

1 INTERNET-DRAFT **5 ISSUES** are highlighted like this.
2 <draft-ietf-ipp-notify-get-poll-00.txt>

Carl-Uno Manros
Tom Hastings
Robert Herriot
Xerox Corp.
Harry Lewis
IBM, Corp.
February 2, 2000~~December 7, 1999~~

9 Internet Printing Protocol/1.1: **The 'ipp-notify-pollget' Notification Polling Method**

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

11 Status of this Memo

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

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

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

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

20 **Abstract**

21 The IPP notification specification [ipp-ntfy] is an OPTIONAL extension to IPP/1.0 and IPP/1.1 that
22 requires the definition of one or more delivery methods for dispatching event notification reports to
23 Notification Recipients. This document describes the semantics and syntax of the 'ipp-notify-pollget' event
24 notification delivery method. For this delivery method, the client uses an explicit IPP Get-Notifications
25 Printer operation in order to request (pull) event Notifications from the IPP Printer.

26 When a Printer supports the 'ipp-notify-poll' delivery method, it queues Notification events for a window of
27 time for each Subscription object. Notification Recipients poll these Subscription objects at the rate
28 specified by the time window. The Get-Notifications request indicates whether the client wants to receive
29 all ~~future-pending~~ events Notifications for (1) any Subscription for which the client is the owner or (2) a
30 particular Subscription object. The Get-Notifications operation retrieves all pending Notifications that
31 occurred for an interval of time in the past for ~~one or more specified subscription-ids~~ the requested
32 Subscription objects. The Printer returns the all pending Notifications along with two time intervals that
33 specify the next time window: one is the minimum interval that the client should wait before performing
34 another Get-Notifications on the subscription-id and the other is the maximum interval that the Printer is
35 guaranteed to keep any new Notifications associated with the subscription-id.

36 The Printer may keep the channel open if the minimum interval is sufficiently short, but in any case the
37 client performs a new Get-Notifications operation each time it wants more Notifications. Since the client
38 will be making Get-Notification requests before the time window expires, the Printer will, on occasion,
39 return the same event Notification in two successive responses. The later ones in the previous response will
40 become the earliest in the next response. The client is expected to filter out these duplicates which is easy
41 to do because of the sequence number in each Notification. ~~Each Get-Notifications request returns a new~~
42 set of Notifications, that is, it does not return any returned in previous responses. In either case, the event
43 Notifications are returned as MIME multi-part related responses to the Get-Notifications request. The
44 HTTP channel is kept open, so that subsequent event Notifications are returned using additional MIME
45 multi-part related responses.

46 The full set of IPP documents includes:

- 47 Design Goals for an Internet Printing Protocol [RFC2567]
- 48 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
- 49 Internet Printing Protocol/1.1: Model and Semantics [ipp-mod]
- 50 Internet Printing Protocol/1.1: Encoding and Transport [ipp-pro]
- 51 Internet Printing Protocol/1.1: Implementer's Guide [ipp-iig]
- 52 Mapping between LPD and IPP Protocols [RFC2569]
- 53 Internet Printing Protocol/1.0 & 1.1: Event Notification Specification [ipp-ntfy]

54

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

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

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

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

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

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

80 The "Event Notification Specification" document defines OPTIONAL operations that allow a client to
81 subscribe to printing related events. Subscriptions include "Per-Job subscriptions" and "Per-Printer
82 subscriptions". Subscriptions are modeled as Subscription objects. Four other operations are defined for
83 subscription objects: get attributes, get subscriptions, renew a subscription, and cancel a subscription.

84

85

Table of Contents

86	1	Introduction	5
87	2	Terminology	5
88	3	Model and Operation	6
89	4	Get-Notifications operation	7
90	4.1	GET-NOTIFICATIONS REQUEST	8
91	4.2	GET-NOTIFICATIONS RESPONSE	8
92	5	Extension to Print-Job, Print-URI, Create-Job, Create-Printer-Subscription and Create-Printer-Subscription	10
93			
94	5.1	RESPONSE	10
95	6	Encoding	10
96	7	IANA Considerations	11
97	8	Internationalization Considerations	11
98	9	Security Considerations	11
99	10	References	11
100	11	Author's Addresses	12
101	12	Full Copyright Statement	12
102			

