INTERNET-DRAFT 1 ISSUES are highlighted like this.

<draft-ietf-ipp-notify-poll-~~00~~01.txt>

Robert Herriot
Xerox Corp.
Tom Hastings
Xerox Corp.
Carl-Uno Manros
Xerox Corp.
Harry Lewis
IBM, Corp.

~~March 8~~May 11, 2000

Internet Printing Protocol (IPP):
The '~~ipp~~ipp-get' Notification Polling Method

Status of this Memo

Abstract

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

When a Printer supports the '~~ipp~~ipp-get' delivery method, it holds each Event Notification for a certain length of time. The amount of time is called the ~~"~~event-~~lease ~~lease time~~".~~. A Printer may assign the same event-lease time to each Event Notification or different times. If a Notification Recipient does not want to miss Event Notifications, the time between consecutive pollings of Subscription objects must be less than the event-~~lease ~~lease time of the events that occur between pollings. The Get-Notifications request indicates whether the client wants to receive all pending event Notifications for ~~(1) any Subscription for which the client is the owner, (2) any Subscription associated with a Job, (3)~~\ any Subscription with a particular delivery-method URL~~, or (4) an identified set of Subscription objects~~. With the Get-Notifications operation, the Printer returns all existing Event Notifications along with two time intervals. One specifies the minimum time at which event-leases of future events of the type returned will begin to expire and the

other specifies the recommended interval for the client to wait before sending the next Get-Notifications operation. The second time interval is less than the first.

The Printer may keep the channel open if the recommended interval is sufficiently short, but in any case the client performs a new Get-Notifications operation each time it wants more Event Notifications. Since the time interval between consecutive client requests is normally less than the event-lease time, consecutive responses will normally contain some Event Notifications 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 full 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]
Internet Printing Protocol/1.1: Encoding and Transport [ipp-pro]
Internet Printing Protocol/1.1: Implementer's Guide [ipp-iig]
Mapping between LPD and IPP Protocols [RFC2569]
Internet Printing Protocol/1.0 & 1.1: 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 '~~ipp~~ipp-get' 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 defines OPTIONAL operations that allow a client to subscribe to printing related events. Subscriptions include "Per-Job subscriptions" and "Per-Printer subscriptions". Subscriptions are modeled as Subscription objects. Four other operations are defined for subscription objects: get attributes, get subscriptions, renew a subscription, and cancel a subscription.

Table of Contents

1 Introduction

IPP printers that support the OPTIONAL IPP notification extension [ipp-ntfy] either a) accept, store, and use notification subscriptions to generate Event Notification reports and implement one or more delivery methods for notifying interested parties, or b) support a subset of these tasks and farm out the remaining tasks to a Notification Delivery Service. The '~~ipp~~ipp-get' Event Notification delivery method specified in this document defines a Get-Notifications operation that may be used in a variety of notification scenarios. Its primary intended use is for clients that want to be Notification Recipients. However, the Get-Notifications operation may also be used by Notification Delivery Services for subsequent distribution to the Ultimate Notification Recipients.

When a Printer supports the '~~ipp~~ipp-get' delivery method, it holds each Event Notification for a certain length of time. The amount of time is called the ''event-lease time''. A Printer may assign the same event-lease time to each event or different times. If a Notification Recipient does not want to miss Event Notifications, the time between consecutive pollings of Subscription objects must be less than the event-lease time of the Event Notifications that occur between pollings. The Get-Notifications request indicates whether the client wants to receive all pending Event Notifications for ~~(1) any Subscription for which the client is the owner, (2) any Subscription associated with a particular Job, (3)~~ any Subscription with a particular notification recipient URL~~, or (4) an identified set of Subscription objects~~. With the Get-Notifications operation, the Printer returns all existing Event Notifications along with two time intervals. One specifies the minimum time at which event-leases of future events of the type returned will begin to expire and the other specifies the recommended interval for the client to wait before sending the next Get-Notifications operation. The second time interval is less than the first.

The Printer may keep the channel open if the recommended interval is sufficiently short, but in any case the client performs a new Get-Notifications operation each time it wants more Notifications. Since the time interval between consecutive client requests is normally less than the event-lease time, consecutive 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 Notification. The reason for not removing the Notifications from the Subscription object 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.

2 Terminology

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

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

OPTIONAL: if an implementation supports the extensions described in this document, it MAY support an OPTIONAL feature.

