

1 INTERNET-DRAFT      **There are 5 ISSUES highlighted like this.**  
2 <draft-ietf-ipp-notify-send-ntfy-delivery-00.txt>

3  
4 Hugo Parra  
5 Novell, Inc.  
6 Tom Hastings  
7 Xerox Corp.

February 2, 2000 ~~December 9, 1999~~

8 Internet Printing Protocol/1.1: **The 'ipp-ntfy'-notify-send' Notification Delivery Method and Protocol**  
9 Copyright (C) The Internet Society (1999). All Rights Reserved.

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

12 **ISSUE 02 - What should the scheme name be? Consider 'ipp-notify-send' a working title, until we see**  
13 **several schemes. The 'ipp-notify-poll' delivery method is another example. The IETF likes words or well-**  
14 **recognized acronyms, not abbreviations in scheme names, so lets include "notify"?**

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

16 Status of this Memo

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

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

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

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

## 25 Abstract

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

32 The full set of IPP documents includes:

33 Design Goals for an Internet Printing Protocol [RFC2567]

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

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

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

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

38 Mapping between LPD and IPP Protocols [RFC2569]

39

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

45 The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document  
46 describes IPP from a high level view, defines a roadmap for the various documents that form the suite of  
47 IPP specification documents, and gives background and rationale for the IETF working group's major  
48 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  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85

**Table of Contents**

1 Introduction ..... 4

2 Terminology ..... 4

3 Model and Operation ..... 4

4 Notification Operations ..... 5

    4.1 SEND-NOTIFICATIONS OPERATION ..... 6

        4.1.1 *Send-Notifications Request* ..... 6

        4.1.2 *Send-Notifications Response* ..... 7

    4.2 NOTIFICATION PROTOCOL URI SCHEME ..... 7

5 Encoding of the Operation Layer ..... 7

    5.1 NEW ATTRIBUTE TAG ..... 8

    5.2 NEW STATUS CODES ..... 8

        5.2.1 *unknown-notification-recipient (0xYYYY)* ..... 8

        5.2.2 *unable-to-delivery-notification-report (0xZZZZ)* ..... 8

        5.2.3 *successful-ok-but-cancel-subscription (0XXXXX)* ..... 8

6 Encoding of Transport Layer ..... 10

7 IANA Considerations ..... 10

8 Internationalization Considerations ..... 11

9 Security Considerations ..... 11

    9.1 SECURITY CONFORMANCE ..... 11

10 References ..... 11

11 Author's Addresses ..... 12

12 Full Copyright Statement ..... 12

86

87 **1 Introduction**

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

98 **2 Terminology**

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

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

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

- 105     **REQUIRED:** if an implementation supports the extensions described in this document, it **MUST**  
 106         support a **REQUIRED** feature.
- 107     **OPTIONAL:** if an implementation supports the extensions described in this document, it **MAY** support  
 108         an **OPTIONAL** feature.
- 109     Event Notification (Notification for short) - See [ip-ntfy]
- 110     Notification Source - See [ipp-ntfy]
- 111     Notification Recipient - See [ipp-ntfy]
- 112     Subscription object - See [ipp-ntfy]
- 113     Ultimate Notification Recipient - See [ipp-ntfy]

114 **3 Model and Operation**

115 In the IPP Notification Model [ipp-ntfy], one or more Per-Job Subscriptions can be supplied in the Job  
 116 Creation operation or **OPTIONALLY** as subsequent Create-Job-Subscription operations; one Per-Printer  
 117 Subscription can be supplied in the Create-Printer operation. The client that creates these Subscription  
 118 objects becomes the owner of the Subscription object.

119 When creating each Subscription object, the client supplies the "notify-recipient" (uri) attribute. The  
 120 "notify-recipient" attribute specifies both a single Notification Recipient that is to receive the Notifications

121 when subsequent events occur and the method for notification delivery that the IPP Printer is to use. For  
122 the Notification delivery method defined in this document, the notification method is ['ipp-ntfyipp-notify-  
123 send'](#) and the rest of the URI is the address of the Notification Recipient to which the IPP Printer will send  
124 the Send-Notifications operations using HTTP as a transport.

