Internet Printing Protocol WG There are 2 ISSUES highlighted like this
INTERNET-DRAFT
<draft-ietf-ipp-notify-get-05.txt>
Updates: RFC 2911
[Target category: standards track]
Expires: April 17, 2002

Robert Herriot (editor)
Xerox Corp.
Carl Kugler
Harry Lewis
IBM, Corp.

October 17, 2001

Internet Printing Protocol (IPP): **The ‘ippget’ Delivery Method for Event Notifications**

Status of this Memo:

Abstract

This document describes an extension to the Internet Printing Protocol/1.0 (IPP) [RFC2566, RFC2565] and IPP/1.1 [RFC2911, RFC2910]. This document specifies the ‘ippget’ Delivery Method for use with the “IPP Event Notifications and Subscriptions” specification [ipp-ntfy]. When IPP Notification [ipp-ntfy] is supported, the Delivery Method defined in this document is one of the RECOMMENDED Delivery Methods for Printers to support.

The ‘ippget’ Delivery Method is a ‘pull’ Delivery Method with aspects of a ‘push’ method as well. That is, when an Event occurs, the Printer saves the Event Notification for a period of time called the Event Life. The Notification Recipient fetches (pulls) Event Notifications using the Get-Notifications operation. If the Notification Recipient has selected the **Event Wait Mode** option to wait for additional Event Notifications, the Printer continues to return (similar to push) Event Notifications to the Notification Recipient as Get-Notification responses as Events occur. This push aspect is not a true ‘push’, since the Printer does not open the connect, but rather continues to return responses as Events occur using the connection originated by the Notification Recipient.

Either the Notification Recipient or the Printer can terminate **Event Wait Mode** without closing the connection.

Table of Contents

Table of Tables

1 Introduction

The "IPP Event Notifications and Subscriptions" document [ipp-ntfy] defines an OPTIONAL extension to Internet Printing Protocol/1.0 (IPP) [RFC2566, RFC2565] and IPP/1.1 [RFC2911, RFC2910]. For a description of the base IPP documents, see section 19. The [ipp-ntfy] extension defines operations that a client can perform in order to create Subscription Objects in a Printer and carry out other operations on them. A Subscription Object represents a Subscription abstraction. A client associates Subscription Objects with a particular Job by performing the Create-Job-Subscriptions operation or by submitting a Job with subscription information. A client associates Subscription Objects with the Printer by performing a Create-Printer-Subscriptions operation. Four other operations are defined for Subscription Objects: Get-Subscriptions-Attributes, Get-Subscriptions, Renew-Subscription, and Cancel-Subscription. The Subscription Object specifies that when one of the specified Events occurs, the Printer sends an asynchronous Event Notification to the specified Notification Recipient via the specified Delivery Method (i.e., protocol).

The "IPP Event Notifications and Subscriptions" document [ipp-ntfy] specifies that each Delivery Method is defined in another document. This document is one such document, and it specifies the 'ippget' delivery method. When IPP Notification [ipp-ntfy] is supported, the Delivery Method defined in this document is one of the RECOMMENDED Delivery Methods for Printers to support.

The 'ippget' Delivery Method is a 'pull' Delivery Method with aspects of a 'push' method as well. That is, 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 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 continues to return (similar to push) Event Notifications to the Notification Recipient as Get-Notification responses as Events occur. This push aspect is not a true 'push', since the Printer does not open the transaction, but rather continues to return 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" 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" operation attribute in a Get-Notifications response which 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:

This document uses the same terminology as [RFC2911], such as "client", "Printer", "Job", "attribute", "attribute value", "keyword", "operation", "request", "response", and "support".

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 RFC 2119

[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 explicitly stated otherwise.

Event Life: The length of time in seconds after an Event occurs during which the Printer will return that Event in a Event Notification in a Get-Notifications response. After the Event Life expires, the Printer will no longer return an Event Notification for that Even in a Get-Notifications response.

Event Notification Attributes Group: The attributes group in a response that contains attributes that are part of an Event Notification.

