

1 INTERNET-DRAFT **There are 5 ISSUES highlighted like this.**  
2 <draft-ietf-ipp-not-~~http~~ntfy-delivery-00.txt>

Hugo Parra  
Novell, Inc.  
Tom Hastings  
Xerox Corp.

~~October 19~~December 9, 1999

8 Internet Printing Protocol/1.1: ~~HTTP-Based IPP~~ **The 'ipp-ntfy' Notification Delivery Method and**  
9 **Protocol**

10 Copyright (C) The Internet Society (1999). All Rights Reserved.

11 **ISSUE 01 - What should the name of this delivery method and protocol be that we use in the title of this**  
12 **document?**

13 **ISSUE 02 - What should the scheme name be? Consider 'ipp-ntfy' a working title, until we see several**  
14 **schemes. The 'ipp-get' delivery method is another example. Should the scheme name somehow include**  
15 **"notification", i.e., 'ntfy'? How about 'ipp-ntfy-send' or 'ipp-ntfy-push' and 'ipp-ntfy-get' or 'ipp-ntfy-pull' to**  
16 **go with the Send-Notifications and Get-Notifications operations, respectively?**

17 **ISSUE 03 - Should the scheme name be used in the title?**

18 Status of this Memo

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

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

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

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

28 **Abstract**

29 The IPP event notification specification [ipp-ntfy] **is an OPTIONAL extension to IPP/1.0 and IPP/1.1.**  
30 **[ipp-ntfy] requires the availability definition of one or more delivery methods for dispatching event**  
31 **notification reports to interested parties Notification Recipients. This document describes the semantics and**  
32 **syntax of a protocol that a the 'ipp-ntfy' event notification delivery method that is itself a request/response**  
33 **protocol. For this delivery method, an IPP Printer may use to deliver sends (pushes) IPP event**  
34 **Notifications to the Notification Recipients using the protocol defined herein which includes HTTP ~~for as~~ a**  
35 **transport.**

36 The full set of IPP documents includes:

37 Design Goals for an Internet Printing Protocol [RFC2567]

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

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

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

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

42 Mapping between LPD and IPP Protocols [RFC2569]

43

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

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

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

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

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

65

66

**Table of Contents**

67	1	Introduction .....	4
68	2	Terminology .....	4
69	3	Model and Operation .....	4
70		Notification Operations .....	5
71		SEND-NOTIFICATIONS OPERATION .....	6
72	4.1.1	<i>Send-Notifications Request</i> .....	6
73	4.1.2	<i>Send-Notifications Response</i> .....	7
74		NOTIFICATION PROTOCOL URI SCHEME .....	8
75	5	Encoding of the Operation Layer .....	8
76	5.1	NEW ATTRIBUTE TAG: .....	8
77	5.2	NEW STATUS CODES: .....	8
78		<i>unknown-notification-recipient (0xYYYY)</i> .....	9
79		<i>unable-to-delivery-notification-report (0xZZZZ)</i> .....	9
80	5.2.3	<i>successful-ok-but-cancel-subscription (0XXXXX)</i> .....	9
81	6	Encoding of Transport Layer .....	10
82	7	IANA Considerations .....	11
83	8	Internationalization Considerations .....	11
84	9	Security Considerations .....	12
85	9.1	SECURITY CONFORMANCE .....	12
86	10	References .....	13
87	11	Author's Addresses .....	13
88	12	Full Copyright Statement .....	14
89			

90

## 91 1 Introduction

92 IPP **p**Printers that support [the OPTIONAL IPP event notification extension \[ipp-ntfy\]](#) either a) accept, store,  
93 and use notification Subscriptions to generate notification reports and implement one or more delivery  
94 methods for notifying interested parties, or b) support a subset of these tasks and farm out the remaining  
95 tasks to a Notification Delivery Service. The ~~protocol~~ ['ipp-ntfy' event notification delivery method](#) specified  
96 in this document [is itself a request/response protocol that](#) may be used in a variety of notification scenarios.  
97 Its primary intended use is for IPP **p**Printers to send [\(push\) event](#) notifications to **n**Notification **r**Recipients  
98 [using the IPP Send-Notifications operation](#) over HTTP. However, it may also be used by IPP **p**Printers to  
99 send notification to Notification Services and by Notification Delivery Services to send notifications to [the](#)  
100 [Ultimate \*\*n\*\*Notification \*\*r\*\*Recipients \(see \[ipp-ntfy\]\)](#). [Furthermore, this protocol can be extended in the](#)  
101 [future to add other operations, such as querying a Notification Recipient for its capabilities](#).