125 The ['ipp-ntfyipp-notify-send'](#) event notification delivery method defined in this document is also a  
126 client/server protocol. The "client" in this HTTP relationship is the Notification Source described in [ipp-  
127 ntfy] while the "server" is the Notification Recipient. The Notification Source invokes the Send-  
128 Notifications operation supported by the ['ipp-ntfyipp-notify-send'](#) notification protocol to communicate IPP  
129 event Notification contents to the Notification Recipient. The Notification Recipient only conveys  
130 information to the Notification Source in the form of responses to the operations initiated by the  
131 Notification Source.

132 All requests defined for this protocol will be issued as HTTP POST operations and their corresponding  
133 HTTP notification responses will be returned in the responses to those HTTP POST operations. Hence,  
134 Notification Sources that implement the ['ipp-ntfyipp-notify-send'](#) delivery method and protocol will need to  
135 include an HTTP client stack while notification recipients that implement this protocol will need to support  
136 an HTTP server stack (see section 6 for more details).

## 137 4 Notification Operations

138 The Notification Source composes the information defined for an IPP Notification [ipp-ntfy] and sends it  
139 using the Sent-Notifications operation to the Notification Recipient supplied in the Subscription object.

140 The ['ipp-ntfyipp-notify-send'](#) delivery method and notification protocol makes extensive use of the  
141 operations model defined by IPP [rfc2566]. This includes, the use of a URI as the identifier for the target of  
142 each operation, the inclusion of a version number, operation-id, and request-id in each request, and the  
143 definition of attribute groups. The Send-Notifications operation uses the Operation Attributes group, but  
144 currently has no need for the Unsupported Attributes, Printer Object Attributes, and Job-Object Attributes  
145 groups. However, it uses a new attribute group, the [Generic-Notification](#) Attributes group.

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

150 In its 1.0 version, the ['ipp-ntfyipp-notify-send'](#) delivery method and notification protocol is composed of a  
151 single operation, but may be extended in the future as needed (e.g., to find out specific capabilities of an  
152 ['ipp-ntfyipp-notify-send'](#) Notification Recipient). The operation currently defined is Send-Notifications.

## 153 4.1 Send-Notifications Operation

154 This REQUIRED operation allows a Notification Source to send one or more Notifications to a Notification  
155 Recipient using HTTP. The operation has been tailored to accommodate the current definition of IPP  
156 Notification [ipp-ntfy].

157 Both Machine-Consumable and Human-Consumable notifications may be sent to a Notification Recipient  
158 through this operation.

### 159 4.1.1 Send-Notifications Request

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

161 Group 1: Operation Attributes

162 Natural Language and Character Set:

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

165

166 Target:

167 The URI of the '[ipp-ntfyipp-notify-send](#)' Notification Recipient.

168 Group 2 to N: Notification Attributes

169 "human-readable-report" (text)

170 The '[ipp-ntfyipp-notify-send](#)' Notification Source OPTIONALLY supports this attribute. This  
171 attribute is a text string generated by the IPP printer or Notification Delivery Service from the  
172 contents of the IPP Notification suitable for human consumption. If the Notification Source  
173 supports this attribute, it MUST supply this attribute if the Subscription object contains the "notify-  
174 text-format" (mimeType) attribute. The text value of this attribute MUST be localized in the  
175 charset identified by the "notify-charset" (charset) attribute and the natural language identified by  
176 the notify-natural-language" (naturalLanguage) attribute supplied in the associated Subscription  
177 object that generates this event Notification. The format of the text value is specified by the value  
178 of the "notify-text-format" (mimeType) supplied in the associated Subscription object.

179

180 **ISSUE 04 - "human-readable-report" has been added to the [ipp-ntfy] Notification Model document,**  
181 **so ok to change this description to be a reference to "human-readable-report" in [ipp-ntfy]?**

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

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

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

#### 193 4.1.2 Send-Notifications Response

194 The '~~ipp-ntfy~~ipp-notify-send' Notification Recipient returns a status code for the entire operation and one for  
 195 each Notification Report in the request if the operation's status code is other than "success-ok". If the '~~ipp-  
 196 ntfy~~ipp-notify-send' notification listener receives a Notification report that it can't pair up with a  
 197 subscription it knows about, it can return an error status-code to indicate that events associated with that  
 198 subscription should no longer be sent to it.