Notification Recipient - See [ipp-ntfy]

Subscription object - See [ipp-ntfy]

Ultimate Notification Recipient - See [ipp-ntfy]

3 Model and Operation

In the IPP Notification Model [ipp-ntfy], at most one ~~or more~~ Per-Job Subscription~~s~~ can be supplied in the Job Creation operation. ~~or~~ In addition one Per-Job Subscription can be supplied in ~~OPTIONALLY as subsequent~~ Create-Job-Subscription operations, and~~;~~ one Per-Printer Subscription can be supplied in the Create-Printer operation. The client that creates these Subscription objects becomes the owner of the Subscription object.

When creating each Subscription object, the client supplies the "~~notify-recipient~~notify-recipient-uri" (uri) attribute. The "~~notify-recipient~~notify-recipient-uri" attribute specifies both a single Notification Recipient that is to receive the Notifications when subsequent events occur and the URL's scheme specifies the method for Notification delivery that the IPP Printer is to use. For the Notification delivery method defined in this document, the scheme of the URL is '~~ipp~~ipp-get' and the host SHOULD be the client host's URL. In addition, the URL MAY contains a path to allow for applications to have a unique URL. Because the Get-Notifications operation uses the "notification-recipient-uri" to specify the events that it wants in the response, the Subscriber can partition events into suitable groups by associating a different URL with each group – the URLs may have the same host but different paths. If a Subscriber wants a friend to receive Event Notification via this delivery method, it can use the friend's URL as the "notification-recipient-uri". When the friend performs the Get-Notifications operation on the URL, it receives all pending the notifications, even those event caused by subscriptions owned by others.

~~ISSUE 1: The 'ipp' is a convenient reuse of 'ipp', but does it imply the existence of a print service at each client that is not a reality?~~

For most Notification delivery methods, a Printer sends Event Notifications to the delivery URL and the Printer does not perform any authentication or authorization with the receivers of the Event Notifications. For the Notification delivery method defined in this document, the client requests Event Notifications from the Printer via a Get-Notifications operation, and the Printer performs the same authentication and authorization as it does for the Get-Job-Attributes operation. That is, a Printer MAY allow a client to perform a Get-Notifications operation on any Subscription object or it MAY restrict access as follows. Any client that is authenticated (1) as an operator or administrator or (2) as the owner of the Subscription object can initiate a Get-Notifications operation for that Subscription object.

Because a Printer has to wait for a client to request Event Notifications for the '~~ipp~~ipp-get' delivery method, any Printer that supports the '~~ipp~~ipp-get' notification delivery method MUST hold each Event Notification at least for the event-lease time that it advertises to clients. With this rule, a single user can login at different places, say his/her office, the lab, and/or several desktops in the same room, and receive the same Event Notifications from a single Subscription object. In addition, a client that gets no response, perhaps because of a network failure, can perform the Get-Notifications operations two or more times in quick succession and get the same results except for a few newly arrived Event Notifications and a few old Event Notifications whose event-leases have expired.

The event-lease time assigned to Event Notifications MAY be different for each implementation. Furthermore, a particular implementation MAY assign different event-lease times to each Event Notification. If a Printer assigns different event-lease times to each Event Notification, the event-lease time returned with Get-Notifications MUST be a value that ensures a client will not miss future Event Notifications.

The client issues a Get-Notifications Printer operation in order to initiate the delivery of the pending Notifications held by the Printer for the Subscription objects requested. In this operation, t~~T~~he client specifies the "notification-recipient-uri" attribute and the Printer returns all pending Event Notifications associated with Subscription objects whose "notification-recipient-uri" attribute matches the "notification-recipient-uri" attribute specified in the operation.

~~can indicate in the Get-Notifications request whether it wants to receive all pending Notifications for:~~

~~1)any existing Subscription objects for which the client is the owner,~~

~~2)any existing Subscription objects whose notification-recipient is a specified URL~~

~~3)any existing Subscription objects associated with a job-id or~~

~~4)particular Subscription object(s) (for which it MUST be the owner or have read-access rights).~~

~~In any case, the Notifications are returned in a response to the Get-Notifications request.~~

If the client requests a persistent channel, then the Printer MAY keep the channel open. Either the client or the IPP Printer can disconnect the HTTP connection.

4 Get-Notifications operation

This REQUIRED operation allows the client to request that pending Event Notifications be delivered as a response to this request. The client MUST be the owner or have read-access rights of the Subscription objects that are involved and the delivery method specified when the Subscription objects were created MUST be '~~ipp~~ipp-get'.