## 102 2 Terminology

103 [This document uses terms such as "attributes", "keywords", and "support". These terms have special](#)  
104 [meaning and are defined in the model terminology \[ipp-mod\] section 12.2.](#)

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

108 [This section defines the following additional terms that are used throughout this document:](#)

109 [REQUIRED: if an implementation supports the extensions described in this document, it MUST](#)  
110 [support a REQUIRED feature.](#)

111 [OPTIONAL: if an implementation supports the extensions described in this document, it MAY support](#)  
112 [an OPTIONAL feature.](#)

113 [Event Notification \(Notification for short\) - See \[ip-ntfy\]](#)

114 [Notification Source - See \[ipp-ntfy\]](#)

115 [Notification Recipient - See \[ipp-ntfy\]](#)

116 [Subscription object - See \[ipp-ntfy\]](#)

117 [Ultimate Notification Recipient - See \[ipp-ntfy\]](#)

## 118 3 Model and Operation

119 [In the IPP Notification Model \[ipp-ntfy\], one or more Per-Job Subscriptions can be supplied in the Job](#)  
120 [Creation operation or OPTIONALLY as subsequent Create-Job-Subscription operations; one Per-Printer](#)  
121 [Subscription can be supplied in the Create-Printer operation. The client that creates these Subscription](#)  
122 [objects becomes the owner of the Subscription object.](#)

123 [When creating each Subscription object, the client supplies the "notify-recipient" \(uri\) attribute. The](#)  
124 ["notify-recipient" attribute specifies both a single Notification Recipient that is to receive the Notifications](#)

125 when subsequent events occur and the method for notification delivery that the IPP Printer is to use. For  
126 the Notification delivery method defined in this document, the notification method is 'ipp-ntfy' and the rest  
127 of the URI is the address of the Notification Recipient to which the IPP Printer will send the Send-  
128 Notifications operations using HTTP as a transport.

129 The 'ipp-ntfy' event notification delivery method defined in this document is also HTTP-Based IPP  
130 Notification Protocol, hereafter referred to as HTTP-notification protocol, is a client/server protocol. The  
131 "client" in this HTTP relationship is the Notification Source described in [ipp-ntfy] while the "server" is the  
132 Notification Recipient. The Notification Source invokes the Send-Notifications operations supported by the  
133 'ipp-ntfy' HTTP-notification protocol to communicate IPP event Notification contents to the Notification  
134 Recipient. The Notification Recipient only conveys information to the Notification Source in the form of  
135 responses to the operations initiated by the Notification Source.

136 HTTP-notification All requests defined for this protocol will be issued as HTTP POST operations and their  
137 corresponding HTTP notification responses will be returned in the responses to those HTTP POST  
138 operations. Hence, Notification Sources that implement the HTTP-notification-'ipp-ntfy' delivery method  
139 and protocol will need to include an HTTP client stack while notification recipients that implement this  
140 protocol will need to support an HTTP server stack (see section 46 for more details).

### 141 **3.14** HTTP Notification Operations

142 The job of an HTTP Notification Source is to use the contents of an IPP Notification as defined in [ipp-  
143 ntfy] to compose the information defined for an IPP Notification [ipp-ntfy] and sends it using the Sent-  
144 Notifications operation to the invoke the appropriate HTTP-notification operation and send it to the  
145 specified HTTP Notification Recipient supplied in the Subscription object.

146 The HTTP-'ipp-ntfy' delivery method and notification protocol makes extensive use of the operations model  
147 defined by IPP [rfc2566]. This includes, the use of a URI as the identifier for the target of each operation,  
148 the inclusion of a version number, operation-id, and request-id in each request, and the definition of  
149 attribute groups. The HTTP-notification protocol-Send-Notifications operation uses the Operation  
150 Attributes group, but currently has no need for the Unsupported Attributes, Printer Object Attributes, and  
151 Job-Object Attributes groups. However, it defines-uses a new attribute group, the Notification-Generic  
152 Attributes group.

