

Network Working Group
Request for Comments: 3996
Updates: 2911
Category: Standards Track

R. Herriot
Global Workflow Solutions
T. Hastings
Xerox Corp.
H. Lewis
IBM Corp.
March 2005

Internet Printing Protocol (IPP):
The 'ippget' Delivery Method for Event Notifications

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This document describes an extension to the Internet Printing Protocol 1.1: Model and Semantics (RFC 2911, RFC 2910). This document specifies the 'ippget' Pull Delivery Method for use with the "Internet Printing Protocol (IPP): Event Notifications and Subscriptions" specification (RFC 3995). This IPPGET Delivery Method is REQUIRED for all clients and Printers that support RFC 3995. The Notification Recipient, acting as a client, fetches (pulls) Event Notifications by using the Get-Notifications operation defined in this document.

Table of Contents

1.	Introduction	3
2.	Terminology	3
	2.1. Conformance Terminology	4
	2.2. Other Terminology	4
3.	Model and Operation	4
4.	General Information	5
5.	Get-Notifications Operation	7
	5.1. Get-Notifications Request	8
	5.1.1. notify-subscription-ids (1setOf integer(1:MAX))	8
	5.1.2. notify-sequence-numbers (1setOf integer(1:MAX))	9

- 5.1.3. notify-wait (boolean) 10
- 5.2. Get-Notifications Response. 10
 - 5.2.1. notify-get-interval (integer(0:MAX)). 13
 - 5.2.2. printer-up-time (integer(1:MAX)). 14
- 6. Additional Information about Subscription Template Attributes 17
 - 6.1. notify-pull-method (type2 keyword). 17
- 7. Subscription Description Attributes 18
- 8. Additional Printer Description Attributes 18
 - 8.1. ippget-event-life (integer(15:MAX)) 18
- 9. New Values for Existing Printer Description Attributes. 19
 - 9.1. notify-pull-method-supported (1setOf type2 keyword) . . . 19
 - 9.2. operations-supported (1setOf type2 enum). 19
- 10. New Status Codes. 19
 - 10.1. successful-ok-events-complete (0x0007) 20
- 11. Encoding and Transport. 20
- 12. Conformance Requirements. 21
 - 12.1. Conformance for IPP Printers 21
 - 12.2. Conformance for IPP Clients. 22
- 13. Normative References. 23
- 14. Informative References. 23
- 15. IANA Considerations 24
 - 15.1. Attribute Registrations. 24
 - 15.2. Delivery Method and Additional Keyword Attribute Value
registrations for Existing Attributes. 24
 - 15.3. Additional Enum Attribute Values 25
 - 15.4. Operation Registrations. 25
 - 15.5. Status Code Registrations. 25
- 16. Internationalization Considerations 25
- 17. Security Considerations 26
 - 17.1. Notification Recipient Client Access Rights. 26
 - 17.2. Printer Security Threats 27
 - 17.3. Notification Recipient Security Threats. 27
 - 17.4. Security Requirements for Printers 27
 - 17.5. Security Requirements for clients. 28
- 18. Description of Base IPP Documents (Informative) 28
- 19. Contributors. 29
- Authors' Addresses 30
- Full Copyright Statement 31

Table of Tables

- Table 1. Information about the Delivery Method. 5
- Table 2. Combinations of "notify-wait", "status-code", and
"notify-get-interval". 13
- Table 3. Attributes in Event Notification Content 15
- Table 4. Additional Attributes in Event Notification Content for
Job Events 16

Table 5. Combinations of Events and Subscribed Events for "job-
impressions-completed" 17

Table 6. Additional Attributes in Event Notification Content for
Printer Events 17

Table 7. Operation-id Assignments 19

Table 8. The "event-notification-attributes-tag" Value. 21

1. Introduction

This document describes an extension to the Internet Printing Protocol/1.1: Model and Semantics [RFC 2911], [RFC 2910]. This document specifies the 'ippget' Pull Delivery Method for use with the "Internet Printing Protocol (IPP): Event Notifications and Subscriptions" specification [RFC3995]. This IPPGET Delivery Method is REQUIRED for all clients and Printers that support [RFC3995]. The Notification Recipient, acting as a client, fetches (pulls) Event Notifications by using the Get-Notifications operation defined in this document. For a description of the base IPP documents, see section 21 of this document. For a description of the IPP Event Notification Model, see [RFC3995].

With this Pull Delivery Method, when an Event occurs, the Printer saves the Event Notification for a period of time called the Event Life. The Notification Recipient fetches (pulls) the Event Notifications by using the Get-Notifications operation. This operation causes the Printer to return all Event Notifications held for the specified Subscription object(s). If the Notification Recipient has selected the Event Wait Mode option to wait for additional Event Notifications, the Printer MAY continue to return Event Notifications to the Notification Recipient as asynchronous Get-Notification responses as Events occur using the transaction originated by the Notification Recipient.

The Notification Recipient can terminate Event Wait Mode (without closing the connection) by supplying the "notify-wait" (boolean) attribute with a 'false' value in a subsequent Get-Notifications request. Similarly, the Printer can terminate Event Wait Mode (without closing the connection) by returning the "notify-get-interval" (integer) operation attribute in a Get-Notifications response that tells the Notification Recipient how long to wait before trying again.

2. Terminology

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

2.1. Conformance Terminology

Capitalized terms such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED NOT, and OPTIONAL have special meaning relating to conformance as defined in [RFC2119] and [RFC2911], section 12.1. If an implementation supports the extension defined in this document, then these terms apply; otherwise, they do not. These terms define conformance to this document only; they do not affect conformance to other documents, unless it is explicitly stated otherwise.