103

104

1 Introduction

105 IPP printers that support the OPTIONAL IPP notification extension [ipp-ntfy] either a) accept, store, and
106 use notification subscriptions to generate event Notification reports and implement one or more delivery
107 methods for notifying interested parties, or b) support a subset of these tasks and farm out the remaining
108 tasks to a Notification Delivery Service. The ~~'ipp-get~~ipp-notify-poll' event notification delivery method
109 specified in this document defines a Get-Notifications operation that may be used in a variety of
110 notification scenarios. Its primary intended use is for clients that want to be Notification Recipients. ~~to~~
111 ~~explicitly request (pull) event Notifications from the IPP Printer upon request.~~ However, the Get-
112 Notifications operation may also be used by Notification Delivery Services ~~to request (pull) event~~
113 ~~Notifications from an IPP Printer~~ for subsequent distribution to the Ultimate Notification Recipients. ~~The~~
114 ~~HTTP channel is kept open, so that subsequent event Notifications are returned using additional MIME~~
115 ~~multi-part related responses.~~

116 When a Printer supports the 'ipp-notify-poll' delivery method, it queues Notification events for a window of
117 time for each Subscription object. Notification Recipients poll these Subscription objects at the rate
118 specified by the time window. The Get-Notifications request indicates whether the client wants to receive
119 all pending events Notifications for (1) any Subscription for which the client is the owner or (2) a particular
120 Subscription object. The Get-Notifications operation retrieves all pending Notifications that occurred for
121 an interval of time in the past for ~~for one or more specified subscription ids~~ the requested Subscription
122 objects. The Printer returns all pending Notifications along with two time intervals that specify the next
123 time window: one is the minimum interval that the client should wait before performing a Get-Notifications
124 on the subscription-id and the other is the maximum interval that the Printer is guaranteed to keep any new
125 Notifications associated with the subscription-id.

126 The Printer may keep the channel open if the minimum interval is sufficiently short, but in any case the
127 client performs a new Get-Notifications operation each time it wants more Notifications. Since the client
128 will be making Get-Notification requests before the time window expires, the Printer will, on occasion,
129 return the same event Notification in two successive responses. The later ones in the previous response will
130 become the earliest in the next response. The client is expected to filter out these duplicates which is easy
131 to do because of the sequence number in each Notification. The reason for not removing the Notifications
132 from the Subscription object with every Get-Notifications request, is so that multiple Notification
133 Recipients can be polling the same subscription object. This is useful if you are logged in to several
134 desktops at the same time and want to see the same events at both places.~~Each Get-Notifications request~~
135 ~~returns a new set of notifications, that is, it does not return any returned in previous responses.~~

136

2 Terminology

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

138 REQUIRED: if an implementation supports the extensions described in this document, it MUST
139 support a REQUIRED feature.

140 OPTIONAL: if an implementation supports the extensions described in this document, it MAY support
141 an OPTIONAL feature.
142 Notification Recipient - See [ipp-ntfy]
143 Subscription object - See [ipp-ntfy]
144 Ultimate Notification Recipient - See [ipp-ntfy]

145 3 Model and Operation

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

150 When creating each Subscription object, the client supplies the "notify-recipient" (uri) attribute. The
151 "notify-recipient" attribute specifies both a single Notification Recipient that is to receive the Notifications
152 when subsequent events occur and the method for Notification delivery that the IPP Printer is to use. For
153 the 'ipp-notify-poll' Notification delivery method defined in this document, there is no notify-recipient
154 because the Printer waits for one or more~~some~~ clients to ask for Notifications from a Subscription object
155 rather than sending them. ~~the notification method is 'ipp-get', and the Notification Recipient is omitted,~~
156 ~~since~~ Rather, any client that is authenticated (1) as an operator or administrator or (2) as the owner of the
157 Subscription object can initiate a Get-Notifications operation for that Subscription object. Therefore, any
158 Printer that supports the 'ipp-notify-poll' notification delivery method MUST queue event Notifications for
159 a sliding window of time for each Subscription object. Thus a single user can login at different places, say
160 his/her office, the lab, and/or several desktops in the same room, and receive the same event Notifications
161 from a single Subscription object.

162 ~~For the 'ipp-get' notification delivery method defined in the document, the client who created the~~
163 ~~Subscription objects is also the Notification Recipient.~~ The client issues a Get-Notifications Printer
164 operation in order to initiate the delivery of the ~~next event~~ pending Notifications ~~that occur~~ held by the
165 Printer for the ~~specified subscription id~~ Subscription objects requested. The client can indicate in the Get-
166 Notifications request whether it wants to receive all ~~future event~~ pending Notifications for (1) any existing
167 ~~or future~~ Subscription objects for which it is the owner or (2) ~~a~~ particular Subscription object(s) (for which
168 it MUST be the owner or have read-access rights). In either case, the Notifications are returned ~~as MIME~~
169 ~~multi-part related~~ in a responses to the Get-Notifications request.

