural-language" (naturalLanguage) attribute supplied in the associated Subscription object that generates this event Notification. The format of the text value is specified by the value of the "notify-text-format" (mimeType) supplied in the associated Subscription object.

523

“human-readable-report-format” (mime)

524

525

526

527

This attribute MUST be supplied by the Notification Source whenever the “human-readable-report” attribute is present. It indicates the format, e.g., text/plain, text/html, etc. of the “human-readable-report” attribute.

528 All of the REQUIRED attributes and any of the OPTIONAL attributes indicated in [ipp-ntfy] for a
 529 Push event Notification, including "notify-text-format-type" (mimeType), if the "human-
 530 readable-report" (text) attribute is included, so that the Notification Recipient will know the text
 531 format of the "human-readable-report" (text) attribute value. These attributes communicate the same
 532 information as the notification attributes by the same name described in sections 7.4, 7.5, and 7.6 of
 533 [ipp-ntfy].

534
 535 The rules that govern when each individual attribute MUST or MAY be included in this operation
 536 precisely mirror those specified in [ipp-ntfy] with the following exception: if the Send-Notifications
 537 operation is being used by an IPP Printer to communicate events to a Notification Delivery Service
 538 using a "registration-id", Group 2 of this operation MUST only include the "trigger-event", "trigger-
 539 time", and "trigger-date-time" Notification attributes.

540 **4.3.2 Send-Notifications Response**

541 The target of the Send-Notifications operation, Notification Delivery Method or Notification Recipient,
 542 returns a status code for the entire operation and one for each Notification Report in the request if the
 543 operation's status code is other than "success-ok". If the Notification Recipient receives a Notification
 544 report that it can't pair up with a subscription it knows about, it can return an error status-code to indicate
 545 that events associated with that subscription should no longer be sent to it.

546 Group 1: Operation Attributes

547 Natural Language and Character Set:

548 The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566]
 549 section 3.1.4.1.

550

551 Group 2 to N: Notification Attributes

552 "notification-report-status-code" (type2 enum)

553 Indicates whether the intended target, i.e., Notification Delivery Service or Notification
 554 Recipient was able to consume the n-th Notification Report.

555 **4.4 Register-Notification-Source Operation**

556 This REQUIRED operation allows an IPP Printer to register itself as a Notification Source with a
 557 Notification Delivery Service. While registered, the Printer can add Subscription objects to the Server
 558 object. The Printer can then send Notifications to the Server object for the Server object to dispatch
 559 Notifications to all interested Recipients.

560 **4.4.1 Register-Notification-Source Request**

561 The following sets of attributes are part of the Register-Notification-Source Request:

562 Group 1: Operation Attributes

563

564 Natural Language and Character Set:
565 The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566]
566 section 3.1.4.1.
567
568 “server-uri”:
569 The URI of the Notification Delivery Service.
570
571 “registration-lease-time-requested” (integer(0:86,400)):
572 This REQUIRED attribute specifies the time in the future when the IPP Printer would like its
573 registration lease to expire. When the Server object accepts a Registration request, it keeps
574 track of this information. When the expiration time arrives, the Server object purges the
575 registration.
576
577 An IPP Printer is able to extend its registration lease using the Renew-Notification-Source-
578 Registration operation. The maximum value for a registration lease is one day.
579
580 “notification-source-name” (name(127)):
581 This REQUIRED attribute specifies the name of the IPP Printer. The Server object may use
582 this information to organize current registrations. This information may also be useful to a
583 Notification Delivery Service’s manager. Note: Management of a Notification Delivery
584 Service is outside the scope of INDP v1.0.
585
586 “persistent-registration-storage-uri” (uri):
587 Through this OPTIONAL attribute an IPP Printer may communicate to the Service object
588 where to retrieve persistent Subscriptions from previous registrations. The Service object
589 also uses this location to store away future persistent Subscriptions. It the IPP Printer
590 doesn’t provide this attribute, it will not be able to add Subscription objects that require
591 persistent storage.