199 Group 1: Operation Attributes

200 Natural Language and Character Set:

201 The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section  
 202 3.1.4.1.

203 Group 2 to N: Notification Attributes

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

205 Indicates whether the '~~ipp-ntfy~~ipp-notify-send' Notification Recipient was able to consume the n-th  
 206 Notification Report.

#### 207 4.2 Notification Protocol URI Scheme

208 The '~~ipp-ntfy~~ipp-notify-send' event notification delivery method uses the '~~ipp-ntfy~~ipp-notify-send' URI  
 209 scheme in the "notify-recipients" attribute in the Subscription object in order to indicate the event  
 210 notification delivery method defined in this document. The remainder of the URI indicates the host and  
 211 address of the Notification Recipient that is to receive the Send-Notification operation.

212 REPEAT OF ISSUE 02 - What should the scheme name be? Consider 'ipp-notify-send' a working title,  
 213 until we see several schemes.

### 214 5 Encoding of the Operation Layer

215 The '~~ipp-ntfy~~ipp-notify-send' event notification delivery method and protocol uses the same operation layer  
 216 encoding model and syntax as IPP [ipp-pro] with two extensions:

217 **5.1 New attribute tag:**218 a) A new notification attributes tag is defined:219 genericnotification-attributes-tag = %x07 ; tag of 7220 **5.2 New status codes:**221 ISSUE 05 - Should we move the status codes into the Notification Model document in order to have the  
222 same status codes for any other delivery method that might be defined?

223 b) The following status codes are defined:

224 **5.2.1 unknown-notification-recipient. (0xYYYY)**225 The Notification Recipient returns this status code in order to indicate that the intended Ultimate  
226 Notification Recipient is not known to the Notification Recipient.227 **5.2.2 unable-to-delivery-notification-report (0xZZZZ)**228 The Notification Recipient returns this status code in order to indicate that it was unable to deliver the event  
229 Notification to the intended Ultimate Notification Recipient.230 **5.2.3 successful-ok-but-cancel-subscription (0XXXXX)**231 The Notification Recipient indicates that it no longer wants to receive Notifications for this Subscription  
232 object. Therefore, the Subscription object is canceled. Note: this status code allows the Notification  
233 Recipient to cancel a Subscription object without having to be the owner of the Subscription object. Only  
234 the owner of the Subscription object can cancel a Subscription object using the Cancel-Subscription  
235 operation.

236

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

238	-----	
239	version-number	2 byte
240	-----	
241	operation-id	2 bytes
242	-----	
243	request-id	4 bytes
244	-----	
245	operation-attributes-tag	1 byte
246	-----	
247	<del>attributes-charset</del> <del>natural language attribute</del>	u bytes
248	-----	
249	<del>attributes-natural language</del> <del>charset attribute</del>	v bytes
250	-----	
251	target-attribute	w bytes



```

252 -----
253 | genericnotification-attributes-tag | 1 byte |
254 |-----| - 1 or more
255 | notification-attr-list | x bytes |
256 |-----|
257 | end-of-attributes-tag | 1 byte
258 |-----|

```

259 Where:

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

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

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

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

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

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

```

273 -----
274 | version-number | 2 byte
275 |-----|
276 | status-code | 2 bytes
277 |-----|
278 | request-id | 4 bytes
279 |-----|
280 | operation-attributes-tag | 1 byte
281 |-----|
282 | natural language attributeattributes-charset | u |
283 bytes
284 |-----|
285 | attributes-natural-languagecharset-attribute | v |
286 bytes
287 |-----|
288 | target-attribute | w bytes
289 |-----|
290 | genericnotification-attributes-tag | 1 byte |
291 |-----| - 1 or more
292 | ntfy-status-code | 2 bytes |
293 |-----|

```

294 | end-of-attributes-tag | 1 byte  
295 -----

## 296 6 Encoding of Transport Layer

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

298 The operation layer has been designed with the assumption that the transport layer contains the following  
299 information:

- 300 - the URI of the target job or printer operation.
- 301 - the total length of the data in the operation layer, either as a single length or as a sequence of  
302 chunks each with a length.

