Internet Printing Protocol WG
INTERNET-DRAFT-
<draft-ietf-ipp-notify-get-043.txt>
Updates: RFC 2911
[Target category: standards track]
Expires: January 17, 2002

Robert Herriot (editor)
Xerox Corp.
Carl Kugler
Harry Lewis
IBM, Corp.
July 17April 5, 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" Sspecification [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 and push' 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 Notification Lease Time.* The Notification Recipient fetches (pulls) Event Notifications using the Get-Notifications operation. If the Notification Recipient has selected the 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.

38
39 **Table of Contents**

Table of Tables

1 Introduction

The "IPP Event Notifications and Subscriptions" ~~Specification"~~ 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. ~~This~~ 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" ~~Specification"~~ 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' ~~and push'~~ 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 Notification Lease Time.* 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 Notification Recipient. If the Notification Recipient has selected the 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.

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 ~~to this specification. These terms are defined in [RFC2911 section 13.1 on conformance terminology, most of which is taken from~~ 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 Notification Lease: The lease that is associated with an Event Notification. When the lease expires, the Printer discards the associated Event Notification.

Event Notification Lease Time: The expiration time assigned to a lease that is associated with an Event Notification.

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 Notification, Compound Event Notification, Printer, etc., are defined in [ipp-ntfy], have the same meanings, and are not reproduced here.

~~For other capitalized terms that appear in this document, see [ipp-ntfy].~~

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 Notification Lease Time. The Printer MUST hold an Event Notification for its assigned Event Notification Lease Time. The Printer MUST assign the same Event Notification Lease Time to each Event Notification.

When a Notification Recipient wants to receive Event Notifications, it performs the Get-Notifications operation, which causes the Printer to return all un-expired Event Notifications held for the Notification Recipient. 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 the Get-Notification operation, the Printer sends only those Event Notifications that are generated from Subscription Objects whose "notify-recipient-uri" attribute value equals the value of the "notify-recipient-uri" Operation Attribute in the Get-Notifications operation.

If a Notification Recipient performs the Get-Notifications operation twice in quick succession, it will receive nearly the same Event Notification both times because most of the Event Notifications are those that the Printer saves for a few seconds after the Event occurs. There are two possible differences. Some old Event Notifications may not be present in the second response because their Event Notification Leases have expired. Some new Event Notifications may be present in the second response but not the first response, because they occurred after the first response.

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 is likely to save Event Notifications for several seconds, the Notification Recipient is unlikely to miss any Event Notifications that occur between the Subscription Creation and the Get-Notifications operation.

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 ~~and~~ 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
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?	None

5 Get-Notifications operation

This operation is issued by a client acting in the role of a Notification Recipient and causes the Printer to return all Event Notifications held for the Notification Recipient.

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" attribute equals the "notify-subscription-id" Operation attribute if supplied AND

2. Whose associated Subscription Object's "notify-recipient-uri" attribute equals the "notify-recipient-uri" Operation attribute AND

3. Whose associated Subscription Object's "notify-recipient-uri" attribute ~~has a~~ matches the scheme value of 'ippget' using the matching rules in section 11.5.2 AND

4. Whose Event Notification Lease Time has not yet expired AND

5. Where the Notification Recipient is the owner of or has read-access rights to the associated Subscription Object.

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" attribute to indicate when the client should try again.

2. If the Notification Recipient did not request Event Wait Mode, the Printer MUST respond to this operation immediately with whatever Event Notifications it currently holds and return the "notify-get-interval" attribute with number of seconds from now at which the Notification Recipient MAY repeat the Get-Notifications Request to get future Event Notifications.

3. If the Notification Recipient requested Event Wait Mode, ~~T~~the Printer MUST respond to this operation immediately with whatever Event Notifications it currently holds~~.~~ ~~If the Notification Recipient has selected the option to wait for additional Event Notifications, the Printer~~ MUST continue to send Event Notifications as they occur until all of the associated Subscription Objects are cancelled. A Subscription Object is cancelled either via the Cancel-Subscription operation or by the Printer (e.g., the Subscription Object is cancelled when the associated Job completes and is no longer in the Job Retention or Job History phase - see the "ippget-event-time-to-live (integer(0: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 an additional Event Notification Attributes Group that contains the single "notify-get-interval" 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.