2.2. Other terminology

This document uses the same terminology as [RFC2911], including "client", "Printer", "Job", "attribute", "attribute value", "keyword", "operation", "request", "response", and "support", with the same meanings. This document also uses terminology defined in [RFC3995], such as "Subscription (object)", "Notification Recipient", "Event", "Event Notification", "Compound Event Notification", "Event Life", and "Event Notification Attribute Group", with the same meanings. In addition, this document defines the following terms for use in this document:

Event Wait Mode: The mode requested by a Notification Recipient client in its Get-Notifications Request and granted by a Printer to keep the connection open while the Printer sends subsequent Get-Notification operation responses to the Notification Recipient in the form of Event Notifications as they occur.

3. Model and Operation

In a Subscription Creation Operation, when the "notify-pull-method" attribute is present and has the "ippget" keyword value, the client is requesting that the Printer use the "ippget" Pull Delivery Method for the Event Notifications associated with the new Subscription Object.

When an Event occurs, the Printer MUST generate an Event Notification and MUST assign it the Event Life. The Printer MUST hold an Event Notification for its assigned Event Life.

When a Notification Recipient wants to receive Event Notifications for a Subscription object, it performs the Get-Notifications operation supplying the Subscription object's subscription-id, which causes the Printer to return all un-expired Event Notifications held for that Subscription object. If the Notification Recipient has selected the Event Wait Mode option to wait for additional Event Notifications, the response to the Get-Notifications request continues indefinitely as the Printer continues to send Event

Notifications in the response as Events occur for that Subscription object.

When the Notification Recipient requests Event Notifications for Per-Job Subscription Objects, the Notification Recipient typically performs the Get-Notifications operation within a second of performing the Subscription Creation operation. Because the Printer MUST save Event Notifications for at least 15 seconds (see section 8.1), the Notification Recipient is unlikely to miss any Event Notifications that occur between the Subscription Creation and the Get-Notifications operation.

The 'ippget' Delivery Method is designed primarily for (1) a client that wants to get Events (from the job's Per-Job Subscription object) for a job that it has submitted and (2) a privileged client that wants to get all job or printer Events from a Per-Printer Subscription object.

4. General Information

If a Printer supports this Delivery Method, the following are its characteristics.

Table 1. Information about the Delivery Method

Document Method Conformance Requirement Realization	Delivery Method
1. What is the URL scheme name for the Push Delivery Method, or the keyword method name for the Pull Delivery Method?	'ippget' keyword method name
2. Is the Delivery Method REQUIRED, RECOMMENDED, or OPTIONAL for an IPP Printer to support?	REQUIRED
3. What transport and delivery protocols does the Printer use to deliver the Event Notification Content; i.e., what is the entire network stack?	IPP with one new operation.
4. Can several Event Notifications be combined into a Compound Event Notification?	Yes.

- | | |
|---|--|
| 5. Is the Delivery Method initiated by the Notification Recipient (pull), or by the Printer (push)? | This Delivery Method is a pull method with aspects of a push method, though the Printer does not initiate the operation. |
| 6. Is the Event Notification content Machine Consumable or Human Consumable? | Machine Consumable. |
| 7. What section in this document answers the following questions? For a Machine Consumable Event Notification, what is the representation and encoding of values defined in section 9.1 of [RFC3995], and what are the conformance requirements thereof? For a Human Consumable Event Notification, what is the representation and encoding of pieces of information defined in section 9.2 of [RFC3995], and the conformance requirements thereof? | Section 5. |
| 8. What are the latency and reliability of the transport and delivery protocol? | Same as IPP and the underlying HTTP transport. |
| 9. What are the security aspects of the transport and delivery protocol; e.g., how it is handled in firewalls? | Same as IPP and the underlying HTTP transport and in the same direction, so no new firewall considerations. |
| 10. What are the content length restrictions? | None. |
| 11. What are the additional values or pieces of information that a Printer sends in an Event Notification content and the conformance requirements thereof? | None. |

- | | |
|--|---|
| 12. What are the additional Subscription Template and/or Subscription Description attributes and the conformance requirements thereof? | None. |
| 13. What are the additional Printer Description attributes and the conformance requirements thereof? | "ipp-event-life"
(integer (15: MAX)) |

5. Get-Notifications Operation

This operation is issued by a client acting in the role of a Notification Recipient requesting the Printer to return all Event Notifications held for the identified Subscription object(s).

A Printer MUST support this operation, MUST accept the request in any state (see [RFC2911] "printer-state" and "printer-state-reasons" attributes), and MUST remain in the same state with the same "printer-state-reasons" values.

When a Printer performs this operation, it MUST return all and only those Event Notifications

1. whose associated Subscription Object's "notify-subscription-id" Subscription Description attribute equals one of the values of the "notify-subscription-ids" (1setOf integer(1:MAX)) operation attribute AND
2. whose associated Subscription Object contains the "notify-pull-method" attribute and it has the 'ippget' keyword value, AND
3. whose "notify-sequence-number" is equal to or greater than the corresponding value of the "notify-sequence-numbers" (1setOf integer(1:MAX)) operation attribute if supplied AND
4. whose Event Life has not yet expired AND
5. where the Notification Recipient client has read-access rights to the identified Subscription Object (see Access Rights paragraph below).

The Notification Recipient client MUST either (a) request Event Wait Mode by supplying the "notify-wait" operation attribute with a 'true' value or (b) suppress Event Wait Mode by omitting the "notify-wait" operation attribute or by supplying it with a 'false' value. To terminate Event Wait Mode subsequently, the Notification Recipient client MUST close the connection. To terminate Event Wait Mode, the Printer MUST either (a) return the "notify-get-interval" operation

attribute in a Get-Notifications response (RECOMMENDED behavior) or (b) close the connection. The "notify-get-interval" operation attributes tell the Notification Recipient how long to wait before trying a subsequent Get-Notifications request.