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

306 Each HTTP operation MUST use the POST method where the request-URI is the object target of the  
307 operation, and where the "Content-Type" of the message-body in each request and response MUST be  
308 "application/[ipp-ntfyipp-notify-send](#)". The message-body MUST contain the operation layer and MUST  
309 have the syntax described in section 3, "Encoding of Operation Layer". An '[ipp-ntfyipp-notify-send](#)'  
310 Notification Source implementation MUST adhere to the rules for a client described for HTTP1.1  
311 [rfc2616]. An '[ipp-ntfyipp-notify-send](#)' Notification Recipient implementation MUST adhere the rules for  
312 an origin server described for HTTP1.1 [rfc2616].

313 An '[ipp-ntfyipp-notify-send](#)' Notification Source sends a response for each request that it receives. If a  
314 notification recipient detects an error, it MAY send a response before it has read the entire request. If the  
315 HTTP layer of the Notification Recipient completes processing the HTTP headers successfully, it MAY  
316 send an intermediate response, such as "100 Continue", with no notification data before sending the  
317 notification response. The '[ipp-ntfyipp-notify-send](#)' Notification Sources MUST expect such a variety of  
318 responses from notification recipients. For further information on HTTP/1.1, consult the HTTP documents  
319 [rfc2616].

320 An '[ipp-ntfyipp-notify-send](#)' Notification Recipient (server) MUST support chunking for HTTP notification  
321 requests, and an '[ipp-ntfyipp-notify-send](#)' Notification Source (client) MUST support chunking for HTTP  
322 notification responses according to HTTP/1.1[rfc2616]. Note: this rule causes a conflict with non-compliant  
323 implementations of HTTP/1.1 that don't support chunking for POST methods, and this rule may cause a  
324 conflict with non-compliant implementations of HTTP/1.1 that don't support chunking for CGI scripts

## 325 7 IANA Considerations

326 IANA will be asked to register this '[ipp-ntfyipp-notify-send](#)' notification delivery scheme and protocol and  
327 will be asked to assign a default port.

## 328 8 Internationalization Considerations

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

## 333 9 Security Considerations

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

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

### 342 9.1 Security Conformance

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

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

349 Notification Recipients MAY support TLS for client authentication, server authentication and operation  
 350 privacy. If a notification recipient supports TLS, it MUST support the  
 351 TLS\_DHE\_DSS\_WITH\_3DES\_EDE\_CBC\_SHA cipher suite as mandated by RFC 2246 [rfc2246]. All  
 352 other cipher suites are OPTIONAL. Notification recipients MAY support Basic Authentication (described  
 353 in HTTP/1.1 [rfc2616]) for client authentication if the channel is secure. TLS with the above mandated  
 354 cipher suite can provide such a secure channel.

## 355 10 References

356 [ipp-mod]

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

- 359 [ipp-ntfy]  
360 Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing  
361 Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-012.txt>, ~~October 14,~~  
362 ~~1999~~February 2, 2000.
- 363 [ipp-pro]  
364 Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and  
365 Transport", draft-ietf-ipp-protocol-v11-03.txt, June, 1999.
- 366 [rfc2026]  
367 S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.
- 368 [rfc2616]  
369 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext  
370 Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
- 371 [rfc2617]  
372 J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, "HTTP  
373 Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

## 374 11 Author's Addresses

375 Hugo Parra  
376 Novell, Inc.  
377 122 E 1700 S  
378 Provo, UT 84606  
379  
380 Phone: 801-861-3307  
381 Fax: 801-861-2517  
382 e-mail: hparra@novell.com  
383  
384 Tom Hastings  
385 Xerox Corporation  
386 737 Hawaii St. ESAE 231  
387 El Segundo, CA 90245  
388  
389 Phone: 310-333-6413  
390 Fax: 310-333-5514  
391 e-mail: hastings@cp10.es.xerox.com  
392

## 393 12 Full Copyright Statement

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

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

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

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