592 4.4.2 Register-Notification-Source Response

593 The Server object returns the following set of attributes as part of the Register-Notification-Source
594 Response:

595 Group 1: Operation Attributes

596 Natural Language and Character Set:
597 The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566]
598 section 3.1.4.1.
599
600 “registration-id” (integer(0:MAX)):
601 The Server object MUST return the registration ID that the IPP Printer can use in subsequent
602 calls such as Renew-Notification-Source-Registration, Create-Subscription, etc.
603

604 “notify-events” (1setOf type2 keyword):
605 If in this operation’s request the IPP Printer specifies a “persistent-registration-storage-uri”
606 and as a result one or more Registrations are instantiated by the Server object during
607 registrations, this attribute MUST contain the list of events that the printer must notify the
608 Server object of to satisfy those Subscriptions.
609

610 “registration-lease-expiration-time” (integer(0:86,400)):
611 This REQUIRED attribute specifies the time in the future when the registration lease will
612 expire. If the Server object is not able to grant the lease-time requested by the IPP Printer,
613 this attribute may contain a different value than the one provided in the request.
614

615 An IPP Printer is able to extend its registration lease using the Renew-Notification-Source-
616 Registration operation. The maximum value for a registration lease is one day.

617 **4.5 Cancel-Notification-Source-Registration Operation**

618 This REQUIRED operation allows an IPP Printer to terminate a current registration to a Notification
619 Delivery Service. This causes the Server object to save all current persistent Subscriptions into the location
620 specified for this purpose at registration time, if one was specified. The Server object then cleans up any
621 data and processes associated with that registration. Notification Delivery Service implementations should
622 consider periodically saving away persistent Subscription objects to reduce the risk of failing to save
623 everything at deregistration time.

624 **4.5.1 Cancel-Notification-Source-Registration Request**

625 The following set of attributes is part of the Cancel-Notification-Source-Registration Request:

626 Group 1: Operation Attributes

627
628 Natural Language and Character Set:
629 The "attributes-charset" and "attributes-natural-language" attributes are defined in [rfc 2566]
630 section 3.1.4.1.
631

632 “server-uri”:
633 The URI of the Notification Delivery Service.
634

635 “registration-id” (integer(0:MAX)):
636 The IPP Printer MUST specify this REQUIRED attribute using the registration-id it obtained
637 from the Server object via the Register-Notification-Source operation.

638 **4.5.2 Cancel-Notification-Source-Registration Response**

639 The Server object returns the following set of attributes as part of the Cancel-Notification-Source-
640 Registration Response:

641 Group 1: Operation Attributes

642 Natural Language and Character Set:

643 The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566]
644 section 3.1.4.1.

645

646 “notify-events” (1setOf type2 keyword):

647 The Server object MUST return in this attribute the list of events that the printer must
648 discontinue as a result of ending its registration to the Notification Delivery Service. This
649 feature may be useful to IPP Printers that implement some delivery methods internally and
650 others via a Notification Delivery Service and those who may use more than one Notification
651 Delivery Service simultaneously.

652 4.6 Renew-Notification-Source-Registration Operation

653 This REQUIRED operation allows an IPP Printer to renew its lease on an existing registration to a
654 Notification Delivery Service. It MUST be issued before the lease-time specified in the Register-
655 Notification-Source operation or the previous Renew-Notification-Source-Registration operation expires.

656 4.6.1 Renew-Notification-Source-Registration Request

657 The following set of attributes is part of the Renew-Notification-Source-Registration Request:

658 Group 1: Operation Attributes

659

660 Natural Language and Character Set:

661 The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566]
662 section 3.1.4.1.

663

664 “server-uri”:

665 The URI of the Notification Delivery Service.

666

667 “registration-id” (integer(0:MAX)):

668 The IPP Printer MUST specify this REQUIRED attribute using the registration-id it obtained
669 from the Server object via the Register-Notification-Source operation.

670 “registration-lease-time-requested” (integer(0:86,400)):

671 This REQUIRED attribute specifies the time in the future when the IPP Printer would like
672 the registration lease to expire.

673 4.6.2 Renew-Notification-Source-Registration Response

674 The Server object returns the following set of attributes as part of the Renew-Notification-Source-
675 Registration Response:

676 Group 1: Operation Attributes

677 Natural Language and Character Set:
 678 The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566]
 679 section 3.1.4.1.

680 “registration-lease-expiration-time” (integer(0:86,400)):
 681 This REQUIRED attribute specifies the time in the future when the registration lease will
 682 expire. If the Server object is not able to grant the lease-time requested by the IPP Printer,
 683 this attribute may contain a different value than the one provided in the request.