153 **ISSUE 04 - Ok to add a "Generic Attributes" group tag to [ipp-pro], instead of adding a special tag for each**  
154 **new object and/or operation that needs a different set of attributes than Job or Printer? The same issue for**  
155 **the Subscription object in [ipp-ntfy]. Either we define separate tags for both or use a single generic tag for**  
156 **both and future objects and attribute groups.**

157 In its 1.0 version, the HTTP-'ipp-ntfy' delivery method and notification protocol is composed of a single  
158 operation, but may be extended in the future as needed (e.g., to find out specific capabilities of an HTTP  
159 'ipp-ntfy' Notification listener Recipient). The operation currently defined is Send-Notifications.

#### 160 [4.1.14.1 Report-IppSend-Notifications Operation](#)

161 This REQUIRED operation allows a Notification Source to send one or more Notifications to [a](#) Notification  
162 Recipient using HTTP. The operation has been tailored to accommodate the current definition of IPP  
163 Notification [\[ipp-ntfy\]](#).

164 Both ~~'machine-consumable'~~[Machine-Consumable](#) and ~~'human-consumable'~~[Human-Consumable](#)  
165 notifications may be sent to ~~an HTTP~~[a](#) Notification Recipient through this operation.

#### 166 [3.1.1.4.1.1 Send-Notifications Request](#)

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

168 Group 1: Operation Attributes

169 Natural Language and Character Set:

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

172

173 Target:

174 The URI of the [HTTP'ipp-ntfy'](#) Notification Recipient.

175 Group 2 to N: Notification Attributes

176 "human-readable-report" (text)

177 The [HTTP'ipp-ntfy'](#) Notification Source OPTIONALLY ~~supplies~~[supports](#) this attribute. [A](#)[This](#)  
178 [attribute is a](#) text string generated by the IPP printer or Notification Delivery Service from the  
179 contents of the IPP Notification suitable for human consumption. [If the Notification Source](#)  
180 [supports this attribute, it MUST supply this attribute if the Subscription object contains the "notify-](#)  
181 [text-format" \(mimeType\) attribute. The text value of this attribute MUST be localized in the](#)  
182 [charset identified by the "notify-charset" \(charset\) attribute and the natural language identified by](#)  
183 [the notify-natural-language" \(naturalLanguage\) attribute supplied in the associated Subscription](#)  
184 [object that generates this event Notification. The format of the text value is specified by the value](#)  
185 [of the "notify-text-format" \(mimeType\) supplied in the associated Subscription object.](#)

186 **ISSUE 5 - Ok to extend Notification Model to allow a single notification to have both Human**  
187 **Consumable form and Machine Consumable form when the client asks for Human Consumable**  
188 **form by supplying the "notify-text-format" attribute rather than the Human Consumable being sent**  
189 **instead or in addition to the Machine Consumable using MIME multi-part-related?**

190 [All of the REQUIRED attributes and any of the OPTIONAL attributes indicated in \[ipp-ntfy\] for a Push](#)  
191 [event Notification, including "notify-text-format-type" \(mimeType\), if the "human-readable-](#)  
192 [report" \(text\) attribute is included, so that the Notification Recipient will know the text format of the](#)  
193 ["human-readable-report" \(text\) attribute value.](#)