~~Note, the Printer terminates the operation in the same way that it normally terminates IPP operations. For example, if the Printer is sending chunked data, it can send a 0 length chunk to denote the end of the operation or it can close the connection.~~ If the Notification Recipient wishes to terminate the Get-Notifications operation, it can close the connection. See section 12 for the encoding and transport rules for the Get-Notifications Response for the Event Wait Mode.

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 either be the owner of each Subscription Object identified by the "notify-recipient-uri" Operation attribute (as determined 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.

"notify-subscription-id" (integer(1:MAX)):
The client SHOULD supply this attribute, if known, and the client is only monitoring a single Subscription object. The Printer object MUST support this attribute. If supplied, but no Subscription Object exists with this identifier, the Printer MUST return the 'client-error-not-found' status code.

If supplied and the identified Subscription Object exists, the Printer MUST check that the Subscription Object's "notify-recipients-uri" attribute scheme is 'ippget' (case insensitive-

match - see section 11.5.2). If the scheme does not match 'ippget', the Printer MUST reject the request and return the 'client-error-uri-scheme-not-supported' status code.

Note: If Notification Recipients supplies this attribute, if known, then the Event Notifications will be sent in time stamp order since only one Subscription object is involved (see "Event Notification Ordering" requirements in [ipp-ntfy] section 9). Supplying this attribute also reduces the Event processing time on the Printer since the Printer doesn't have to search all of the Subscription Objects in order to match the "notify-recipient-uri" operation attribute (see next attribute).

"notify-recipient-uri" (uri~~l~~(255)):

The client ~~MUST~~ MAY supply this attribute whether or not it also supplies the "notify-subscription-id" operation attribute. The Printer object MUST support this attribute. If the client supplies neither the "notify-subscription-id" nor the "notify-recipient-uri", the Printer MUST reject the request and return the 'client-error-bad-request' status code.

If the supplied scheme is not ippget (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.

If the client also supplied the "notify-subscription-id" attribute, then the value of this attribute MUST match the "notify-recipient-uri" Subscription Description attribute for the identified Subscription object. If they do not match, the Printer MUST return the 'client-error-not-found' status code.

If the client did not supply the "notify-subscription-id" operation attribute, ~~T~~the Printer matches the value of this "notify-recipient-uri" attribute ~~(byte for byte with no case conversion)~~ against the value of the "notify-recipient-uri" Subscription Description attribute in each Subscription Object in the Printer using the URI matching rules specified in section 11.5.2. If there are no matches, the IPP Printer MUST return the 'client-error-not-found' status code. ~~For each matched Subscription Object, the IPP Printer MUST return all unexpired Event Notifications associated with it. The Printer MUST send additional Event Notifications as Events occur if and only if the value of the "notify-no-wait" attribute is 'false' or not supplied by the client (see the next attribute below).~~

The value of this attribute is defined to be shorter (255 octets) than the 'uri' attribute syntax (1023 octets) in [RFC2911], since this uri is used for identification, not for locating a network resource.

The [ipp-ntfy] specification REQUIRES that Subscription Object's "notify-recipient-uri" attribute be returned in any operation with the identical representation as supplied by the original Subscribing Client in the Subscription Creation Request. Therefore the Printer implementation MUST use other means to perform the URI match than changing the Subscription Object's original "notify-recipient-uri" value to a canonical form.

Note: this attribute allows a subscribing client to pick URLs that are unique, e.g. the client's own URL or a friend's URL, which in both cases is likely the URL of the person's host. An application could make a URL unique for each application.

"notify-~~no-~~wait" (boolean):

The client MAY supply this attribute. The Printer object MUST support both values of this attribute. If the value is 'true', the client is requesting Event Wait Mode. See the beginning of section 5 for the rules for Event Wait Mode.~~If the value of this attribute is 'false', the Printer MUST send all un-expired Event Notifications (as defined in the previous attribute) and it MUST continue to send responses for as long as the Subscription Objects associated with the specified "notify-recipient-uri" continue to exist. If the value of this attribute is 'true', the Printer MUST send all un-expired Event Notifications (as defined in the previous attribute) and the Printer MUST conclude the operation without waiting for any additional Events to occur. If the client doesn't supply this attribute, the Printer MUST behave as if the client had supplied this attribute with the value of 'false'.~~

5.2 Get-Notifications Response

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 "notify-recipient-uri". If the specified Subscription Objects have no associated Event Notification, the response MUST contain zero Event Notifications.