Access Rights: The authenticated user (see [RFC2911], section 8.3) performing this operation MUST be (1) the owner of each Subscription Object identified by the "notify-subscription-ids" operation attribute (see section 5.1.1), (2) an operator or administrator of the Printer (see [RFC2911], sections 1 and 8.5), or (3) otherwise authorized by the Printer's administrator-configured security policy to request Event Notifications from the target Subscription Object(s). Otherwise, the IPP Printer MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' status code, as appropriate. Furthermore, the Printer's security policy MAY limit the attributes returned by the Get-Notifications operation, in a manner similar to that of the Get-Job-Attributes operation (see [RFC2911], end of section 3.3.4.2).

5.1. Get-Notifications Request

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

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes, as described in [RFC2911], section 3.1.4.1.

Target:

The "printer-uri" (uri) operation attribute that is the target for this operation as described in [RFC2911], section 3.1.5.

Requesting User Name:

The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as described in [RFC2911], section 8.3.

5.1.1. notify-subscription-ids (lsetOf integer(1:MAX))

This attribute identifies one or more Subscription objects for which Events are requested. The client MUST supply this attribute with at least one value. The Printer object MUST support this attribute with multiple values.

If no Subscription Object exists with the supplied identifier, or if the identified Subscription Object does not contain the "notify-

pull-method" attribute with the 'ippget' keyword value, the Printer MUST return the 'client-error-not-found' status code.

Note: The name of both the "notify-subscription-ids" and "notify-sequence-numbers" end in 's', as they are multi-valued. However, there are other occurrences of these attribute names without the 's' that are single valued.

5.1.2. notify-sequence-numbers (1setOf integer(1:MAX))

This attribute specifies one or more of the lowest Event Notification sequence number values for the Subscription objects identified by the corresponding values of the "notify-subscription-ids" operation attribute. The Notification Recipient SHOULD supply this attribute, and the number of values SHOULD be the same as that of the "notify-subscriptions-ids" attribute. The Printer MUST support this attribute with multiple values.

The Printer MUST NOT return Notification Events with lower sequence numbers for the corresponding Subscription object. Therefore, by supplying the proper values for this attribute the Notification Recipient can prevent getting the same Event Notifications from a Subscription object that were returned on a previous Get-Notifications request. The Notification Recipient SHOULD remember the highest "notify-sequence-number" value returned for each Subscription object requested and SHOULD pass that value for each requested Subscription object on the next Get-Notifications request.

If the Notification Recipient supplies fewer values for this attribute (including omitting this attribute) than it does for the "notify-subscription-ids" operation attribute, the Printer assumes a '1' value for each missing value. A value of '1' causes the Printer to return any un-expired Event Notification for that Subscription object, as '1' is the lowest possible sequence number. If the Notification Recipient supplies more values for this attribute than the number of values for the "notify-subscription-ids" operation attribute, the Printer ignores the extra values.

Note: If a Notification Recipient performs two consecutive Get-Notifications operations with the same value for "notify-sequence-number" (or omits the attribute), the time stamp value of the first Event Notification in the second Get-Notifications Response may be less than that of the time stamp of the last Event Notification in the first Get-Notification Response. This happens because the Printer sends all unexpired Event Notifications with an equal or higher sequence number according to the ordering specified in [RFC3995], and some Event Notifications from the first Get-

Notifications operation may not have expired by the time the second Get-Notifications operation occurs.

5.1.3. notify-wait (boolean)

This value indicates whether the Notification Recipient wants Event Wait Mode. The client MAY supply this attribute. The Printer object MUST support both values of this attribute.

If the client supplies the 'false' value or omits this attribute, the client is not requesting Event Wait Mode. If the value is 'true', the client is requesting Event Wait Mode. See the beginning of section 5.2 for the rules for Event Wait Mode.

5.2. Get-Notifications Response

The Printer has the following options for responding to a Get-Notifications Request:

1. The Printer can reject the request and return the 'server-error-busy' status code if the Printer is too busy to accept this operation at this time. In this case, the Printer MUST return the "get-notify-interval" operation attribute to indicate when the client SHOULD try again.
2. If the Notification Recipient did not request Event Wait Mode ("notify-wait-mode" = 'false' or omitted), the Printer MUST immediately return whatever Event Notifications it currently holds in the requested Subscription object(s) and MUST return the "notify-get-interval" operation attribute with the number of seconds from now, at which the Notification Recipient SHOULD repeat the Get-Notifications Request to get future Event Notifications.
3. If the Notification Recipient requested Event Wait Mode ("notify-wait-mode" = 'true'), the Printer MUST immediately return whatever Event Notifications it currently holds in the requested Subscription object(s) and MUST continue to return Event Notifications as they occur until all the requested Subscription Objects are canceled. A Subscription Object is canceled either via the Cancel-Subscription operation or by the Printer (e.g., the Subscription Object is canceled when the associated Job completes and is no longer in the Job Retention or Job History phase; see the "ippget-event-life (integer(15:MAX))" attribute discussion in section 8.1).

However, the Printer MAY decide to terminate Event Wait Mode at any time, including in the first response. In this case, the

Printer MUST return the "notify-get-interval" operation attribute. This attribute indicates that the Printer wishes to leave Event Wait Mode and the number of seconds in the future that the Notification Recipient SHOULD try the Get-Notifications operation again. The Notification Recipient MUST accept this response and MUST disconnect. If the Notification Recipient does not disconnect, the Printer SHOULD do so.

From the Notification Recipient's view, the response appears as an initial burst of data, which includes the Operation Attributes Group and one Event Notification Attributes Group per Event Notification that the Printer is holding. After the initial burst of data, if the Notification Recipient has selected the Event Wait Mode option to wait for additional Event Notifications, the Notification Recipient receives occasional Event Notification Attribute Groups. Proxy servers may delay some Event Notifications or cause time-outs to occur. The client MUST be prepared to perform the Get-Notifications operation again when time-outs occur.

Each attribute is encoded by using the IPP rules for encoding attributes [RFC2910] and MAY be encoded in any order. Note: the Get-Jobs response in [RFC2911] acts as a model for encoding multiple groups of attributes. See section 11 for the encoding and transport rules.