194 ~~"version-number" (integer (0:32767))~~  
195 ~~"status-code" (integer (0:32767))~~  
196 ~~"request-id" (integer (0:MAX))~~  
197 ~~"attributes-charset" (charset)~~  
198 ~~"attributes-natural-language" (naturalLanguage)~~  
199 ~~"printer-uri" (uri)~~  
200 ~~"printer-name" (name(127))~~  
201 ~~"job-id" (integer(1:MAX))~~  
202 ~~"job-name" (name(MAX))~~  
203 ~~"trigger-event" (type2 keyword)~~  
204 ~~"trigger-time" (integer(MIN:MAX))~~  
205 ~~"trigger-date-time" (dateTime)~~  
206 ~~"subscription-id" (integer(1:MAX))~~  
207 ~~"subscriber-user-name" (name(MAX))~~  
208 ~~"subscriber-user-data" (octetString(63))~~  
209 ~~"job-state" (type1 enum)~~  
210 ~~"job-state-reasons" (1setOf type2 keyword)~~  
211 ~~"job-k-octets-processed" (integer(0:MAX))~~  
212 ~~"job-impressions-completed" (integer(0:MAX))~~  
213 ~~"job-media-sheets-completed" (integer(0:MAX))~~  
214 ~~"job-collation-type" (type2 enum)~~  
215 ~~"sheet-completed-copy-number" (integer(-2:MAX))~~  
216 ~~"sheet-completed-document-number" (integer(-2:MAX))~~  
217 ~~"impressions-interpreted" (integer(-2:MAX))~~  
218 ~~"impressions-completed-current-copy" (integer(-2:MAX))~~  
219 ~~"printer-state" (type1 enum)~~  
220 ~~"printer-state-reasons" (1setOf type2 keyword)~~  
221 ~~"printer-is-accepting-jobs" (boolean)~~

222 These attributes communicate the same information as the notification attributes by the same name  
223 described in sections 7.4, 7.5, and 7.6 of [ipp-ntfy]. The rules that govern when each individual attribute  
224 MUST or MAY be included in this operation precisely mirror those specified in [ipp-ntfy].

### 225 [3.1.1.24.1.2](#) **Send-Notifications Response**

226 The [HTTP'ipp-ntfy'](#) Notification Recipient returns a status code for the entire operation and one for each  
227 Notification Report in the request if the operation's status code is other than "success-ok". If the [HTTP'ipp-](#)  
228 [ntfy'](#) notification listener receives a Notification report that it can't pair up with a subscription it knows  
229 about, it can return an error status-code to indicate that events associated with that subscription should no  
230 longer be sent to it.

231 Group 1: Operation Attributes

232 Natural Language and Character Set:

233 The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section  
234 3.1.4.1.

235 Group 2 to N: Notification Attributes

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

237 Indicates whether the [HTTP 'ipp-ntfy' Notification listener-Recipient](#) was able to consume the n-th  
238 Notification Report.

## 239 4.2 [HTTP Notification Protocol URI Scheme](#)

240 [The 'ipp-ntfy' event notification delivery method uses the 'ipp-ntfy' URI scheme in the "notify-recipients"](#)  
241 [attribute in the Subscription object in order to indicate the event notification delivery method defined in this](#)  
242 [document. The remainder of the URI indicates the host and address of the Notification Recipient that is to](#)  
243 [receive the Send-Notification operation.](#)

244 REPEAT OF ISSUE 02 - What should the scheme name be? Consider 'ipp-ntfy' a working title, until we  
245 see several schemes. The 'ipp-get' delivery method is another example. Should the scheme name somehow  
246 include "notification", i.e., 'ntfy'? How about 'ipp-ntfy-send' or 'ipp-ntfy-push' and 'ipp-ntfy-get' or 'ipp-ntfy-  
247 pull' to go with the Send-Notifications and Get-Notifications operations, respectively?

248 *ISSUE 2 - Should the URI scheme for this protocol be "http://", "ipp://", or something else like "ipp-  
249 ntfy://". If we intent this proposal to go to the IESG, something along the lines of the third option might be  
250 our only alternative*

## 251 5 Encoding of the Operation Layer

252 The [HTTP 'ipp-ntfy' event notification delivery method and](#) protocol uses the same operation layer  
253 encoding model and syntax as IPP [ipp-pro] with two extensions:

### 254 [5.1 New attribute tag:](#)

255 a) A new attribute tag is defined:

256 ~~notification-report~~[generic-attributes](#)-tag = %x07 ; tag of 7

### 257 [5.2 New status codes:](#)

258 b) The following status codes are defined:



259 **5.2.1 0xYYYY--unknown-notification-recipient. (0xYYYY)**

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

262 **5.2.2 0xZZZZ--unable-to-delivery-notification-report (0xZZZZ)**

263 The Notification Recipient returns this status code in ordre to indicate that it was unable to deliver the event  
 264 Notification to the intended Ultimate Notification Recipient.