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 where the Printer sends subsequent Event Notifications to the Notification Recipient as they occur as additional Get-Notification Responses.

Other capitalized terms, such as Notification Recipient, Event, Event Notification, Compound Event Notification, Printer, etc., are defined in [ipp-ntfy], have the same meanings, and are not reproduced here. However, for convenience the following key terms are reproduced here:

Event – some occurrence (either expected or unexpected) within the printing system of a change of state, condition, or configuration of a Job or Printer object. An Event occurs only at one instant in time and does not span the time the physical Event takes place. For example, jam-occurred and jam-cleared are two distinct, instantaneous Events, even though the jam may last for a while.

Event Notification – the information about an Event that the Printer sends when an Event occurs.

3 Model and Operation

In a Subscription Creation Operation, when the value of the “notify-recipient-uri” attribute has the scheme ‘ippget’, the client is requesting that the Printer use the ‘ippget’ Delivery Method for the Event Notifications associated with the new Subscription Object. The client SHOULD choose a value for the address part of the “notify-recipient-uri” attribute that uniquely identifies the Notification Recipient.

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.

ISSUE 01: Although we agreed to extend Job Creation operations to support **Event Wait Mode**, it seems to be an unnecessary complication, since the Printer MUST keep events for at least 15 seconds. So OK NOT to add the “notify-wait” (boolean) operation attribute to Job Creation operations and NOT have to have Job Creation responses return Event Notification Groups (in addition to returning Subscription Attribute Groups).

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) for a privileged client that wants to get all job or printer Events from a per-Printer Subscription object. If several groups of users expect to receive jobs from other users (FAX paradigm) and each group has a different designated person, say, a secretary, to receive job completion Events, the Printer should be configured to support multiple URLs, one for each group. Then the designated person can run an application that gets the events for jobs submitted to that URL from the per-Printer Subscription object that the application creates.

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	Delivery Method Realization
1. What is the URL scheme name for the Delivery Method?	ippget
2. Is the Delivery Method REQUIRED, RECOMMENDED or OPTIONAL for an IPP Printer to support?	RECOMMENDED
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 connection.
6. Is the Event Notification content Machine Consumable or Human Consumable?	Machine Consumable
7. What section in this document answers the following question? For a Machine Consumable Event Notification, what is the representation and encoding of values defined in section 9.1 of [ipp-ntfy] and 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 [ipp-ntfy] 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.

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's "notify-recipient-uri" attribute matches the scheme value of 'ippget' using the (case-insensitive) matching rules in section 11.5.2 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 is the owner of or has read-access rights to the identified Subscription Object.

The Notification Recipient client can request **Event Wait Mode** by supplying the "notify-wait" operation attribute with a 'true' value.

The Notification Recipient client can terminate **Event Wait Mode** (without closing the connection) by supplying the "notify-wait" 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" operation attribute in a Get-Notifications response which tells the Notification Recipient how long to wait before trying again.

The Printer 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.

Access Rights: If the policy of the Printer is to allow all users to access all Event Notifications, then the Printer MUST accept this operation from any user. Otherwise, the authenticated user (see [RFC2911] section 8.3) performing this operation MUST be the owner of each Subscription Object identified by the "notify-subscription-ids" operation attribute (as returned during a Subscription Creation Operation) or an operator or administrator of the Printer (see [RFC2911] Sections 1 and 8.5). Otherwise, the IPP object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' status code as appropriate.

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 which 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" (1setOf 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, the Printer MUST return the 'client-error-not-found' status code.

If an identified Subscription Object does not have a "notify-recipients-uri" Subscription attribute with 'ippget' scheme (case insensitive-match - see section 11.5.2), the Printer MUST reject the request and return the 'client-error-uri-scheme-not-supported' status code.

