INTERNET-DRAFT
~~<draft-ietf-ipp-notify-poll-03.txt>~~<draft-ietf-ipp-notify-get-00.txt>

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

~~August 29,~~September 18, 2000

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

Abstract

The notification extension document [ipp-ntfy] 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. 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 notification extension 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.

The ‘ippget’ Delivery Method is a ‘pull and push’ Delivery Method. That is, the Printer saves Event Notification for a period of time and expects the Notification Recipient to fetch the Event Notifications (the pull part). The Printer continues to send Event Notifications to the Notification Recipient as Events occur (the push part) if the client has selected the option to wait for additional Event Notifications.

When a Printer supports this Delivery Method, it holds each Event Notification for an amount of time, called the *Event Notification Lease Time.*

When a Notification Recipient wants to receive Event Notifications, it performs an IPP operation called ‘Get-Notifications’, which this document defines. This operation causes the Printer to return all Event Notifications held for the Notification ~~Recipient along with information that tells the client when to perform this operation~~Recipient. If the Notification Recipient has selected the option to wait for additional ~~again.~~Event Notifications, the Printer continues sending Event Notifications to the Notification Recipient as additional Events occur.

The basic 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 ~~[ipp-mod]~~[RFC2911]
Internet Printing Protocol/1.1: Encoding and Transport ~~[ipp-pro]~~[RFC2910]
Internet Printing Protocol/1.1: Implementer's Guide [ipp-iig]
Mapping between LPD and IPP Protocols [RFC2569]
Internet Printing Protocol/1.0 & 1.1: IPP Event Notification 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 a new scheme named 'ippget' 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 "Event Notification Specification" document describes an extension to the IPP/1.0, IPP/1.1, and future versions. This extension allows a client to subscribe to printing related Events. Subscriptions are modeled as *Subscription Objects.* The Subscription Object specifies that when one of the specified *Event* occurs, the Printer sends an asynchronous *Event Notification* to the specified *Notification Recipient* via the specified *Delivery Method* (i.e., protocol). 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.

Table of Contents

1 Introduction

The notification extension document [ipp-ntfy] 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. 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 notification extension 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.

The ‘ippget’ Delivery Method is a ‘pull and push’ Delivery Method. That is, the Printer saves Event Notification for a period of time and expects the Notification Recipient to fetch the Event Notifications (the pull part). The Printer continues to send Event Notifications to the Notification Recipient as Events occur (the push part) if the client has selected the option to wait for additional Event Notifications.

When a Printer supports this Delivery Method, it holds each Event Notification for an amount of time, called the *Event Notification Lease Time.*

When a Notification Recipient wants to receive Event Notifications, it performs an IPP operation called ‘Get-Notifications’, which this document defines. This operation causes the Printer to return all Event Notifications held for the Notification ~~Recipient along with information that tells the client when to perform this operation~~Recipient. If the Notification Recipient has selected the option to wait for additional ~~again.~~ Event Notifications, the Printer the Printer continues to send Event Notifications to the Notification Recipient as Events occur.

2 Terminology

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

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 ~~[ipp-mod~~[RFC2911 section 13.1 on conformance terminology, most of which is taken from RFC 2119 [RFC2119].

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.

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” attributes 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 ~~MUST~~SHOULD choose a value for the address part of the ~~“notify-~~“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 ~~an~~the Event Notification Lease Time. The Printer MUST hold an Event Notification for its assigned Event Notification Lease Time~~ and MUST discard it when its Event Notification Lease Time expires~~. The Printer ~~MAY~~MUST assign the same Event Notification Lease Time to each Event Notification~~ or it MAY assign a different time~~.

When a Notification Recipient wants to receive Event Notifications, it performs the Get-Notifications operation, which causes the Printer to return all unexpired Event Notifications held for the Notification ~~Recipient along with two time-intervals.~~

Recipient. If the Notification Recipient has selected the option to wait for additional Event Notifications, the response to the ~~The first returned time-interval is the suggested time a Notification Recipient should wait before performing the Get-Notifications operation again. The second time-interval is the time that Event Notification Leases begin to expire for Event Notifications created after the Get-Notifications operation. A Notification Recipient SHOULD perform this operation at the suggested time and somewhat before the Event Notification Leases begin to expire.~~

~~The Notification Recipient identifies its own Event Notifications with a “notify-recipient-uri” Operation attribute in the request. It matches any Event Notifications associated with a Subscription Object~~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 has the same value as~~equals the “notify-recipient-uri” Operation ~~attribute of the request. To avoid~~ Attribute in the Get-Notifications operation.

~~getting Event Notification that belong to another Notification Recipient, a client SHOULD pick values for the “notify-recipient-uri” attribute that are unique, e.g. the client’s host address.~~

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.