684 4.7 Create-Subscription Operation

685 This REQUIRED operation allows an IPP Printer to cause a Subscription object to be instantiated in a
 686 Server object to which it is currently registered as a Notification Source. The Server object is responsible
 687 for keeping track of all registrations until their corresponding IPP Printer removes them via the Cancel-
 688 Subscription operation or until the registration is terminated by the Printer or it expires. The Server object
 689 uses Subscription object to know who and how to notify when it receives Notifications specifying a
 690 registration-id.

691 4.7.1 Create-Subscription Request

692 The Request for this operation includes the union of all of the REQUIRED attributes and any of the
 693 OPTIONAL attributes indicated in [ipp-ntfy] for the Create-Job-Subscription and Create-Printer-
 694 Subscription operations, with the following changes:

- 695
- 696 a) The “printer-uri” operational attribute is replaced by “server-uri” and MUST contain the URI of the
 697 Notification Delivery Service.
 - 698 b) The request MUST include the operational attribute “registration-id” (integer(0:MAX)) specifying the
 699 registration-id the IPP Printer obtained from the Server object via the Register-Notification-Source
 700 operation.

701

702 The rules that govern when each individual attribute MUST or MAY be included in this operation precisely
 703 mirror those specified in [ipp-ntfy] for the Create-Job-Subscription and Create-Printer-Subscription
 704 operations, but obviously not simultaneously. If the request contains a “job-id” the Server object enforces
 705 applies the validation rules defined for the Create-Job-Subscription operation. If the “job-id” is not present,
 706 the Server object enforces the validation rules defined for the Create-Printer-Subscription operation.

707 4.7.2 Create-Subscription Response

708 The Response for this operation is defined to be identical to the Response for the Create-Printer-
 709 Subscription operation as specified in [ipp-ntfy] except for the following changes:

- 710 a) The Response MUST include the operational attribute “notify-events” (1setOf type2 keyword)
 711 containing the list of events that the printer must notify the Server object of to satisfy the creation of the
 712 new Subscription object.
- 713 b) The “notify-printer-up-time” operational attribute ... ???

714 ISSUE 02: What should be done with this attribute. Should it be called the “notify-server-up-time” and be
715 populated with the Notification Delivery Server’s up time, or should it be filled in by the printer? There are
716 other ramifications here.

717 c) The Response does not include the “Unsupported Attribute” Group.

718 The Response that results from creating a job-related Subscription object doesn’t include the “notify-lease-
719 expiration-time” and “notify-server-up-time” attributes.

720 4.8 Validate-Subscription Operation

721 This REQUIRED operation allows an IPP Printer to request the Sever object to validate the contents of
722 what could become a Subscription object without actually creating the object. It employs the same logic
723 used by the Create-Subscription operation to validate a request.

724 4.8.1 Validate-Subscription Request

725 The Request for this operation is identical to the Create-Subscription operation Request.

726 4.8.2 Validate-Subscription Response

727 The Server object returns the following set of attributes as part of the Validate-Subscription Registration
728 Response:

729 Group 1: Operation Attributes

730 Natural Language and Character Set:

731 The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566]
732 section 3.1.4.1.

733 4.9 Cancel-Subscription Operation

734 This REQUIRED operation allows an IPP Printer to cause the Server object to cancel a Subscription object
735 currently associated with a given registration-id.

736 4.9.1 Cancel-Subscription Request

737 The following set of attributes is part of the Cancel-Subscription Request:

738 Group 1: Operation Attributes

739

740 Natural Language and Character Set:

741 The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566]
742 section 3.1.4.1.

743

- 744 “server-uri”:
745 The URI of the Notification Delivery Service.
746
- 747 “registration-id” (integer(0:MAX)):
748 The IPP Printer MUST specify this REQUIRED attribute using the registration-id it obtained
749 from the Server object via the Register-Notification-Source operation.
- 750 “subscription-id” (integer(0:MAX)):
751 This REQUIRED attribute specifies the ID of the Subscription object to be cancelled. The
752 IPP Printer must provide here the same “subscription-id” that it received back from the
753 Create-Subscription or Get-Subscriptions operations.

754 4.9.2 Cancel-Subscription Response

755 The Server object returns the following set of attributes as part of the Cancel-Subscription Response:

756 Group 1: Operation Attributes

- 757 Natural Language and Character Set:
758 The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566]
759 section 3.1.4.1.
760
- 761 “notify-events” (1setOf type2 keyword):
762 The Server object MUST return in this attribute the list of events that the printer must
763 discontinue as a result of canceling the Subscription object.