> Note: The name of both the "notify-subscription-ids" and "notify-sequence-numbers" end in 's', since 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 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 the number of values 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 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, since '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 of the first Event Notification in the second Get-Notifications Response may be less than the time stamp of the last Event Notification in the first Get-Notification Response. This happens because the Printer sends all unexpired Event Notification with a sequence number equal or higher according to the ordering specified in [ipp-ntfy] 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 or not 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 return immediately whatever Event Notifications it currently holds in the requested Subscription object(s) and MUST return the "notify-get-interval" operation attribute with 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 return immediately whatever Event Notifications it currently holds in the requested Subscription object(s) and MUST continue to return Event Notifications as they occur until all of 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 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 12 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 is a description 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: indicate when this return is the last return for all Subscription objects that match the request, whether or not there are Event

Notifications being returned. This condition occurs for **Event Wait Mode** with Notification Recipients waiting for responses when the Subscription Object is: (1) canceled with a Cancel-Subscription operation, (2) deleted when the Per-Printer Subscription lease time expires, or (3) when 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 Object’s whose “notify-subscription-id” attribute equals any of the values of the “notify-subscription-ids” operation attribute supplied.

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 for determining when to perform this operation again.

redirection-other-site: The Printer does not handle this operation and requests the Notification Recipient to perform the operation again with the uri specified by the “redirect-uri” Operation Attribute in the response. See section 10.2.

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 re-issue the Get-Notifications operation (with or without “notify-wait” = ‘true’) 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 the value of the Printer’s “ippget-event-life” Printer Description attribute (see section 8.1). The Printer MAY return a value that is larger than the value 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 be staying in **Event Wait Mode** or may be requesting the Notification Recipient to leave **Event Wait Mode**):

Table 2 - Combinations of “notify-wait”, “status-code”, and “notify-get-interval”

client sends: “notify-wait”	Printer returns: “status-code”	Printer returns: “notify-get-interval”	Event Notification Attribute Groups
1. ‘false’/omitted	‘successful-ok’	MUST return N	maybe
2. ‘false’/omitted	‘not-found’	MUST NOT	MUST NOT
3. ‘false’/omitted	‘busy’	MUST return N	MUST NOT
4. ‘false’/omitted	‘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

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: Printer either (1) 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, while for case (2), it is unlikely that any Event Notifications are returned; 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 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.

5.2.3 “redirect-uri” (uri):

The value of this attribute is the uri that the Notification Recipient MUST use for a subsequent Get-Notifications operation. The Printer MAY support this attribute. This attribute MUST be returned in the Operation Attributes Group if and only if the Printer returns the ‘redirection-other-site’ status code (see section 10.2).

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 [ipp-ntfy]). The matched Event Notifications are all un-expired Event Notification associated with the matched Subscription Objects and MUST follow the “Event Notification Ordering” requirements for Event Notifications within a Compound Event Notification specified in [ipp-ntfy] 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 [ipp-ntfy] with exceptions denoted by asterisks in the tables below.

The tables below are copies of the tables in section 9.1 ("Content of Machine Consumable Event Notifications") of [ipp-ntfy] 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 [ipp-ntfy] 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	Sends	Source Object
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. The 'ippget' Delivery Method has the same conformance requirements for Subscription Template attributes as defined in [ipp-ntfy]. This section defines additional information about Subscription Template attributes defined in [ipp-ntfy].

6.1 notify-recipient-uri (uri)

This section describes the syntax of the value of this attribute for the ‘ippget’ Delivery Method. The syntax for values of this attribute for other Delivery Method is defined in other Delivery Method Documents.

In order to support the ‘ippget’ Delivery Method and Protocol, the Printer MUST support the following syntax:

> The ‘ippget://’ URI scheme. The remainder of the URI indicates something unique about the Notification Recipient, such as its host name or host address (and optional path). However, the remainder of the URI is not used by the Printer in any way. Its value MAY be useful to Notification Recipients who are not the Subscription Creation clients. See section 11 for a complete definition of the syntax of the IPPGET URL.