client-error-not-found: The Printer has no Subscription Object's whose "notify-recipient-uri" attribute equals the "notify-recipient-uri" Operation attribute, if supplied or whose "notify-subscription-id" attribute equals the "notify-subscription-id" Operation attribute, if supplied.

server-error-busy: The Printer is too busy to accept this operation. If the "~~suggested-ask-again-time-interval~~notify-get-interval" operation attribute is present in the Operation Attributes of the response, then the Notification Recipient SHOULD wait for the number of seconds specified by the "~~suggested-ask-again-time-interval~~notify-get-interval" attribute before performing this operation again. If the "~~suggested-ask-again-time-interval~~notify-get-interval" Operation Attribute is not present, the Notification Recipient ~~should~~ 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 "~~notify-ippget-redirect~~redirect-uri" Operation Attribute in the response.

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.

"printer-up-time" (integer(~~0~~1:MAX)):

The value of this attribute is the Printer's "printer-up-time" attribute at the time the Printer sends this response. 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.

~~"suggested-ask-again-time-interval" (integer(0:MAX)):~~

~~The value of this attribute is the number of seconds that the Notification Recipient SHOULD wait before trying this operation again when~~

~~a)the Printer returns the 'server-error-busy' status code OR~~

~~the Printer returns the 'successful-ok' status code and the client supplied the "notify-no-wait" attribute with a value of 'true'This value is intended to help the client be a good network citizen.~~

"~~notify-ippget-redirect~~redirect-uri" (uri):

The value of this attribute is the uri that the Notification Recipient MUST use for ~~the~~ a subsequent Get-Notifications operation. This attribute is ~~present~~ returned in the Operation Attributes Group if and only if the status code has the value 'redirection-other-site'.

Group 2: Unsupported Attributes

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

~~If the "subscription-ids" attribute contained subscription-ids that do not exist, the Printer returns them in this group as value of the "subscription-ids" attribute.~~

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 last Event Notification Attributes Group MAY contain a single "notify-get-interval" (see section 7.1 and 12), in which case the Printer will return no future responses. The initial 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.

If the Notification Recipient has selected the Event Wait Mode option to wait for additional Event Notifications (the "notify-wait" attribute was set to 'true'), the Printer sends ~~the~~ subsequent Event Notifications in the response each time it processes additional Events~~are Event Notifications associated with the matched Subscription Objects as the corresponding Event occurs~~. Each time the Printer sends such Event Notifications, their ordering MUST follow the "Event Notification Ordering" requirements in [ipp-ntfy] section 9.

Note: If a Notification Recipient performs two consecutive Get-Notifications operations, 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 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.

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~~s~~ 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 Event Notification Group MUST start with an 'event-notification-attributes-tag' (see the section "Encodings of Additional Attribute Tags" in [ipp-ntfy]).~~

Each attribute is encoded using the IPP rules for encoding attributes [RFC2910] and ~~may~~ 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.

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".

For an Event Notification for all Events, the Printer includes the attributes shown in Table 2.

Table 2 – 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(~~MIN~~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

* 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 3.

Table 3 – 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 4.

Table 4 – 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 5.

Table 5 – 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 Subscription Template Attributes

This section defines the Subscription object conformance requirements for Printers.

6.1 Subscription Template Attribute Conformance

The 'ippget' Delivery Method has the same conformance requirements for Subscription Template attributes as defined in [ipp-ntfy]. The 'ippget' Delivery Method does not define any addition Subscription Template attributes.

6.2 Additional Information about Subscription Template Attributes

This section defines additional information about Subscription Template attributes defined in [ipp-ntfy].

6.2.1 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) that the Printer uses to match the "notify-recipient-uri" Operation attribute supplied in the Get-Notifications request. See section 11 for a complete definition of the syntax of the IPPGET URL.

6.3 Subscription Description Attribute Conformance

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.

7 Attributes only in Event Notifications

This section defines attributes that exist only in Event Notifications and do no exist in any IPP-defined objects.

7.1 "notify-get-interval" (integer(0:MAX))

The Printer returns this attribute to give the client an indication of when to try another Get-Notifications request in the future. The value of this attribute is the number of seconds that the Notification Recipient SHOULD wait before trying the Get-Notifications operation again. This value is intended to help the client be a good network citizen.