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

Group 1: Operation Attributes

Status Message: In addition to the REQUIRED status code returned in every response, the response OPTIONALLY includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation attribute, as described in [RFC2911], sections 13 and 3.1.6.

The Printer can return any status codes defined in [RFC2911]. If the status code is not 'successful-xxx', the Printer MUST NOT return any Event Notification Attribute groups. The following are descriptions of the important status codes:

successful-ok: The response contains all Event Notification associated with the specified subscription-ids that had been supplied in the "notify-subscription-ids" operation attribute in the request. If the requested Subscription Objects have no associated Event Notification, the response MUST contain zero Event Notifications.

successful-ok-events-complete: Indicates when this return is the last return for all Subscription objects that match the request, whether or not Event Notifications are returned. This condition occurs for Event Wait Mode with Notification Recipients waiting for responses when (1) the Subscription Object is canceled with a Cancel-Subscription operation, (2) the Subscription Object is deleted, when the Per-Printer Subscription lease time expires, or (3) the 'job-completed' event occurs for a Per-Job Subscription. This condition also occurs for a Get-Notifications request that a Notification Recipient makes after the job completes, but before the Event Life expires. See section 10.1.

client-error-not-found: The Printer has no Subscription Objects whose "notify-subscription-id" attribute equals any of the values of the "notify-subscription-ids" operation attribute supplied, or the identified Subscription Object does not contain the "notify-pull-method" attribute with the 'ippget' keyword value.

server-error-busy: The Printer is too busy to accept this operation. The Printer SHOULD return the "notify-get-interval" operation attribute in the Operation Attributes of the response; then the Notification Recipient SHOULD wait for the number of seconds specified by the "notify-get-interval" operation attribute before performing this operation again. If the "notify-get-interval" Operation Attribute is not present, the Notification Recipient SHOULD use the normal network back-off algorithms to determine when to perform this operation again.

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes, as described in [RFC2911], section 3.1.4.2.

The Printer MUST use the values of "notify-charset" and "notify-natural-language", respectively, from one Subscription Object associated with the Event Notifications in this response.

Normally, there is only one matched Subscription Object, or the value of the "notify-charset" and "notify-natural-language" attributes is the same in all Subscription Objects. If not, the Printer MUST pick one Subscription Object from which to obtain the value of these attributes. The algorithm for picking the Subscription Object is implementation dependent. The choice of natural language is not critical, because 'text' and 'name' values can override the "attributes-natural-language" operation attribute. The Printer's choice of charset

is critical because a bad choice may leave it unable to send some 'text' and 'name' values accurately.

5.2.1. notify-get-interval (integer(0:MAX))

The value of this operation attribute is the number of seconds that the Notification Recipient SHOULD wait before trying the Get-Notifications operation again. The Printer MUST return this operation attribute if (1) it is too busy to return events, (2) the Notification Recipient client did not request Event Wait Mode, or (3) the Printer is terminating Event Wait Mode. The client MUST accept this attribute and SHOULD reissue the Get-Notifications operation (with or without "notify-wait" = 'true') at the indicated number of seconds in the future in order to get more Event Notifications This value is intended to help the client be a good network citizen.

The value of this attribute MUST be at least as large as that of the Printer's "ippget-event-life" Printer Description attribute (see section 8.1). The Printer MAY return a value that is larger than that of the "ippget-event-life" Printer Description attribute provided that the Printer increases the Event Life for this Subscription object so that Notification Recipients taking account of the larger value and polling with a longer interval will not miss events. Note: Implementing such an algorithm requires some hidden attributes in the Subscription object that are IMPLEMENTATION DEPENDENT.

If the Printer wants to remain in Event Wait Mode, then the Printer MUST NOT return this attribute in the response.

Here is a complete table of combinations of "notify-wait", "status-code", "notify-get-interval", and Event Notification Attributes Groups for Get-Notification initial (Wait and No Wait) Responses and subsequent Event Wait Mode Responses (which may stay in Event Wait Mode or may request the Notification Recipient to leave Event Wait Mode):

Table 2. Combinations of "notify-wait", "status-code", and "notify-get-interval"

Client sends:	Printer returns:	Printer returns:	Event Notification Attribute Groups
"notify-wait"	"status-code"	"notify-get-interval"	
1. 'false'*	'successful-ok'	MUST return N	maybe
2. 'false'*	'not-found'	MUST NOT	MUST NOT
3. 'false'*	'busy'	MUST return N	MUST NOT

- | | | | | |
|----|----------|-----------------------|---------------|---------------------------------------|
| 4. | 'false'* | 'events-
complete' | MUST NOT | 'job-
completed' |
| 5. | 'true' | 'successful-ok' | MUST NOT | MUST |
| 6. | 'true' | 'successful-ok' | MUST return N | maybe |
| 7. | 'true' | 'not-found' | MUST NOT | MUST NOT |
| 8. | 'true' | 'busy' | MUST return N | MUST NOT |
| 9. | 'true' | 'events-
complete' | MUST NOT | 'job-
completed' or
maybe other |
- * 'false' or client omits the "notify-wait" attribute.

Explanation:

- 1-4: Client does not request Event Wait Mode.
 5-9: Client requests Event Wait Mode.
 2,7: Subscription object not found, or was canceled earlier; client should NOT try again.
 3,8: Server busy, tells client to try later; client should try again in N seconds.
 4: Client polled after job completed, but before Event Life expired, and got the 'job-completed' event, so the client shouldn't bother trying again; client should NOT try again later.
 5: Printer returns one or more Event Notifications and is OK to stay in Event Wait Mode; the client waits for more Event Notifications to be returned.
 6: Printer wants to leave Event Wait mode. Can happen on the first response (with or without Event Notifications) or happen on a subsequent response with or without Event Notifications; the client SHOULD try again in N seconds.
 9: Either (1) the printer returns 'job-completed' event, or (2) the Subscription Object was canceled by either a Cancel-Job or a Per-Printer Subscription expired without being renewed. For case (1), at least one Event Notification MUST be returned; for case (2), it is unlikely that any Event Notifications are returned, and the client should NOT try again.