265 **5.2.3 successful-ok-but-cancel-subscription (0xXXXX)**

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

271 ~~ISSUE 3—Should we add a success status code, say, 'successful-ok-but-cancel-subscription'~~  
 272 ~~which requests that the subscription be canceled. Then the Notification Recipient can cancel~~  
 273 ~~a subscription that another party established even though the Notification Recipient is not~~  
 274 ~~the owner of the Subscription.~~  
 275

276 The encoding for the [Report-IPP-Send-Notification Request](#) consists of:

277	-----		
278	version-number	2 byte	
279	-----		
280	operation-id	2 bytes	
281	-----		
282	request-id	4 bytes	
283	-----		
284	operation-attributes-tag	1 byte	
285	-----		
286	natural-language-attribute	u bytes	
287	-----		
288	charset-attribute	v bytes	
289	-----		
290	target-attribute	w bytes	
291	-----		
292	<del>notification</del> generic-attributes-tag	1 byte	
293	-----		- 1 or more
294	notification-attr-list	x bytes	
295	-----		
296	end-of-attributes-tag	1 byte	
297	-----		

298 Where:

299 *version-number* is made up of a major-version-number of %d1 and a minor-version-number of %d0  
 300 indicating the 1.0 version of the ['ipp-ntfy' event HTTP-notification delivery method and](#) protocol.

301 *operation-id*, in the 1.0 version of the protocol, can only be 0x00003, [Report-IPP](#)Send-Notification.

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

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

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

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

312	-----			
313		version-number		2 byte
314	-----			
315		status-code		2 bytes
316	-----			
317		request-id		4 bytes
318	-----			
319		operation-attributes-tag		1 byte
320	-----			
321		natural-language-attribute		u bytes
322	-----			
323		charset-attribute		v bytes
324	-----			
325		target-attribute		w bytes
326	-----			
327		<del>notification</del> <u>generic-attributes</u> -tag		1 byte
328	-----			
329		ntfy-status-code		2 bytes
330	-----			
331		end-of-attributes-tag		1 byte
332	-----			

\  
 |  
 > ~~Not needed in 1.0~~  
 > ~~<ISSUE 4: Do we~~  
 > ~~want to keep it?>~~  
 |  
 /  
 - 1 or more

## 333 6 Encoding of Transport Layer

334 HTTP/1.1 [[rfc2616](#)~~rfc2068~~] is the transport layer for this protocol.

335 The operation layer has been designed with the assumption that the transport layer contains the following  
 336 information:

337 - the URI of the target job or printer operation.

338 - the total length of the data in the operation layer, either as a single length or as a sequence of  
339 chunks each with a length.

340 It is REQUIRED that an [HTTP 'ipp-ntfy' Notification Recipient](#) implementation support HTTP over the  
341 IANA assigned Well Known Port XXX (the [HTTP 'ipp-ntfy'](#) notification protocol default port), though a  
342 notification recipient implementation MAY support HTTP over some other port as well.

343 Each HTTP operation MUST use the POST method where the request-URI is the object target of the  
344 operation, and where the "Content-Type" of the message-body in each request and response MUST be  
345 "application/ipp-ntfy". The message-body MUST contain the operation layer and MUST have the syntax  
346 described in section 3, "Encoding of Operation Layer". An ['ipp-ntfy' HTTP Notification Source](#)  
347 implementation MUST adhere to the rules for a client described for HTTP1.1 [[rfc2616#fe2068](#)]. An ['ipp-  
348 ntfy' HTTP Notification Recipient](#) implementation MUST adhere the rules for an origin server described for  
349 HTTP1.1 [[rfc2616#fe2068](#)].

350 An ['ipp-ntfy' HTTP Notification Source](#) sends a response for each request that it receives. If a notification  
351 recipient detects an error, it MAY send a response before it has read the entire request. If the HTTP layer of  
352 the Notification Recipient completes processing the HTTP headers successfully, it MAY send an  
353 intermediate response, such as "100 Continue", with no notification data before sending the notification  
354 response. [The 'ipp-ntfy' HTTP Notification Sources](#) MUST expect such a variety of responses from  
355 notification recipients. For further information on HTTP/1.1, consult the HTTP documents  
356 [[rfc2616#fe2068](#)].