The Printer MUST return this attribute by itself in a separate Event Notification Attributes Group. The Printer MUST return this attribute if and only if:

1. Printer busy case: the Printer returns the 'server-error-busy' status code OR
2. No wait case: the Printer returns the 'successful-ok' status code and the client either (1) supplied the "notify-wait" attribute with a value of 'false' or (2) omitted the attribute entirely OR
3. Printer leaves Event Wait Mode: the Printer returns the 'successful-ok' status code and the client supplied the "notify-wait" attribute with the 'true value (Event Wait Mode) but the Printer wants the client to disconnect (no wait), instead of staying connected. The client MUST accept this response and MUST disconnect. If the client does not disconnect, the Printer SHOULD do so. The Printer returns this attribute for this case only if the implementation does not want to keep the connection open at this time. If the Printer wants the client to keep the connection open and remain in Event Wait Mode, then the Printer MUST NOT return this attribute in the response.

8 Additional Printer Description Attributes

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

~~7.1~~8.1 ippget-event-time-to-live~~begin-to-expire-time-interval~~ (integer(0:MAX))

This Printer Description attribute specifies the number of seconds that a Printer keeps an Event Notification that is associated with the 'ippget' Delivery Method.

The Printer MUST support this attribute if it supports the 'ippget' Delivery Method.

The value of this attribute is the minimum number of seconds that MUST elapse between the time the Printer creates an Event Notification object for the 'ippget' Delivery Method and the time the Printer discards the same Event Notification.

For example, assume the following:

1. a client performs a Job Creation operation that creates a Subscription Object associated with this 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 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.

The value of this attribute also specifies the minimum number of seconds that the Printer, if supporting the ippget Delivery Method, MUST keep 'completed', 'canceled', or 'aborted' Job objects in the Job Retention and/or Job History phases. 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 completing. However, this attribute has no effect on the Cancel-Subscription operation which deletes the object immediately, whether or not it contain the ippget scheme. Immediately thereafter, subsequent Get-Notifications Responses MUST NOT contain Event Notifications associated with the cancelled Subscription object.

~~Additional Printer Description Attributes~~

~~This section defines the Printer Description Attributes conformance requirements for Printers.~~

~~7.1Printer Description Attribute Conformance~~

~~The 'ippget' Delivery Method has the same conformance requirements for Printer Description attributes as defined in [ipp-ntfy]. The 'ippget' Delivery Method does not define any addition Printer Description attributes.~~

9 New Values for Existing Printer Description Attributes

This section defines additional values for existing Printer Description attributes define 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 6 lists the "operation-id" value defined in order to support the new Get-Notifications operation defined in this document.

Table 6 – 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 redirection-other-site (0x0300)

This status code means that the Printer doesn't perform that Get-Notifications operation and that the "~~notify-ippget-redirect~~redirect-uri" Operation Attribute in the response contains the uri that the Notification Recipient MUST use for performing the Get-Notifications operation.

11 The IPPGET URL Scheme

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

~~10.1~~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).

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-MIMEREG]. An 'ippget' URL MUST uniquely identify an IPP Client that support this 'application/ipp' MIME media type.

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

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; 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 IPP protocol places a limit of 1023 octets (NOT characters) on the length of a URI (see section 4.1.5 'uri' in [RFC2911]). 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.

Note: IPP Clients and IPP Printers ought to be cautious about depending on URI lengths above 255 bytes, because some older client or proxy implementations might not properly support these lengths.

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/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].

~~11~~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]), but only the last group ends 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. The Notification Recipient is expected to close the connection.

2. If the Notification Recipient client requests **Event Wait Mode** ("notify-wait" = 'true') and the Printer wishes to honor the request, the Printer ends the Response without an 'end-of-attributes-tag' and MUST return the response as an application/ipp part inside a multi-part/related MIME media type. Neither the Notification Recipient nor the Printer close the connection. 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 to disconnect, the Printer MUST return the "notify-get-interval" attribute (see section 7.1) as the last Event Notifications Attributes Group - see section 5.2), the Printer ends the Response with an 'end-of-attributes-tag'. 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 MUST disconnect. If the client does not disconnect, the Printer SHOULD do so.

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" attribute (see section 7.1) as the last Event Notifications Attributes Group - see section 5.2), the Printer ends the Response with an 'end-of-attributes-tag'. The Printer returns the response as an application/ipp part which MUST be inside an multi-part/related type.