7 Subscription Description Attributes

The ‘ippget’ Delivery Method has the same conformance requirements for Subscription Description attributes as defined in [ipp-ntfy]. 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, AND

2. an Event associated with the new Job occurs immediately after the Subscription Object is created, AND

3. the same client or some other client performs a Get-Notifications operation such 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 a time that is sufficiently less that 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 that this value. See [RFC2911] section 4.3.7.1 and the discussion in [ipp-ntfy] ‘job-completed’ event) that explains that a Notification Recipients 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 contain the ‘ippget’ scheme. 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 defined in [ipp-ntfy].

9.1 notify-schemes-supported (1setOf uriScheme)

The following value for the “notify-schemes-supported” attribute is added in order to support the new Delivery Method defined in this document:

‘ippget’ - The IPP Notification 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 codes are defined as extensions for this Delivery Method and are returned as the status code of the Get-Notifications operation.

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 the Subscription Object is: (1) canceled with a Cancel-Subscription operation, (2) deleted when the Per-Printer Subscription lease time expires, or (3) when 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.

10.2 redirection-other-site (0x0300)

This status code means that the Printer doesn’t perform that Get-Notifications operation and that the “redirect-uri” operation attribute in the response contains the uri that the Notification Recipient MUST use for performing the Get-Notifications operation. If the client issues subsequent Get-Notifications operations, it MUST use the value of the “redirect-uri” operation attribute returned by the Printer as the target of the operation.

11 The IPPGET URL Scheme

This section defines the ‘ippget' URL and the conformance requirements for using it.

11.1 The IPPGET URL Scheme Applicability and Intended Usage

This section is intended for use in registering the ‘ippget’ URL scheme with IANA and fully conforms to the requirements in [RFC2717]. This document defines the ‘ippget’” URL (Uniform Resource Locator) scheme for specifying a unique identifier for an IPP Client which implements the IPP Get-Notifications operation specified in this document (see section 5).

ISSUE 02: How unique do we need now that the Printer doesn’t use anything but the scheme?

The intended usage of the ‘ippget’ URL scheme is COMMON.

11.2 The IPPGET URL Scheme Associated Port

None.

An ‘ippget’ URL behaves as a unique identifier for IPP Clients and is NOT used to initiate any over-the-wire protocol associations.

See: IANA Port Numbers Registry [IANA-PORTREG].

11.3 The IPPGET URL Scheme Associated MIME Type

All IPP Get-Notifications operations (requests and responses) MUST be conveyed in an ‘application/ipp’ MIME media type as registered in [IANA-MT]. An ‘ippget’ URL MUST uniquely identify an IPP Client that support this ‘application/ipp’ MIME media type.

See: IANA MIME Media Types Registry [IANA-MT].

11.4 The IPPGET URL Scheme Character Encoding

The ‘ippget’ URL scheme defined in this document is based on the ABNF for the URI Generic Syntax [RFC2396] and further updated by [RFC2732] and [RFC2373] (for IPv6 addresses in URLs). The ‘ippget’ URL scheme is case-insensitive in the scheme and 'authority' part as in [RFC2396]; however, the 'abs_path' part is case-sensitive, as in [RFC2396]. Code points outside [US-ASCII] MUST be hex escaped by the mechanism specified in [RFC2396].

11.5 The IPPGET URL Scheme Syntax in ABNF

This document is intended for use in registering the ‘ippget’ URL scheme with IANA and fully conforms to the requirements in [RFC2717]. This document defines the ‘ippget’ URL (Uniform Resource Locator) scheme for specifying a unique identifier for an IPP Client which implements IPP ‘Get-Notifications’ operation specified in this document.

The intended usage of the ‘ippget’ URL scheme is COMMON.

The value of an ‘ippget’ URI MUST NOT exceed 255 octets (see section 8.1), since the URI is for identification rather than for identifying the location of a network resource. An IPP Printer MUST return the ‘client-error-request-value-too-long’ status code (see section 13.1.4.10 in [RFC2911]) when a URI received in a request is too long.

An ‘ippget’ URL MUST be represented in absolute form. Absolute URLs always begin with a scheme name followed by a colon. For definitive information on URL syntax and semantics, see “Uniform Resource Identifiers (URI): Generic Syntax and Semantics” [RFC2396]. This specification adopts the definitions of “authority”, “abs_path”, “query”, “reg_name”, “server”, “userinfo”, and “hostport” from [RFC2396], as updated by [RFC2732] and [RFC2373] (for IPv6 addresses in URLs).