~~The Printer may keep the channel open if the suggested time-interval is sufficiently short, but in any case the client performs a new~~When the Notification Recipient requests Event Notifications for per-Job Subscription Objects, the Notification Recipient typically performs the Get-Notifications operation ~~each time it wants more Event Notifications. Since the time interval between consecutive client requests is normally less than the Event Notification Lease Time, consecutive~~within a second of performing the Subscription Creation operation. Because the Printer is likely to save Event Notifications for several seconds, the ~~responses will normally contain some events that are identical. The youngest ones in the previous response will become the oldest in the next response. The client is expected to filter out these duplicates, which is easy to do because of the sequence number in each Event Notification. The reason for not removing the Event Notifications from the Printer with every Get-Notifications request, is so that multiple Notification Recipients can be polling the same Subscription Object and so the Get-Notification operation satisfies the rule of idempotency. The former is useful if someone is logged in to several desktops at the same time and wants to see the same events at both places. The latter is useful if the network loses the response.~~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
~~Is the Delivery Method REQUIRED or OPTIONAL for an IPP Printer to support?~~	~~OPTIONAL~~
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.
~~Is the Delivery Method initiated by the Notification Recipient (pull), or by the Printer (push)?~~	~~This Delivery Method is a pull.~~
5. Is the Delivery Method initiated by the Notification Recipient (pull), or by the Printer (push)?	This Delivery Method is a pull and a push.
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-	Section 5

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?	
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 causes the Printer to return all Event Notifications held for the Notification Recipient~~ along with information about when to perform this operation again~~.

A Printer MUST support this operation.

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

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

b) Whose associated Subscription Object's "notify-recipient-uri" attribute has a scheme value of 'ippget' AND

c) Whose Event Notification Lease Time has not yet expired AND

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

~~When a Printer performs this operation, it MUST also return two time-intervals:~~

~~a)the suggested time for a Notification Recipient to perform the Get-Notifications operation again.~~

~~b)the time at which the Printer will begin to discard Event Notifications that occur after this operation. This may be the Event Notification Lease Time (see section 5.2 for details).~~

~~Note: the Subscription Creation Operations also return these two time-intervals (see section 6).~~

The Printer MUST respond to this operation immediately with whatever Event Notifications it currently holds. ~~It MUST NOT~~If the Notification Recipient has selected the option to wait for additional ~~Events to occur before sending a response.~~

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).

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.

The Printer MUST accept the request in any state (see ~~[ipp-mod]~~[RFC2911] "printer-state" and "printer-state-reasons" attributes) and MUST remain in the same state with the same "printer-state-reasons".

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 ~~[ipp-mod]~~[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 ~~[ipp-mod]~~[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' 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 ~~[ipp-mod]~~[RFC2911] section 3.1.4.1.

Target:

The "printer-uri" (uri) operation attribute which is the target for this operation as described in ~~[ipp-mod]~~[RFC2911] section 3.1.5.

Requesting User Name:

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

"notify-recipient-uri" (url):

The client MUST supply this attribute. The Printer object MUST support this attribute. The Printer matches the value of this attribute (byte for byte with no case conversion) against the value of the "notify-recipient-uri" in each Subscription Object in the Printer. 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).

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.~~

application.

"notify-no-wait" (boolean):

The client MAY supply this attribute. The Printer object MUST support this attribute. 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 ~~[ipp-mod]~~[RFC2911] sections 13 and 3.1.6.

The Printer can return any status codes defined in ~~[ipp-mod].~~[RFC2911]. If the status code is not 'successful-', 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.

server-error-busy: The Printer is too busy to accept this operation. If the "suggested-ask-again-time-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" attribute before performing this operation again. If the "suggested-ask-again-time-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 with the uri specified by the "notify-ippget-redirect" Operation Attribute in the response..

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes as described in ~~[ipp-mod]~~[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: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 ~~suggested number of seconds that SHOULD elapse before the client performs the Get-Notifications operation again for these Subscription Objects. A client MAY perform the Get-Notifications operation at any time, and a Printer MUST respond with all unexpired Event Notifications. A~~ number of seconds that the Notification Recipient SHOULD wait before trying this operation again when

~~Notification Recipient waits until this time interval has elapsed in order to be a “good network citizen”. It is RECOMMENDED that the value of this attribute be 80% of the “begin-to-expire-time-interval” (see the next attribute) in order to give a Notification Recipient plenty of time to perform the Get-Notifications operation again before new Event Notifications expire.~~

a) ~~“begin-to-expire-time-interval” (integer(0:MAX)):~~the Printer returns the ‘server-error-busy’ status code OR
b) 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” (uri):

The value of this attribute is~~the minimum number of seconds that MUST elapse before Event Notification Leases begin to expire on Event Notifications produced by matching Subscriptions Objects after the Printer sends the Get-Notifications response. The Printer MUST discard an Event Notification when its EventNotification Lease has expired. That is, if the Printer performs the Get-Notifications operation before the time specified by the “begin-to-expire-time-interval” attribute returned in the previous operation, the Printer MUST still have all of the Event Notifications that have occurred since the previous operation. If the Printer assigns the same Event Notification Lease Time to all Event Notifications, the value of this attribute MUST equal the Event Notification Lease Time. If a Notification Recipient waits until after this time or even slightly less than this time,~~uri that the Notification Recipient MUST ~~expect to lose some Event Notifications.~~use for the Get-Notifications operation. This attribute is present in the Operation Attributes if and only if the status code has the value ‘redirection-other-site’.