5.2.2. printer-up-time (integer(1:MAX))

The value of this attribute is the Printer's "printer-up-time" attribute at the time when the Printer sends this response. The Printer MUST return this attribute. Because each Event Notification also contains the value of this attribute when the event occurred, the value of this attribute lets a Notification Recipient know when each Event Notification occurred relative to the time of this response.

Group 2: Unsupported Attributes

See [RFC2911], section 3.1.7, for details on returning Unsupported Attributes.

Group 3 through N: Event Notification Attributes

The Printer responds with one Event Notification Attributes Group per matched Event Notification. The entire response is considered a single Compound Event Notification (see [RFC3995]). The matched Event Notifications are all un-expired Event Notifications associated with the matched Subscription Objects and MUST follow the "Event Notification Ordering" requirements for Event Notifications within a Compound Event Notification specified in [RFC3995] section 9. In other words, the Printer MUST order these Event Notification groups in ascending time stamp (and sequence number) order for a Subscription object. If Event Notifications for multiple Subscription objects are being returned, the Notification Events for the next Subscription object follow in ascending time stamp order, etc.

Each Event Notification Group MUST contain all of attributes specified in section 9.1 ("Content of Machine Consumable Event Notifications") of [RFC3995], with exceptions denoted by asterisks in the tables below.

The tables below are identical to those in section 9.1 ("Content of Machine Consumable Event Notifications") of [RFC3995], except that each cell in the "Sends" column is a "MUST".

If more than one Event Notification is being returned and the status of each is not the same, then the Printer MUST return a "notify-status-code" attribute in each Event Notification Attributes group to indicate the differing status values.

For an Event Notification for all Events, the Printer includes the attributes shown in Table 3.

Table 3. Attributes in Event Notification Content

Source Value	Sends	Source Object
notify-subscription-id (integer(1:MAX))	MUST	Subscription
notify-printer-uri (uri)	MUST	Subscription
notify-subscribed-event (type2 keyword)	MUST	Event Notification
printer-up-time (integer(1:MAX)) *	MUST	Printer
printer-current-time (dateTime)	MUST **	Printer

notify-sequence-number (integer (0:MAX))	MUST	Subscription
notify-charset (charset)	MUST	Subscription
notify-natural-language (naturalLanguage)	MUST	Subscription
notify-user-data (octetString(63))	MUST ***	Subscription
notify-text (text)	MUST	Event Notification
attributes from the "notify-attributes" attribute	MUST ****	Printer
attributes from the "notify-attributes" attribute	MUST ****	Job
attributes from the "notify-attributes" attribute	MUST ****	Subscription

* As specified in [RFC3995] section 9, the value of the "printer-up-time" attribute sent in each Event Notification MUST be the time at which the Event occurred, not the time at which the Event Notification was sent.

** The Printer MUST send the "printer-current-time" attribute if and only if it supports the "printer-current-time" attribute on the Printer object.

*** If the associated Subscription Object does not contain a "notify-user-data" attribute, the Printer MUST send an octet-string of length 0.

**** If the "notify-attributes" attribute is present on the Subscription Object, the Printer MUST send all attributes specified by the "notify-attributes" attribute. Note: If the Printer doesn't support the "notify-attributes" attribute, it is not present on the associated Subscription Object.

For Event Notifications for Job Events, the Printer includes the additional attributes shown in Table 4.

Table 4. Additional Attributes in Event Notification Content for Job Events

Source Value	Sends	Source Object
job-id (integer(1:MAX))	MUST	Job
job-state (type1 enum)	MUST	Job
job-state-reasons (1setOf type2 keyword)	MUST	Job
job-impressions-completed (integer(0:MAX))	MUST *	Job

* The Printer MUST send the "job-impressions-completed" attribute in an Event Notification only for the combinations of Events and Subscribed Events shown in Table 5.

Table 5. Combinations of Events and Subscribed Events for "job-impressions-completed"

Job Event	Subscribed Job Event
'job-progress'	'job-progress'
'job-completed'	'job-completed'
'job-completed'	'job-state-changed'

For Event Notification for Printer Events, the Printer includes the additional attributes shown in Table 6.

Table 6. Additional Attributes in Event Notification Content for Printer Events

Source Value Object	Sends	Source
printer-state (type1 enum)	MUST	Printer
printer-state-reasons (1setOf type2 keyword)	MUST	Printer
printer-is-accepting-jobs (boolean)	MUST	Printer

6. Additional Information about Subscription Template Attributes

The 'ippget' Delivery Method does not define any addition Subscription Template attributes and has the conformance requirements for Subscription Template attributes defined in [RFC3995]. This section defines additional information about Subscription Template attributes defined in [RFC3995].

6.1. notify-pull-method (type2 keyword)

This Subscription Template attribute identifies the Pull Delivery Method to be used for the Subscription Object (see [RFC3995]). To support the 'ippget' Pull Delivery Method defined in this document, the Printer MUST support this attribute with the following keyword value:

'ippget': Indicates that the 'ippget' Pull Delivery Method is to be used for this Subscription Object.

7. Subscription Description Attributes

The 'ippget' Delivery Method has the conformance requirements for Subscription Description attributes defined in [RFC3995]. The 'ippget' Delivery Method does not define any addition Subscription Description attributes.

8. Additional Printer Description Attributes

This section defines additional Printer Description attributes for use with the 'ippget' Delivery Method.

8.1. ippget-event-life (integer(15:MAX))

This Printer Description attribute specifies the Event Life value that the Printer assigns to each Event; i.e., the number of seconds after an Event occurs during which a Printer will return that Event in an Event Notification in a Get-Notifications response. After the Event Life expires for the Event, the Printer MAY no longer return an Event Notification for that Event in a Get-Notifications response.