The ‘ippget’ URL scheme syntax in ABNF is as follows:

```
   ippget_URL = “ippget:” “//” authority [ abs_path [ “?” query ]]
   authority  = server | reg_name
   reg_name   = 1*( unreserved | escaped | “$” | “,” |
                    “;” | “:” | “@” | “&” | “=” | “+” )
   server     = [ [ userinfo “@” ] hostport ]
   userinfo   = *( unreserved | escaped |
                   “;” | “:” | “&” | “=” | “+” | “$” | “,” )
   hostport   = host [ “:” port ]
   abs_path   = “/”  path_segments
```

If the port is empty or not given, then no port is assumed. The semantics are that the ‘ippget’ URL is a unique identifier for an IPP Client that will retrieve IPP event notifications via the IPP Get-Notifications operation.

Note: The use of IP addresses in URLs SHOULD be avoided whenever possible (see [RFC1900]).

11.5.1 IPPGET URL Examples

The following are examples of valid ‘ippget’ URLs for IPP Clients (using DNS host names):

```
       ippget://abc.com
       ippget://abc.com/listener
       ippget://bob@abc.com
       ippget://bob@abc.com/listener/1232
```

Note: The use of IP addresses in URLs SHOULD be avoided whenever possible (see [RFC1900]).

The IPP Client that creates the Subscription object and the Notification Recipient have to agree on a unique IPPGET URL value in order for the Get-Notifications operations to retrieve the proper Event Notifications. Therefore, the choice of ‘userinfo@hostport’ versus the simpler ‘hostport’ production in an ‘ippget’ URL may be influenced by the intended usage.

If a given IPP Client creates an IPP Subscription object for event notifications intended for retrieval by the same IPP Client, then the simple ‘hostport’ production may be most appropriate. In this case, the IPP Client and the Notification Recipient both know the 'hostport' of the client.

On the other hand, if a given IPP Client creates an IPP Subscription object for event notifications intended for retrieval by a *different* IPP Client, then the ‘userinfo@hostport’ production (using, for example, the right-hand side of a ‘mailto:’ URL, see [RFC2368]) may be most appropriate. For this case, a mail address serves as the prior agreement on the IPPGET URL value between the IPP Client and the Notification Recipient.

11.5.2 IPPGET URL Comparisons

When comparing two ‘ippget’ URLs to decide if they match or not, an IPP Client or IPP Printer MUST use the same rules as those defined for HTTP URI comparisons in [RFC2616].

12 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 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 [ipp-ntfy]), 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 but wants the client explicitly poll 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 which MAY be inside an multi-part/related type. The client MUST accept this response and re-issue 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 which MUST be inside an multi-part/related type.

Note: All of the above is without either the Printer or the Notification Recipient closing the connection. In fact, the connection SHOULD remain open for any subsequent IPP operations. However, either the Notification Recipient or the Printer can abnormally terminate by closing the connection. But, if the Printer closes the connection too soon after returning the response, the client may not receive the response.

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

Note: While HTTP/1.1 allows a proxy to collect chunked responses over a period of time and return them back as a single un-chunked response (with a Content Length instead). However, in practice no proxy wants to have an infinite buffer. Also no proxy want to hold up responses, since user would be furious.

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

Table 8 – The "event-notification-attributes-tag" value

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

13 Conformance Requirements

The 'ippget' Delivery Method is RECOMMEND for Printers to support.

13.1 Conformance for IPP Printers

IPP Printers that conform to this specification:

1. MUST meet the conformance requirements defined in [ipp-ntfy];

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 support the "redirection-other-site" status code defined 10.2, if it redirects Get-Notifications operations;

9. 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;

10. SHOULD NOT listen for IPP Get-Notifications operation requests on any other port, unless explicitly configured by system administrators or site policies.