170 If the client requests a persistent channel and if the Printer has returned minimum intervals that are
171 sufficiently short, then the Printer keeps the channel open. ~~The HTTP channel is kept open for an indefinite~~
172 ~~period, so that the IPP Printer continues to return additional parts of the MIME multi-part-related responses~~
173 ~~for each event Notification as it occurs.~~ Either the client or the IPP Printer can disconnect the HTTP
174 connection. ~~However, if the IPP Printer grants an HTTP connection it SHOULD disconnect only under~~
175 ~~unusual circumstances.~~

176 **ISSUE 01:** Should it be possible for a client to ask for the Per-Job Subscriptions for a particular job using a
177 "job-id", instead of the subscription-id, which currently isn't returned by a Job Creation operation?

178 ~~ISSUE 01: Is there a limit to the number of outstanding Get-Notifications requests that an IPP Printer~~
 179 ~~supports? What is this number? How does it relate to the maximum number of Subscriptions? Can the~~
 180 ~~client determine the number?~~

181 ~~ISSUE 02: Should an implementation be able to queue event Notifications, so that a client can get event~~
 182 ~~Notifications that had occurred prior to the Get-Notifications? If so, how long does the IPP Printer keep the~~
 183 ~~event Notifications before discarding them (for this delivery method only)? The lease time of the~~
 184 ~~Subscription object? If this is possible, should the subscriber get to say whether to queue or not, or is it just~~
 185 ~~baked into the implementation. If the former, does the subscriber indicate via a parameter in the~~
 186 ~~notification method URL? If the latter, how does a client discover whether event Notifications are queued~~
 187 ~~or not? Should we have two different notification methods, one the queues and one that doesn't?~~

188 4 Get-Notifications operation

189 This REQUIRED operation allows the client to request that ~~future event~~pending Notifications be delivered
 190 as ~~MIME multi-part related a~~ responses to this request. The client MUST be the owner or have write-
 191 access rights of the Subscription objects that are involved and the delivery method specified when the
 192 Subscription objects were created MUST be ~~'ipp-get'~~'ipp-notify-poll'~~unspecified~~. When the Printer creates a
 193 Subscription Object, either with a job-creation~~Job Creation~~ operation or with a Create-Printer-Subscription
 194 or Create-Job-Subscription operation and a subscription object contains a NULL~~the~~ 'ipp-notify-poll' value
 195 for the "notify-recipient" operation attribute, the Printer returns a minimum and maximum interval in the
 196 response. The client SHOULD perform a Get-Notifications operation after the minimum interval and if the
 197 Printer receives the Get-Notifications before the maximum interval has elapsed, it MUST have all of the
 198 Notifications that has occurred since the Subscription object was created. However, the client can and
 199 SHOULD issue the Get-Notifications request before having created any Subscription objects, in order not
 200 to miss any event Notifications.

201 **ISSUE 02: Is there anything useful that we could define for the rest of the "notification-recipient" (uri)**
 202 **attribute, since there is no recipient address needed after the 'ipp-notify-poll://' since the recipient(s) poll?**

203 The IPP Printer MUST accept the request in any state (see [ipp-mod] "printer-state" and "printer-state-
 204 reasons" attributes) and MUST remain in the same state with the same "printer-state-reasons".

Current "printer-state"	New "printer-state"	new "printer- state-reasons"	IPP Printer's response status code and action:
'idle'	'idle'	no-change	'successful-ok'
'processing'	'processing'	no-change	'successful-ok'
'stopped'	'stopped'	no-change	'successful-ok'

205 ~~ISSUE 03: What "printer-state-reasons" might cause an error return, if any? 'paused', 'shutdown',~~
 206 ~~'quiescent'?~~

207 *Access Rights:* The authenticated user (see [ipp-mod] section 8.3) performing this operation must either be
 208 the Subscription object owner (as determined when the Subscription object was created by the Job Creation
 209 operation, Create-Job-Subscription, or Create-Printer-Subscription operations) or an operator or
 210 administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5). Otherwise, the IPP object MUST

211 reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-
212 authorized' as appropriate.