The Printer MUST support this attribute if it supports the 'ippget' Delivery Method. The value MUST be 15 or more (at least 15 seconds), and 60 (seconds) is the RECOMMENDED value to align with the PWG Job Monitoring MIB [RFC2707] jmGeneralJobPersistence and jmGeneralAttributePersistence objects.

For example, assume the following:

1. A client performs a Job Creation operation that creates a Subscription Object associated with the 'ippget' Delivery Method;
2. An Event associated with the new Job occurs immediately after the Subscription Object is created;
3. the same client or some other client performs a Get-Notifications operation so that the client is connected N seconds after the Job Creation operation.

Then, if N is less than the value of this attribute, the client(s) performing the Get-Notifications operations can expect not to miss any Event-Notifications, barring some unforeseen lack of memory space in the Printer. Note: The client MUST initiate the Get-Notifications at a time that is sufficiently less than N seconds to account for network latency so that it is connected to the Printer before N seconds elapses.

If a Printer supports the 'ippget' Delivery Method, it MUST keep 'completed', 'canceled', or 'aborted' Job objects in the Job Retention and/or Job History phases for at least as long as this attribute's value. The Printer MAY retain jobs longer than this value. See [RFC2911], section 4.3.7.1, and the discussion in [RFC3995] (regarding the 'job-completed' event). The latter explains that a Notification Recipient can query the Job after receiving a 'job-completed' Event Notification in order to find out other information about the job that is 'completed', 'aborted', or 'canceled'. However, this attribute has no effect on the Cancel-Subscription operation, which deletes the Subscription object immediately whether or not it contains the "notify-pull-method" attribute with the 'ippget' keyword value. Immediately thereafter, subsequent Get-Notifications Responses MUST NOT contain Event Notifications associated with the canceled Subscription object.

9. New Values for Existing Printer Description Attributes

This section defines additional values for existing Printer Description attributes as defined in [RFC3995].

9.1. notify-pull-method-supported (1setOf type2 keyword)

The following keyword value for the "notify-pull-method-supported" attribute is added in order to support the new Delivery Method defined in this document:

'ippget': The IPP Notification Pull Delivery Method defined in this document.

9.2. operations-supported (1setOf type2 enum)

Table 7 lists the "operation-id" value defined in order to support the new Get-Notifications operation defined in this document.

Table 7. Operation-id Assignments

Value	Operation Name
0x001C	Get-Notifications

10. New Status Codes

The following status code is defined as an extension for this Delivery Method and is returned as the status code of the Get-Notifications operation in Group 1 or Group 3 to N (see section 5.2).

10.1. successful-ok-events-complete (0x0007)

The Printer MUST return the 'successful-ok-events-complete' status code to indicate when this Get-Notifications response is the last response for a Subscription object, whether or not there are Event Notifications being returned. This condition occurs for Event Wait Mode with Notification Recipients waiting for responses when (1) the Subscription Object is canceled with a Cancel-Subscription operation, (2) the Subscription object is deleted, when the Per-Printer Subscription lease time expires, or (3) the 'job-completed' event occurs for a Per-Job Subscription. This condition also occurs for a Get-Notifications request that a Notification Recipient makes after the job completes, but before the Event Life expires.

11. Encoding and Transport

This section defines the encoding and transport considerations for this Delivery Method based on [RFC2910].

The encoding of a Get-Notifications Response is modeled after the Get-Jobs Response (see [RFC2911]). In a Get-Notifications Response, each Event Notification Attributes Group MUST start with an 'event-notification-attributes-tag' (see the section "Encodings of Additional Attribute Tags" in [RFC3995]), and end with an 'end-of-attributes-tag'. In addition, for Event Wait Mode the multi-part/related is used to separate each multiple response (in time) to a single Get-Notifications Request.

The Printer returns Get-Notification Response as follows:

1. If the Notification Recipient client did not request Event Wait Mode ("notify-wait" = 'false' or omitted), the Printer ends the response with an 'end-of-attributes-tag' (see [RFC2911], Get-Jobs encoding), as with any operation response.
2. If the Notification Recipient client requests Event Wait Mode ("notify-wait" = 'true') and the Printer wishes to honor the request, the Printer MUST return the response as an application/ipp part inside a multi-part/related MIME media type. When one or more additional Events occur, the Printer returns each as an additional Event Notification Group using a separate application/ipp part under the multi-part/related type.
3. If the client requested Event Wait Mode ("notify-wait" = 'true'), but the Printer does not wish to honor the request in the initial response and wants the client explicitly polled for Event Notifications, the Printer MUST return the "notify-get-interval" operation attribute (see section 5.2.1). The Printer returns the

response as an application/ipp part that MAY be inside an multi-part/related type. The client MUST accept this response and reissue the Get-Notifications request in the future indicated by the value of the "notify-get-interval" attribute value.

4. If the client requested Event Wait Mode ("notify-wait" = 'true'), and the Printer initially honored the request but later wishes to leave Event Wait Mode, the Printer MUST return the "notify-get-interval" operation attribute (see section 5.2.1). The Printer returns the response as an application/ipp part that MUST be inside an multi-part/related type.

NOTE: If a Notification Recipient fails to receive a response, it can ask the Printer for the same Event Notifications again. The Notification Recipient will receive the same Event Notifications that it should have received the first time, except for those Event Notifications that have expired in the meantime.

The Printer MAY chunk the responses, but this has no significance to the IPP semantics.

This notification delivery method uses the IPP transport and encoding [RFC2910] for the Get-Notifications operation with the following extension, allocated in [RFC3995]:

Table 8. The "event-notification-attributes-tag" Value

Tag Value (Hex)	Meaning
0x07	"event-notification-attributes-tag"

12. Conformance Requirements

This section lists the conformance requirements for clients and Printers.