13.2 Conformance for IPP Clients

IPP Clients that conform to this specification:

1. MUST create unambiguously unique 'ippget' URLs in all cases that conform to the ABNF specified in section 11.5 of this document;
2. ;MUST send IPP Get-Notifications operation requests 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].

Note: The use of ambiguous 'ippget' URLs is NOT an optional feature for IPP Clients; it is a non-conformant implementation error.

14 IANA Considerations

IANA shall register the 'ippget' URL scheme as defined in section 11 according to the procedures of [RFC2717].

The rest of this section contains the exact information for IANA to add to the IPP Registries according to the procedures defined in RFC 2911 [RFC2911] section 6.

Note to RFC Editors: Replace RFC NNNN below with the RFC number for this document, so that it accurately reflects the content of the information for the IANA Registry.

14.1 Additional attribute value registrations for existing attributes

This section lists additional attribute value registrations for use with existing attributes defined in other documents.

14.1.1 Additional values for the "notify-schemes-supported" Printer attribute

The following table lists the uriScheme value defined in this document as an additional uriScheme value for use with the "notify-schemes-supported" Printer attribute defined in [ipp-ntfy]. This is to be registered according to the procedures in RFC 2911 [RFC2911] section 6.1.

uriScheme Attribute Values:	Ref.	Section:
ippget	RFC NNNN	9.1

The resulting URI scheme attribute value registrations will be published in the ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/notify-schemes-supported/ area.

14.1.2 Additional values for the “operations-supported” Printer attribute

The following table lists the enum attribute value defined in this document as an additional type2 enum value for use with the “operations-supported” Printer attribute defined in [RFC2911]. This is to be registered according to the procedures in RFC 2911 [RFC2911] section 6.1.

type2 enum Attribute Values:	Value	Ref.	Section:
Get-Notifications	0x001C	RFC NNNN	9.2

The resulting enum attribute value registration will be published in the ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attribute-values/operations-supported/ area.

14.2 Operation Registrations

The following table lists the operation defined in this document. This is to be registered according to the procedures in RFC 2911 [RFC2911] section 6.4.

Operations:	Ref.	Section:
Get-Notifications operation	RFC NNNN	5

The resulting operation registration will be published in the ftp://ftp.iana.org/in-notes/iana/assignments/ipp/operations/ area.

14.3 Attribute Registrations

The following table lists the attribute defined in this document. This is to be registered according to the procedures in RFC 2911 [RFC2911] section 6.2.

Printer Description attributes:	Ref.	Section:
ippget-event-life (integer(15:MAX))	RFC NNNN	8.1

The resulting attribute registration will be published in the ftp://ftp.iana.org/in-notes/iana/assignments/ipp/attributes/ area.

14.4 Status code Registrations

The following table lists the status code defined in this document. This is to be registered according to the procedures in RFC 2911 [RFC2911] section 6.6.

Status codes:	Ref.	Section:
successful-ok-events-complete (0x0007)	RFC NNNN	10.1
redirection-other-site (0x0300)	RFC NNNN	10.2

The resulting status code registration will be published in the ftp://ftp.iana.org/in-notes/iana/assignments/ipp/status-codes/ area.

15 Internationalization Considerations

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

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.

16 Security Considerations

The IPP Model and Semantics document [RFC2911] discusses high-level security requirements (Client Authentication, Server Authentication and Operation Privacy). 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.

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

Because the Get-Notifications operation is sent in the same direction as Job Creation operations, this Event Notification Delivery Method poses no additional firewall or port assignment issues.

17 References

[IANA-MT]
IANA Registry of Media Types: ftp://ftp.iana.orgisi.edu/in-notes/iana/assignments/media-types/

[IANA-PORTREG]
IANA Port Numbers Registry. ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers

[ipp-iig]
Hastings, T., Manros, C., Kugler, K, Holst H., Zehler, P., "Internet Printing Protocol/1.1: draft-ietf-ipp-implementers-guide-v11-03.txt, work in progress, July 17, 2001

[ipp-ntfy]
R. Herriot, Hastings, T., Isaacson, S., Martin, J., deBry, R., Shepherd, M., Bergman, R., "Internet Printing Protocol/1.1: IPP Event Notifications and Subscriptions", <draft-ietf-ipp-not-spec-07.txt>, August 20, 2001.