213 4.1 Get-Notifications Request

214 The following groups of attributes are part of the Get-Notifications Request:

215 Group 1: Operation Attributes

216 Natural Language and Character Set:

217 The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod]
218 section 3.1.4.1.

219

220 Target:

221 The "printer-uri" (uri) operation attribute which is the target for this operation as described in [ipp-
222 mod] section 3.1.5.

223

224 Requesting User Name:

225 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as
226 described in [ipp-mod] section 8.3.

227

228 "subscription-ids" (1setOf integer(1:MAX)):

229 The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It
230 is an integer value that identifies one or more ~~the~~ Subscription objects for which event Notifications
231 are being requested. If the client supplies this attribute, but none of the Subscription objects is-are
232 not-found, the IPP Printer MUST return the 'client-error-not-found' status code. If some are found
233 and others are not, the ones that are not found are return in the Unsupported Attributes.

234

235 If the client does not supply this attribute, then the IPP Printer returns event Notifications for all
236 Subscription objects for which the client is the owner and the "notify-recipients" attribute is
237 NULL-ipp-get'ipp-notify-poll'. It is not an error if there are currently no Subscription objects for this
238 client; the response then contains no Notifications.~~the client can create Subscription objects later~~
239 ~~that will start returning event Notifications as responses to this operation.~~

240 4.2.2 Get-Notifications Response

241 The Printer object returns either an immediate error response or a successful response with status code:
242 'successful-ok' when the first event occurs, i.e., when the Printer delivers the first event Notification.

243 Group 1: Operation Attributes

244 Status Message:

245 In addition to the REQUIRED status code returned in every response, the response OPTIONALLY
246 includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation
247 attribute as described in [ipp-mod] sections 13 and 3.1.6.

248

249 Natural Language and Character Set:

250 The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod]
251 section 3.1.4.2.

252

253 "minimum-time-interval" (integer(0:MAX)):

254 The value of this attribute is the minimum number of seconds that SHOULD elapse before the client
255 performs this operation again for these subscription-ids. A client MAY perform this operation at any
256 time, and a Printer MUST respond with all pending Notifications. A client observes this value in
257 order to be a "good network citizen".

258

259 "maximum-time-interval" (integer(0:MAX)):

260 The value of this attribute is the maximum number of seconds that SHOULD elapse before this
261 client SHOULD issue a Printer receives this operation again for these subscription-ids. A Printer
262 MUST preserve all Notifications that occur for the number of seconds specified by this attribute
263 starting at the time it is sent in a response. A client MAY perform this operation at any time, and a
264 Printer MUST respond with all pending Notifications. If a Printer receives this operation after this
265 time interval, it may SHOULD have discarded some Notifications since the last response.

266

267 **ISSUE 04 - Or MUST the Printer discard events that occurred earlier than the sliding time window**
268 **specified by the difference between these two values? Otherwise, the clients may get back a lot of**
269 **duplicate events on subsequent requests.**

270

271

272 Group 2: Unsupported Attributes

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

274

275 If the "~~requested-attributes~~subscription-ids" attribute containeds subscription-ids that do not exist,
276 the Printer returns them they are returned in this group as value of the "subscription-ids~~requested-~~
277 attributes" attribute.

278

279 Group 3 through N: ~~Generic Object Attributes~~Notification Attributes

280 The Printer object responds with one event Notification per Group for each pending Notification
281 that meets the criteria specified by the subscription-ids attribute and requesting user name.(see [ipp-
282 ntfy]). ~~If there are multiple events that occur at the same time, the Printer object returns them in~~
283 ~~separate MIME multi-part related responses, each as separate IPP operation responses, as well. The~~
284 ~~HTTP channel is kept open for an indefinite period, so that the IPP Printer continues to return~~
285 ~~additional parts of the MIME multi-part related responses for each event Notification as it occurs.~~
286 ~~ISSUE 04 - Is this correct for MIME multi-part related responses? This need prototyping.~~

287 **5 Extension to Print-Job, Print-URI, Create-Job, Create-Printer-Subscription** 288 **and Create-Printer-Subscription**

289 5.1 Response

290 When Print-Job, Print-URI or Create-Job contains a “job-notify” attribute and the “notify-recipient” is 'ipp-
291 notify-poll'NULL, the response contains two additional Operation Attributes that pertain to subscriptions.

292 When Create-Job-Subscription or Create-Printer-Subscription operation contains a “notify-recipient” that is
293 'ipp-notify-poll'NULL, the response contains two additional Operation Attributes that pertain to
294 subscriptions.