12.1. Conformance for IPP Printers

It is OPTIONAL for a Printer to support IPP Notifications as defined in [RFC3995]. However, if a Printer supports IPP Notifications, the Printer MUST support the 'ippget' Delivery Method, as defined in this document, as one of its Delivery Methods. IPP Printers that conform to this specification

1. MUST meet the conformance requirements defined in [RFC3995] for a Pull Delivery Method;

2. MUST support the Get-Notifications operation defined in section 5, including Event Wait Mode;
3. MUST support the Subscription Template object attributes, as defined in section 6;
4. MUST support the Subscription Description object attributes, as defined in section 7;
5. MUST support the "ippget-event-life" Printer Description attribute defined in section 8.1, including retaining jobs in the Job Retention and/or Job History phases for at least as long as the value specified by the Printer's "ippget-event-life";
6. MUST support the additional values for IPP/1.1 Printer Description attributes defined in section 9;
7. MUST support the 'successful-ok-events-complete' status code, as described in section 10.1;
8. MUST listen for the IPP Get-Notifications operation requests on IANA-assigned well-known port 631, unless explicitly configured by system administrators or site policies;
9. SHOULD NOT listen for IPP Get-Notifications operation requests on any other port, unless explicitly configured by system administrators or site policies; and
10. MUST meet the security conformance requirements stated in section 18.4.

12.2. Conformance for IPP Clients

It is OPTIONAL for an IPP Client to support IPP Notifications as defined in [RFC3995]. However, if a client supports IPP Notifications, the client MUST support the 'ippget' Delivery Method as defined in this document as one of its Delivery Methods. IPP Clients that conform to this specification:

1. MUST create Subscription Objects by sending Subscription Creation operation requests containing the "notify-pull-method" attribute (as opposed to the "notify-recipient-uri" attribute) using the 'ippget' keyword value (see sections 6.1 and 15.2);
2. MUST send IPP Get-Notifications operation requests (see section 5.1) via the port specified in the associated 'ipp' URL (if present) or otherwise via IANA-assigned well-known port 631;

3. MUST convert the associated 'ipp' URLs for use in IPP Get-Notifications operation to their corresponding 'http' URL forms for use in the HTTP layer, according to the rules in section 5, "IPP URL Scheme", in [RFC2910]; and
4. MUST meet the security conformance requirements stated in section 18.5.

13. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2910] Herriot, R., Butler, S., Moore, P., Turner, R., and J. Wenn, "Internet Printing Protocol/1.1: Encoding and Transport", RFC 2910, September 2000.
- [RFC2911] Hastings, T., Herriot, R., deBry, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.1: Model and Semantics", RFC 2911, September 2000.
- [RFC3995] Herriot, R. and T. Hastings, "Internet Printing Protocol (IPP): Event Notifications and Subscriptions", RFC 3995, March 2005.

14. Informative References

- [RFC2565] Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.0: Encoding and Transport", RFC 2565, April 1999.
- [RFC2566] deBry, R., Hastings, T., Herriot, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.0: Model and Semantics", RFC 2566, April 1999.
- [RFC2567] Wright, F., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.
- [RFC2568] Zilles, S., "Rationale for the Structure of the Model and Protocol for the Internet Printing Protocol", RFC 2568, April 1999.
- [RFC2569] Herriot, R., Hastings, T., Jacobs, N., and J. Martin, "Mapping between LPD and IPP Protocols", RFC 2569, April 1999.

- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC2707] Bergman, R., Hastings, T., Isaacson, S., and H. Lewis, "Job Monitoring MIB - V1.0", RFC 2707, November 1999.
- [RFC3196] Hastings, T., Manros, C., Zehler, P., Kugler, C., and H. Holst, "Internet Printing Protocol/1.1: Implementor's Guide", RFC 3196, November 2001.
- [RFC3997] Hastings, T., Ed., deBry, R., and H. Lewis, "Internet Printing Protocol (IPP): Requirements for IPP Notifications", RFC 3997, March 2005.

15. IANA Considerations

This section contains the exact information that the IANA has added to the IPP Registries according to the procedures defined in [RFC2911], section 6. These registrations have been published in the <http://www.iana.org/assignments/ipp-registrations> registry.

15.1. Attribute Registrations

The following table lists the attributes defined in this document. This has been registered according to the procedures in RFC 2911 [RFC2911] section 6.2.

Printer Description attributes:	Reference	Section
-----	-----	-----
ippget-event-life (integer(15:MAX))	[RFC3996]	8.1

15.2. Delivery Method and Additional Keyword Attribute Value Registrations for Existing Attributes

This section lists additional keyword attribute value registrations for use with existing attributes defined in other documents. These have been registered according to the procedures in [RFC2911], section 6.1. According to [RFC3995], section 24.7.3, Pull Delivery Method registrations are the keyword attribute value registrations for the "notify-pull-method" and "notify-pull-method-supported" attributes.

Attribute (attribute syntax) Values	Reference	Section
-----	-----	-----
notify-pull-method (type2 keyword)	[RFC3995]	5.3.2
notify-pull-method-supported (1setOf type2 keyword)	[RFC3995]	5.3.2.1
ippget	[RFC3996]	9.1

15.3. Additional Enum Attribute Values

The following table lists the enum attribute values defined in this document. These have been registered according to the procedures in [RFC2911], section 6.1.

Attribute (attribute syntax) Value Name	Reference	Section
-----	-----	-----
operations-supported (1setOf type2 enum)	[RFC2911]	4.4.15
0x001C Get-Notifications	[RFC3996]	9.2

15.4. Operation Registrations

The following table lists the operations defined in this document. This has been registered according to the procedures in RFC 2911 [RFC2911] section 6.4.

Operations:	Reference	Section
-----	-----	-----
Get-Notifications	[RFC3996]	5

15.5. Status Code Registrations

The following table lists the status codes defined in this document. This has been registered according to the procedures in [RFC2911], section 6.6.

Status codes:	Reference	Section
-----	-----	-----
successful-ok-events-complete (0x0007)	[RFC3996]	10.1

16. Internationalization Considerations

The IPP Printer MUST localize the "notify-text" attribute as specified in section 14 of [RFC3995].