357 An ['ipp-ntfy' HTTP Notification Recipient \(server\)](#) MUST support chunking for HTTP notification  
358 requests, and an ['ipp-ntfy' HTTP Notification Source \(client\)](#) MUST support chunking for HTTP  
359 notification responses according to HTTP/1.1 [[rfc2616#fe2068](#)]. Note: this rule causes a conflict with non-  
360 compliant implementations of HTTP/1.1 that don't support chunking for POST methods, and this rule may  
361 cause a conflict with non-compliant implementations of HTTP/1.1 that don't support chunking for CGI  
362 scripts

## 363 7 IANA Considerations

364 IANA will be asked to register this ['ipp-ntfy' HTTP](#)-notification delivery scheme [and protocol](#) and [will be](#)  
365 [asked to](#) assign a default port.

## 366 8 Internationalization Considerations

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

## 371 9 Security Considerations

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

377 ~~If we add the 'successful-ok-but-cancel-subscription' (see ISSUE 3 in section 3), then~~ The Notification  
378 Recipient can cancel unwanted Subscriptions created by other parties without having to be the owner of the  
379 subscription by returning the 'successful-ok-but-cancel-subscription' status code in the Send-Notifications  
380 response returned to the Notification Source.

### 381 9.1 Security Conformance

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

385 - ~~Digest Authentication [rfc2069].~~

386 ~~MD5 and MD5-sess MUST be implemented and supported.~~

387 - ~~The Message Integrity feature NEED NOT be used.~~

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

391 ~~Digest Authentication [rfc2069].~~

392 ~~MD5 and MD5-sess MUST be implemented and supported.~~

393 ~~The Message Integrity feature NEED NOT be used.~~

394 Notification Recipients MAY support TLS for client authentication, server authentication and operation  
395 privacy. If a notification recipient supports TLS, it MUST support the  
396 TLS\_DHE\_DSS\_WITH\_3DES\_EDE\_CBC\_SHA cipher suite as mandated by RFC 2246 [rfc2246]. All  
397 other cipher suites are OPTIONAL. Notification recipients MAY support Basic Authentication (described  
398 in HTTP/1.1 [rfc2616rfc2068]) for client authentication if the channel is secure. TLS with the above  
399 mandated cipher suite can provide such a secure channel.

## 400 10 References

401 [ipp-mod]

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

404 [ipp-ntfy]

405 Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing  
406 Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-01.txt>, October 14,  
407 1999.

408 [ipp-pro]

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

411 [\[rfc2026\]](#)

412 [S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.](#)

413 [\[rfc2068\]](#)

414 [R. Fielding, et al, "Hypertext Transfer Protocol—HTTP/1.1" RFC 2068, January 1997.](#)

415 [\[rfc2566\]](#)

416 [deBry, R., Hastings, T., Herriot, R., Isaacson, S., Powell, P., "Internet Printing Protocol/1.0: Model  
417 and Semantics", RFC 2566, April 1999.](#)

418 [\[rfc2616\]](#)

419 [R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext  
420 Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.](#)

421 [\[rfc2617\]](#)

422 [J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, "HTTP  
423 Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.](#)

## 424 11 Author's Addresses

425 Hugo Parra  
426 Novell, Inc.  
427 122 E 1700 S  
428 Provo, UT 84606

429  
430 Phone: 801-861-3307

431 Fax: 801-861-2517

432 e-mail: hparra@novell.com

433  
434 [Tom Hastings](#)

435 [Xerox Corporation](#)

436 [737 Hawaii St. ESAE 231](#)  
437 [El Segundo, CA 90245](#)  
438  
439 [Phone: 310-333-6413](#)  
440 [Fax: 310-333-5514](#)  
441 [e-mail: \[hastings@cp10.es.xerox.com\]\(mailto:hastings@cp10.es.xerox.com\)](#)  
442

## 443 **12 Full Copyright Statement**

444 Copyright (C) The Internet Society (1999). All Rights Reserved.

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

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

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