ISSUE: Should we use application/multiplexed (draft-herriot-application-multiplexed-03.txt) which can chunk mime types using content lengths, instead of multi-part/related, which uses boundary strings?

Note: either the Notification Recipient or the Printer can abnormally terminate by closing the connection. However, 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.

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

Table 7 – The "event-notification-attributes-tag" value

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

~~12~~13 Conformance Requirements

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

~~12.1~~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;
3. MUST support the Subscription object attributes as defined in section 6;
4. MUST support the additional values for IPP/1.1 Printer Description attributes defined in section 9;
5. MUST support the "~~begin-to-expire-time-interval~~ippget-event-time-to-live" Printer Description attribute defined in section 8.1;

6. MUST support the "redirection-other-site" status code defined 10.1, if it redirects Get-Notifications operations;

7. SHOULD reject received 'ippget' URLs in 'application/ipp' request bodies (e.g., in the "notify-recipient-uri" attribute in a Get-Notifications request) that do not conform to the ABNF for 'ippget' URLs specified in section 11.5 of this document;

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.

~~12.2~~13.2 Conformance for IPP Clients

IPP Clients that conform to this specification:

1. MUST create unambiguously unique 'ippget' URLs in all cases;

2. MUST send 'ippget' URLs (e.g., in the "notify-recipient-uri" attribute in a Get-Notifications request) that conform to the ABNF specified in section 11.5 of this document;

3. 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;

4. 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.

~~13~~14 IANA Considerations

IANA ~~is requested to~~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 additional IPP entities~~ 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.

~~13.1~~14.1 Operation Registrations

The following table lists the operation~~s~~ defined in this document. This is to be registered ~~will be published by IANA~~ according to the procedures in RFC 2911 [RFC2911] section 6.4~~. with the following path:~~

~~ftp.isi.edu/iana/assignments/ipp/operations/~~

~~The registry entry will contain the following information:~~

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.

~~13.2~~14.2 Additional attribute value registration~~s of~~ for existing attributes

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

~~13.2.1~~14.2.1 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" ~~'uriScheme'~~ Printer attribute defined in [ipp-ntfy]. ~~value defined in this document~~ This is to be registered ~~will be published by IANA~~ according to the procedures in RFC 2911 [RFC2911] section 6.1~~. with the following path:~~

~~ftp.isi.edu/iana/assignments/ipp/attribute-values/notify-schemes-supported/~~

~~The registry entry will contain the following information:~~

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.

~~13.2.2~~14.2.2 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]. ~~type2 enum~~

attribute value defined in this document This is to be registered will be published by IANA according to the procedures in RFC 2911 [RFC2911] section 6.1. with the following path:

ftp.isi.edu/iana/assignments/ipp/attribute-values/operations-supported/

The registry entry will contain the following information:

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.

13.314.3 Attribute Registrations

The following table lists the attributes defined in this document. This is to be registered will be published by IANA according to the procedures in RFC 2911 [RFC2911] section 6.2. with the following path:

ftp.isi.edu/iana/assignments/ipp/attributes/

The registry entry will contain the following information:

Printer Description attributes:	Ref.	Section:
ippget-event-time-to-live (integer(0: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.

13.414.4 Status code Registrations

The following table lists the status codes defined in this document. This is to be registered will be published by IANA according to the procedures in RFC 2911 [RFC2911] section 6.6. with the following path:

ftp.isi.edu/iana/assignments/ipp/status-codes/

The registry entry will contain the following information:

Status codes:	Ref.	Section:
redirection-other-site (0x0300)	RFC NNNN	10.1

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

~~14~~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.

~~15~~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 s 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.

~~16~~17 References

[ipp-iig]
Hastings, T., Manros, C., Kugler, K, Holst H., Zehler, P., "Internet Printing Protocol/1.1: draft-ietf-ipp-implementers-guide-v11-0~~2~~3.txt, work in progress, ~~January 25~~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" ~~Specification"~~, <draft-ietf-ipp-not-spec-0~~6~~7.txt>, ~~February 24~~July 17, 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.

~~[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.

[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.

~~17~~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.

~~18~~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 ~~Specification~~ [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" ~~Specification"~~ 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.

~~19~~20 Full Copyright Statement

Acknowledgement

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