764 4.10 Renew-Subscription Operation

765 The REQUIRED Renew-Subscription operation permits an IPP Printer to request the Server object to
766 extend the lease on a Subscription object instance. This operation is only valid for Subscription object that
767 don't specify a “job-id”, or Per-Printer Subscription objects as they are referred to in [ipp-ntfy].

768 4.10.1 Renew-Subscription Request

769 The following set of attributes is part of the Renew-Subscription Request:

770 Group 1: Operation Attributes

- 771
- 772 Natural Language and Character Set:
773 The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566]
774 section 3.1.4.1.
775
- 776 “server-uri”:
777 The URI of the Notification Delivery Service.
778

779 “registration-id” (integer(0:MAX)):
780 The IPP Printer MUST specify this REQUIRED attribute using the registration-id it obtained
781 from the Server object via the Register-Notification-Source operation.

782 “subscription-id” (integer(0:MAX))
783 The IPP Printer MUST specify the ID of the Subscription object whose lease is being
784 extended.

785 “notify-lease-time-requested” (integer(0:MAX))
786 The IPP Printer MUST specify the time by which it wishes to extend the Subscription
787 object’s lease.

788 4.10.2 Renew-Subscription Response

789 The Server object returns the following set of attributes as part of the Renew-Subscription Response:

790 Group 1: Operation Attributes

791 Natural Language and Character Set:

792 The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566]
793 section 3.1.4.1.

794 “subscription-lease-expiration-time” (integer(0:86,400)):
795 This REQUIRED attribute specifies the time in the future when the Subscription’s lease will
796 expire. If the Server object is not able to grant the lease-time requested by the IPP Printer,
797 this attribute may contain a different value than the one provided in the request.

798 **ISSUE 04: What shall be done with the “notify-printer-up-time” operational attribute?**

799 4.11 Get-Subscriptions Operation

800 This REQUIRED operation allows an IPP Printer to get a list of the Subscription objects associated with a
801 given registration ID.

802 4.11.1 Get-Subscriptions Request

803 The Request for this operation is defined to be identical to the Request for the Get-Subscriptions operation
804 as specified in [ipp-ntfy], except for the following changes:

- 805 a) The “printer-uri” operational attribute is replaced by “server-uri” (uri) and MUST contain the URI of the
806 Notification Delivery Service.
- 807 b) The request MUST include the operational attribute “registration-id” (integer(0:MAX)) specifying the
808 registration-id the IPP Printer obtained from the Server object via the Register-Notification-Source
809 operation.

810 **4.11.2 Get Subscriptions Response**

811 The Response for this operation is defined to be identical to the Response for the Get-Subscriptions
812 operation as specified in [ipp-ntfy].

813 **ISSUE 05: What shall be done with the Subscription object attribute “notify-printer-up-time”?**

814 **5 Encoding of the Operation Layer**

815 INDP uses the same operation layer encoding model and syntax as IPP [ipp-pro] with the following
816 extensions:

817 **5.1 New attribute tag**

818 A new notification attributes tag is defined:

819 notification-attributes-tag = %x07 ; tag of 7

820 **5.2 New status codes**

821 **ISSUE 06 - Should we move the status codes into the Notification Model document in order to have the**
822 **same status codes for any other delivery method that might be defined?**

823 The following status codes are defined:

824 **5.2.1 unknown-notification-recipient. (0xXXX)**

825 The Notification Recipient returns this status code in order to indicate that the intended Ultimate
826 Notification Recipient is not known to the Notification Recipient.

827 **5.2.2 unable-to-delivery-notification-report (0xXXX)**

828 The Notification Recipient returns this status code in order to indicate that it was unable to deliver the event
829 Notification to the intended Ultimate Notification Recipient.

830 **5.2.3 successful-ok-but-cancel-subscription (0xXXXX)**

831 The Notification Recipient indicates that it no longer wants to receive Notifications for this Subscription
832 object. Therefore, the Subscription object is canceled. Note: this status code allows the Notification
833 Recipient to cancel a Subscription object without having to be the owner of the Subscription object. Only
834 the owner of the Subscription object can cancel a Subscription object using the Cancel-Subscription
835 operation.