[RFC1900]
B. Carpenter, Y. Rekhter. Renumbering Needs Work, RFC 1900, February 1996.

[RFC2026]
S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

[RFC2119]
S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March 1997

[RFC2368]
P. Hoffman, L. Masinter, J. Zawinski. The "mailto" URL Scheme, RFC 2368, July 1998.

[RFC2373]
R. Hinden, S. Deering. IP Version 6 Addressing Architecture, RFC 2373, July 1998.

[RFC2396]
Berners-Lee, T. et al. Uniform Resource Identifiers (URI): Generic Syntax, RFC 2396, August 1998

[RFC2565]
Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.0: Encoding and Transport", RFC 2565, April 1999.

[RFC2566]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, and P. Powell, "Internet Printing Protocol/1.0: Model and Semantics", RFC 2566, April 1999.

[RFC2567]
Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.

[RFC2568]
Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol", RFC 2568, April 1999.

[RFC2569]
Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC 2569, April 1999.

[RFC2616]
R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, 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", November 1999.

[RFC2717]
R. Petke and I. King, "Registration Procedures for URL Scheme Names", RFC 2717, November 1999.

[RFC2732]
R. Hinden, B. Carpenter, L. Masinter. Format for Literal IPv6 Addresses in URL's, RFC 2732, December 1999.

[RFC2910]
Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and Transport", RFC 2910, September 2000.

[RFC2911]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.1: Model and Semantics", RFC 2911, September 2000.

18 Authors' Addresses

Robert Herriot
Xerox Corp.
3400 Hill View Ave, Building 1
Palo Alto, CA 94304

Phone: 650-813-7696
Fax: 650-813-6860
e-mail: robert.herriot@pahv.xerox.com

Carl Kugler
IBM

P.O. Box 1900
Boulder, CO 80301-9191

Phone:
Fax:
e-mail: kugler@us.ibm.com

Harry Lewis
IBM
P.O. Box 1900
Boulder, CO 80301-9191

Phone: 303-924-5337
FAX:
e-mail: harryl@us.ibm.com

IPP Web Page: http://www.pwg.org/ipp/
IPP Mailing List: ipp@pwg.org

To subscribe to the ipp mailing list, send the following email:
1) send it to majordomo@pwg.org
2) leave the subject line blank
3) put the following two lines in the message body:
subscribe ipp
end

Implementers of this specification document are encouraged to join the IPP Mailing List in order to participate in any discussions of clarification issues and review of registration proposals for additional attributes and values. In order to reduce spam the mailing list rejects mail from non-subscribers, so you must subscribe to the mailing list in order to send a question or comment to the mailing list.

19 Description of Base IPP documents

The base set of IPP documents includes:

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 [ipp-iig]
Mapping between LPD and IPP Protocols [RFC2569]
Internet Printing Protocol (IPP): IPP Event Notifications and Subscriptions [ipp-ntfy]

The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed printing functionality, and it enumerates real-life scenarios that help to 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. A few OPTIONAL operator operations have been added to IPP/1.1.

The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document 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.

The "Internet Printing Protocol/1.1: Model and Semantics" document 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.

The "Internet Printing Protocol/1.1: Encoding and Transport" document 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 'ippget' scheme for identifying IPP printers and jobs.

The "Internet Printing Protocol/1.1: Implementer's Guide" document 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.

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

The "IPP Event Notifications and Subscriptions" document defines an extension to IPP/1.0 [RFC2566, RFC2565] and IPP/1.1 [RFC2911, RFC2910]. This extension allows a client to subscribe to printing related Events and defines the semantics for delivering asynchronous *Event Notifications* to the specified *Notification Recipient* via a specified *Delivery Method* (i.e., protocols) defined in (separate) Delivery Method documents.

20 Full Copyright Statement

Acknowledgement

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