295 Group 1: Operation Attributes

296 "minimum-time-interval" (integer(0:MAX)):

297 The value of this attribute is the minimum number of seconds that SHOULD elapse before the client
298 performs the Get-Notification operation for the first time with any subscription-ids returned with
299 this job. A client MAY perform the Get-Notification operation at any time, and a Printer MUST
300 respond with all pending Notifications. A client observes this value in order to be a “good network
301 citizen”.

303 ISSUE 05: if we don't want to have Job Creation operations return subscription id's, then allow a
304 "job-ids" operation attribute in the Get-Notifications request in addition to the "subscription-ids"
305 operation attribute.

307 "maximum-time-interval" (integer(0:MAX)):

308 The value of this attribute is the maximum number of seconds that SHOULD elapse before a Printer
309 receives the Get-Notification operation for the first time with any subscription-ids returned with this
310 job. A Printer MUST preserve all Notifications that occur for the number of seconds specified by
311 this attribute starting at the time it is sent in a response. A client MAY perform the Get-Notification
312 operation at any time, and a Printer MUST respond with all pending Notifications. If a Printer
313 receives a Get-Notification operation after this time interval, it may have discarded some
314 Notifications since the last response.

315

316 **6 Encoding**

317 The operation-id assigned for the Get-Notification operation is:

318 0x00??

319 and should be added to the next version of [ipp-mod] section 4.4.15 "operations-supported".

320 This notification delivery method uses the IPP transport and encoding [ipp-pro] for the Get-Notifications
321 operation with one extension:

322 Instead of defining a new object attribute tag, a Generic Object attributes tag is defined that is used
323 for all new objects, such as Subscription objects, etc. Then this one new tag can also be used for the
324 Get-Notifications response Group 3 tag and subsequent groups in section 4.2:

325 ~~generic-object~~notification-attributes-tag = %x07?? ; tag of 7?

326 7 IANA Considerations

327 IANA will be asked to register this 'ipp-getipp-notify-poll' notification delivery scheme.

328 8 Internationalization Considerations

329 With the 'ipp-getipp-notify-poll' method defined in this document, the client cannot request the Human
330 Consumable form by supplying the "notify-text-format" operation attribute (see [ipp-ntfy]). Therefore, the
331 IPP Printer does not have to perform any localization with this notification delivery method. However, the
332 client when it receives the Get-Notifications response is expected to localize the attributes that have the
333 'keyword' attribute syntax according to the charset and natural language requested in the Get-Notifications
334 request.

335 9 Security Considerations

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

341 Unlike other event Notification delivery methods in which the IPP Printer initiates the event Notification,
342 with the method defined in this document, the Notification Recipient is the client who issues the Get-
343 Notifications operation. Therefore, there is no chance of "spam" notifications with this method.
344 Furthermore, such a client can close down the HTTP channel at any time, and so can avoid future unwanted
345 event Notifications at any time.

346 10 References

347 [ipp-mod]

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

350 [ipp-ntfy]

351 Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing
352 Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-01.txt>, October 14,
353 1999.

354 [ipp-pro]

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

357 [rfc2026]

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

359 [RFC2616]

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

362 11 Author's Addresses

363 Robert Herriot

364 3400 Hillview Ave., Bldg #1

365 Palo Alto, CA 94304

366

367 Phone: 650-813-7696

368 Fax: 650-813-6860

369 Email: rherriot@pahv.xerox.com

370 ~~Carl Uno Manros~~

371 ~~Xerox Corporation~~

372 ~~737 Hawaii St. ESAE 231~~

373 ~~El Segundo, CA 90245~~

374

375 ~~Phone: 310-333-8273~~

376 ~~Fax: 310-333-5514~~

377 ~~e-mail: emanros@ep10.es.xerox.com~~

378

379 ~~Tom Hastings~~

380 ~~Xerox Corporation~~

381 ~~737 Hawaii St. ESAE 231~~

382 ~~El Segundo, CA 90245~~

383

384 ~~Phone: 310-333-6413~~

385 ~~Fax: 310-333-5514~~

386 ~~e-mail: hastings@ep10.es.xerox.com~~

387

388 12 Full Copyright Statement

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

390 This document and translations of it may be copied and furnished to others, and derivative works that
391 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and

392 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
393 this paragraph are included on all such copies and derivative works. However, this document itself may not
394 be modified in any way, such as by removing the copyright notice or references to the Internet Society or
395 other Internet organizations, except as needed for the purpose of developing Internet standards in which
396 case the procedures for copyrights defined in the Internet Standards process must be followed, or as
397 required to translate it into languages other than English.

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

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