836 **5.2.4 unknown-registration-id (0xXXX)**837 **5.2.5 successful-ok-but-error-accessing-persistent-storage (0xXXXX)**838 **5.3 Encoding**

839 The encoding of INDP is based strictly on the encoding used by IPP. This specification, however, defines a
 840 new Group tag which is used it to encode multiple notifications in a Request. As multiple instances of the
 841 same group type have only been included in operation Responses in the past, this section describes the
 842 encoding of an operation that uses the new tag for illustration purposes.

843 The encoding for the Send-Notification Request consists of:

844	-----		
845	version-number	2 byte	
846	-----		
847	operation-id	2 bytes	
848	-----		
849	request-id	4 bytes	
850	-----		
851	operation-attributes-tag	1 byte	
852	-----		
853	attributes-charset	u bytes	
854	-----		
855	attributes-natural-language	v bytes	
856	-----		
857	target-attribute	w bytes	
858	-----		
859	notification-attributes-tag	1 byte	- 1 or more
860	-----		
861	notification-attr-list	x bytes	
862	-----		
863	end-of-attributes-tag	1 byte	
864	-----		

865 Where:

866 *version-number* is made up of a major-version-number of %d1 and a minor-version-number of %d0
 867 indicating the 1.0 version of the 'ipp-notify-send' event notification delivery method and protocol.

868 *operation-id*, in the 1.0 version of the protocol, can only be 0x00003, Send-Notification.

869 *request-id* is any 4 byte number provided by the notification source and must be matched by the notification
 870 recipient in the corresponding response to a request. It assists the notification source in associating operation
 871 responses with their corresponding requests. Note that this request id is independent of the request id
 872 embedded in the notification report, which is opaque to the delivery method but assists the notification
 873 recipient order and identity missing or duplicate notification reports.

874 *operation-attribute tag*, *natural-language-attribute*, *charset-attribute*, *target-attribute*, and *end-of-*
 875 *attributes-tag* have the same syntax and semantics as in [ipp-pro].

876 *notification-attr-list* contains a list of the attributes that make up a single notification (see section 2 above)
 877 encoded using the syntax specified in [ipp-pro].

878 The encoding for the Send-Notification Response consists of:

879	-----		
880		version-number	2 byte
881	-----		
882		status-code	2 bytes
883	-----		
884		request-id	4 bytes
885	-----		
886		operation-attributes-tag	1 byte
887	-----		
888		attributes-charset	u bytes
889	-----		
890		attributes-natural-language	v bytes
891	-----		
892		target-attribute	w bytes
893	-----		
894		notification-attributes-tag	1 byte
895	-----		
896		ntfy-status-code	2 bytes
897	-----		
898		end-of-attributes-tag	1 byte
899	-----		

- 1 or more

900 **6 Encoding of Transport Layer**

901 HTTP/1.1 [rfc2616] is the transport layer for this protocol.

902 The operation layer has been designed with the assumption that the transport layer contains the following
 903 information:

- 904 - the URI of the target INDP operation.
- 905 - the total length of the data in the operation layer, either as a single length or as a sequence of
 906 chunks each with a length.

907 It is REQUIRED that a Notification Delivery Service and a 'indp://' Notification Recipient implementation
 908 support HTTP over the IANA assigned Well Known Port XXX (INDP's default port), though a notification
 909 recipient implementation MAY support HTTP over some other port as well.

910 Each HTTP operation MUST use the POST method where the request-URI is the object target of the
 911 operation, and where the "Content-Type" of the message-body in each request and response MUST be
 912 "application/ipp-notify-send". The message-body MUST contain the operation layer and MUST have the
 913 syntax described in section 3, "Encoding of Operation Layer". An INDP client implementation (be it an IPP
 914 Printer or a Notification Delivery Service) MUST adhere to the rules for a client described for HTTP1.1

915 [rfc2616]. An INDP server implementation (be it a Notification Delivery Method or a notification Recipient)
916 MUST adhere the rules for an origin server described for HTTP1.1 [rfc2616].

917 An INDP server implementation sends a response for each request that it receives. If it detects an error, it
918 MAY send a response before it has read the entire request. If the HTTP layer of the INDP server
919 implementation completes processing the HTTP headers successfully, it MAY send an intermediate
920 response, such as "100 Continue", with no notification data before sending the notification response. The
921 INDP client implementation MUST expect such a variety of responses. For further information on
922 HTTP/1.1, consult the HTTP documents [rfc2616].