In addition, when the client receives the Get-Notifications response, it is expected to localize the attributes that have the 'keyword' attribute syntax according to the charset and natural language requested in the Get-Notifications request.

17. Security Considerations

The IPP Model and Semantics document [RFC2911, section 8] discusses high-level security requirements (Client Authentication, Server Authentication and Operation Privacy). The IPP Transport and Encoding document [RFC2910, section 8] discusses the security requirements for the IPP protocol. Client Authentication is the mechanism by which the client proves its identity to the server in a secure manner. Server Authentication is the mechanism by which the server proves its identity to the client in a secure manner. Operation Privacy is defined as a mechanism for protecting operations from eavesdropping.

The 'ippget' Delivery Method with its Get-Notifications operations leverages the security mechanism that are used in IPP/1.1 [RFC2910 and RFC2911] without adding any additional security mechanisms in order to maintain the same security support as IPP/1.1.

The access control model for the Get-Notifications operation defined in this document is the same as the access control model for the Get-Job-Attributes operation (see [RFC2911], section 3.2.6). The primary difference is that a Get-Notifications operation is directed at Subscription Objects rather than at Job objects, and a returned attribute group contains Event Notification attributes rather than Job object attributes.

17.1. Notification Recipient Client Access Rights

The Notification Recipient client MUST have the following access rights to the Subscription object(s) targeted by the Get-Notifications operation request:

The authenticated user (see [RFC2911], section 8.3) performing this operation MUST be (1) the owner of each Subscription Object identified by the "notify-subscription-ids" operation attribute (see section 5.1.1), (2) an operator or administrator of the Printer (see [RFC2911], sections 1 and 8.5), or (3) otherwise authorized by the Printer's administrator-configured security policy to request Event Notifications from the target Subscription Object(s). Furthermore, the Printer's security policy MAY limit

the attributes returned by the Get-Notifications operation, in a manner similar to that of the Get-Job-Attributes operation (see [RFC2911], end of section 3.3.4.2).

17.2. Printer Security Threats

Because the Get-Notifications operation is sent in the same direction as are Job Creation operations, usually by the same client, this Event Notification Delivery Method poses no additional authentication, authorization, privacy, firewall, or port assignment issues above those for the IPP Get-Job-Attributes and Get-Printer-Attributes operations (see [RFC2911], sections 3.2.6 and 3.2.5).

17.3. Notification Recipient Security Threats

Unwanted Events Notifications (spam): Unlike Push Event Notification Delivery Methods in which the IPP Printer initiates the Event Notification, with the Pull Delivery Method defined in this document, the Notification Recipient is the client that initiates the Get-Notifications operation (see section 5). Therefore, with this method there is no chance of "spam" notifications.

Note: When a client stays connected to a Printer by using the Event Wait Mode (see section 5.1.3) in order to receive Event Notifications as they occur, it can close down the IPP connection at any time and so can avoid future unwanted Event Notifications at any time.

It is true that the client has control over whether to ask for Event Notifications. However, if the client subscribes to an event and does a Get-Notifications request, it gets all events for the Subscription Object in the sequence number range (see section 5.1.2), not just those it wants. If a client subscribes to a Per-Printer Subscription job event, such as 'job-completed', and someone then starts and cancels thousands of jobs, the client would have to receive these events in addition to those it is interested in. A client can protect itself better by subscribing to its own jobs by using a Per-Job Subscription, rather than create a Per-Printer subscription whose Job events apply to all jobs.

17.4. Security Requirements for Printers

For the Get-Notifications operation defined in this document, the same Printer conformance requirements apply for supporting and using Client Authentication, Server Authentication and Operation Privacy as stated in [RFC2910] section 8 for all IPP operations.

17.5. Security Requirements for Clients

For the Get-Notifications operation defined in this document, the same client conformance requirements apply for supporting and using Client Authentication, Server Authentication, and Operation Privacy as stated in [RFC2910], section 8, for all IPP operations.

18. Description of Base IPP Documents (Informative)

The base set of IPP documents includes the following:

- Design Goals for an Internet Printing Protocol [RFC2567]
- Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
- Internet Printing Protocol/1.1: Model and Semantics [RFC2911]
- Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
- Internet Printing Protocol/1.1: Implementer's Guide [RFC3196]
- Mapping between LPD and IPP Protocols [RFC2569]

"Design Goals for an Internet Printing Protocol" takes a broad look at distributed printing functionality, and it enumerates real-life scenarios that help clarify the features that need to be included in a printing protocol for the Internet. It identifies requirements for three types of users: end users, operators, and administrators. It calls out a subset of end user requirements that are satisfied in IPP/1.0 [RFC2566, RFC2565]. A few OPTIONAL operator operations have been added to IPP/1.1.

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

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

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

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

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

19. Contributors

Carl Kugler and Harry Lewis contributed the basic idea of in-band "smart polling" coupled with multiple responses for a single operation on the same connection, with one response for each event as it occurs. Without their continual persuasion, we would not have arrived at this Delivery Method specification and would not have been able to agree on a single REQUIRED Delivery Method for IPP.

Carl Kugler
IBM Corporation
6300 Diagonal Highway
Boulder, CO 80301

E-Mail: kugler@us.ibm.com

Authors' Addresses

Robert Herriot
Global Workflow Solutions
706 Colorado Ave.
Palo Alto, CA 94303

Phone: 650-324-4000
EMail: bob@herriot.com

Thomas N. Hastings
Xerox Corporation
710 S Aviation Blvd. ESAE 242
El Segundo CA 90245

Phone: 310-333-6413
Fax: 310-333-6342
EMail: hastings@cp10.es.xerox.com

Harry Lewis
IBM Corporation
6300 Diagonal Hwy
Boulder, CO 80301

Phone: (303) 924-5337
EMail: harryl@us.ibm.com

Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