~~“printer-up-time” (integer(0: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.~~

Group 2: Unsupported Attributes

See ~~[ipp-mod]~~[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 initial matched Event Notifications are all un-expired Event Notification associated with the matched Subscription Objects. If the Notification Recipient has selected the option to wait for additional Event Notifications, the Printer the subsequent Event Notifications in the response are Event Notifications associated with the matched Subscription Objects as the corresponding Event 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 Groups per Event Notification that the Printer is holding. After the initial burst of data, if the Notification Recipient has selected the 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 ~~[ipp-pro]~~[RFC2910] and may be encoded in any order. Note: the Get-Jobs response in ~~[ipp-mod]~~[RFC2911] acts as a model for encoding multiple groups of attributes.

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 following attributes.

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:MAX))	MUST	Printer
printer-current-time (dateTime)*	MUST	Printer

Source Value	**Sends**	**Source Object**
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 “printer-current-time” 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 following additional attributes.

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 following additional attributes.

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

~~6Extensions to Subscription Creation Operations~~

~~6.1Response~~

~~When a Subscription Creation Operation contains a "notify-recipient-uri" attribute and the scheme in its value is 'ippget', the response MUST contain two additional Operation Attributes that pertain to this Delivery Method. Note: Subscription Creation Operations include: Print-Job, Print-URI, Create-Job, Create-Job-Subscriptions and Create-Printer-Subscriptions.~~

~~Group 1: Operation Attributes~~

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

~~This attribute has the same meaning as the "suggested-ask-again-time-interval" attribute in the Get-Notifications operation except that it suggests when to perform the Get-Notifications operation for the first time on all Subscription Objects in the response whose "notify-recipient-uri" scheme is 'ippget'.~~

~~"begin-to-expire-time-interval" (integer(0:MAX)):~~

6 ~~This attribute has the same meaning as the "begin-to-expire-time-interval" attribute in the Get-Notifications operation except that it indicates when the Event Notification Lease begins to expire for~~

~~all Subscription Objects in the response whose “notify-recipient-uri” scheme is ‘ippget’.~~Additional Printer Description Attributes

6.1 begin-to-expire-time-interval” (integer(0:MAX))

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

The Printer MUST support this attribute if it supports this 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 this 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 performing the Get-Notifications operations can expect not miss any Event-Notifications, barring some unforeseen lack of memory space in the Printer.

7 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.

7.1 redirection-other-site (0x300)

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

8 Encoding

The operation-id assigned for the Get-Notifications operation is:

0x001C

and should be added to the next version of ~~[ipp-mod]~~[RFC2911] section 4.4.15 "operations-supported".

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

```
notification-attributes-tag = %x07        ; tag of 7
```

9 Conformance Requirements

If the Printer supports the 'ippget' Delivery Method, the Printer MUST:

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

2. support the Get-Notifications operation defined in section 5.

3. support the "begin-to-expire-time-interval" attribute defined in section 6.1.

4. support the "redirection-other-site" status code defined 7.1

10 IANA Considerations

There is nothing to register.

11 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.

12 Security Considerations

The IPP Model and Semantics document ~~[ipp-mod]~~[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.

13 References

~~[ipp-mod]~~
~~R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, “Internet Printing Protocol/1.1: Model and Semantics”, <draft-ietf-ipp-model-v11-06.txt>, March 1, 2000.~~

[ipp-ntfy]
R. Herriot, Hastings, T., Isaacson, S., Martin, J., deBry, R., Shepherd, M., Bergman, R., “Internet Printing Protocol/1.1: IPP Event Notification Specification”, <draft-ietf-ipp-not-spec-04.txt>, June 30, 2000.

~~[ipp-pro]~~
~~Herriot, R., Butler, S., Moore, P., Tuner, R., “Internet Printing Protocol/1.1: Encoding and Transport”, draft-ietf-ipp-protocol-v11-05.txt, March 1, 2000.~~

[rfc2026]
S. Bradner, “The Internet Standards Process -- Revision 3”, RFC 2026, October 1996.

[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.

[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.

14 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

~~Tom Hastings~~
~~Xerox Corporation~~
~~737 Hawaii St. ESAE 231~~

~~El Segundo, CA 90245~~

~~Phone: 310-333-6413~~
~~Fax: 310-333-5514~~
~~e-mail: hastings@cp10.es.xerox.com~~

~~Carl-Uno Manros~~
~~Xerox Corporation~~
~~701 Aviation Blvd.~~
~~El Segundo, CA 90245~~

~~Phone: 310-333-~~
~~Fax: 310-333-5514~~
~~e-mail: manros@cp10.es.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~~303-924-5337
FAX:
e-mail: harryl@us.ibm.com

15 Full Copyright Statement