923 An INDP server implementation MUST support chunking for HTTP notification requests, and an INDP
924 client implementation MUST support chunking for HTTP notification responses according to
925 HTTP/1.1[rfc2616]. Note: this rule causes a conflict with non-compliant implementations of HTTP/1.1 that
926 don't support chunking for POST methods, and this rule may cause a conflict with non-compliant
927 implementations of HTTP/1.1 that don't support chunking for CGI scripts

928 INDP uses 'indp://' as its URI scheme.

929 **7 IANA Considerations**

930 IANA will be asked to register this 'ipp-notify-send' notification delivery scheme and protocol and will be
931 asked to assign a default port.

932 **8 Internationalization Considerations**

933 When the client requests Human Consumable form by supplying the "notify-text-format" operation attribute
934 (see [ipp-ntfy]), the IPP Printer (or any Notification Service that the IPP Printer might be configured to use)
935 supplies and localizes the text value of the "human-readable-report" attribute in the Notification according to
936 the charset and natural language requested in the notification subscription.

937 **9 Security Considerations**

938 The IPP Model and Semantics document [ipp-mod] discusses high-level security requirements (Client
939 Authentication, Server Authentication and Operation Privacy). Client Authentication is the mechanism by
940 which the client proves its identity to the server in a secure manner. Server Authentication is the mechanism
941 by which the server proves its identity to the client in a secure manner. Operation Privacy is defined as a
942 mechanism for protecting operations from eavesdropping.

943 The Notification Recipient can cancel unwanted Subscriptions created by other parties without having to be
944 the owner of the subscription by returning the 'successful-ok-but-cancel-subscription' status code in the
945 Send-Notifications response returned to the Notification Source.

946 **9.1 Security Conformance**

947 Notification Sources (client) MAY support Digest Authentication [rfc2617]. If Digest Authentication is
948 supported, then MD5 and MD5-sess MUST be supported, but the Message Integrity feature NEED NOT be
949 supported.

950 Notification Recipient (server) MAY support Digest Authentication [rfc2617]. If Digest Authentication is
951 supported, then MD5 and MD5-sess MUST be supported, but the Message Integrity feature NEED NOT be
952 supported.

953 Notification Recipients MAY support TLS for client authentication, server authentication and operation
954 privacy. If a notification recipient supports TLS, it MUST support the
955 TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite as mandated by RFC 2246 [rfc2246]. All
956 other cipher suites are OPTIONAL. Notification recipients MAY support Basic Authentication (described in
957 HTTP/1.1 [rfc2616]) for client authentication if the channel is secure. TLS with the above mandated cipher
958 suite can provide such a secure channel.

959 **10 References**

960 [ipp-mod]

961 R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and
962 Semantics", <draft-ietf-ipp-model-v11-04.txt>, June, 1999.

963 [ipp-ntfy]

964 Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing
965 Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-02.txt>, February 2,
966 2000.

967 [ipp-pro]

968 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and
969 Transport", draft-ietf-ipp-protocol-v11-03.txt, June, 1999.

970 [rfc2026]

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

972 [rfc2616]

973 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext
974 Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.

975 [rfc2617]

976 J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, "HTTP
977 Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

978 11 Author's Addresses

979 Hugo Parra
980 Novell, Inc.
981 122 E 1700 S
982 Provo, UT 84606
983
984 Phone: 801-861-3307
985 Fax: 801-861-2517
986 e-mail: hparra@novell.com
987

988 Tom Hastings
989 Xerox Corporation
990 737 Hawaii St. ESAE 231
991 El Segundo, CA 90245
992
993 Phone: 310-333-6413
994 Fax: 310-333-5514
995 e-mail: hastings@cp10.es.xerox.com
996

997 12 Full Copyright Statement

998 Copyright (C) The Internet Society (2000). All Rights Reserved.

999 This document and translations of it may be copied and furnished to others, and derivative works that
1000 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
1001 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
1002 this paragraph are included on all such copies and derivative works. However, this document itself may not
1003 be modified in any way, such as by removing the copyright notice or references to the Internet Society or
1004 other Internet organizations, except as needed for the purpose of developing Internet standards in which
1005 case the procedures for copyrights defined in the Internet Standards process must be followed, or as
1006 required to translate it into languages other than English.

1007 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its
1008 successors or assigns.

1009 This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET
1010 SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES,
1011 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE
1012 OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
1013 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.