This operation returns all pending Event Notifications specified by the "notify-recipient-uri" operation attribute. To help a client know when to perform this operation again, the Printer returns both the event-lease time and the suggested-ask-again time in the following operations:

a) Job Creation, Create-Printer-Subscription and Create-Job-Subscription operation if the scheme of the "notify-recipient-uri" operation attribute is 'ipp-get'.

b) All Get-Notifications operation

~~When the Printer creates a Subscription Object, either with a Job Creation operation or with a Create-Printer-Subscription or Create-Job-Subscription operation and a subscription object contains the 'ipp' value for the "notify-recipient" operation attribute, the Printer returns the event-lease time for Events and the recommended time interval before the client to performs the next Get-Notifications operation.~~ The client SHOULD perform a Get-Notifications operation at about the suggested-ask-again time ~~recommended interval~~ and if the Printer receives the Get-Notifications before the event-lease time has elapsed, it MUST have all of the Notifications since the previous Get-Notification operation or the Subscription object creation, whichever was most recent.

~~Issue 2: Now that the Get-Notification operation does not affect the Event Notifications in the Printer, it should not require write access to access them.~~

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

Access Rights: The authenticated user (see [ipp-mod] section 8.3) performing this operation must either be the Subscription object owner (as determined when the Subscription object was created by the Job Creation operation, Create-Job-Subscription, or Create-Printer-Subscription operations) or an operator or administrator of the Printer object (see [ipp-mod] 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.

~~Issue 3: Is it possible for this operation to have an option that causes it to delay completing its response? It would initially returns all existing Event Notifications. Then it would return additional notifications as they occur for some period of time. The client would receive these Event Notifications as they occur. The question is whether http servers or proxies would behave in this manner so that the client would get the Event Notifications without delay after they are sent by the http server? It has been suggested that the Printer send each burst of Event Notifications be in a separate message body where each message body is part of a multipart MIME content-type.~~

4.1 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] section 3.1.4.1.

Target:
The "printer-uri" (uri) operation attribute which is the target for this operation as described in [ipp-mod] 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] section 8.3.

"notification-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 against the value of the "notification-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. If there are matches, the IPP Printer MUST return all accumulated Event Notifications associated with Subscription objects that contain the matched "notification-recipient-uri" attribute.

~~The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It is a URL that identifies one or more Subscription objects for which Event Notifications are being requested. If the client supplies this attribute, but no notification-recipients are found, the IPP Printer MUST return the 'client-error-not-found' status code. If some are found and others are not, the ones that are not found are return in the Unsupported Attributes. By definition, if a notification-recipient URL exists, there must be at least one Subscription object.~~

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 if it wants.~~. If a client uses such a URL as the value of this attribute, the client gets event Notifications for all Subscription objects whose "notification-recipient" is the specified URL. This mechanism is more general than getting all subscriptions owned by a client.~~ It allows clients who didn't subscribe to get Event Notifications without knowing job-ids or subscription-ids.

~~ISSUE 4: The "notification-recipient" option allows a client to group multiple Subscription-ids with a single URL. A client might decide to use the same URL for all subscriptions for a user, or it might have a separate URL for each client program. In addition a client might use an URL belonging to some other known user and let that user access Event Notifications using that URL. Is the above option useful?~~

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

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

~~"job-ids" (1setOf integer(1:MAX)):~~

~~The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It is an integer value that identifies one or more job-ids. These job-ids identify the Subscription~~

~~objects for which Event Notifications are being requested. If the client supplies this attribute, but no Jobs are found, the IPP Printer MUST return the 'client-error-not-found' status code. If some are found and others are not, the ones that are not found are returned in the Unsupported Attributes. It is not an error if there are no Subscription objects for a Job.~~

~~If the client supplies none of the last three attributes described for this operation, then the IPP Printer returns Event Notifications for all Subscription objects for which the client is the owner and the "notify-recipients" attribute is 'ipp'. It is not an error if there are currently no Subscription objects for this client; the response then contains no Notifications.~~

~~ISSUE 5: Does the mechanism described in the above paragraph describe a useful option that "notification-recipient" alone cannot do? Should this case be an error instead?~~

~~If a client supplies more than one of the last three attributes described for this operation, the Printer returns Event Notifications for all Subscription objects specified by all attributes. If these attributes describe duplicate Event Notifications, the Printer MAY remove them.~~

~~ISSUE 6: Is it better if "notification-recipient" is the only way to request Event Notification? If so, this behaves more like other notification delivery methods where a recipient receives those and only those events with its delivery URL. Furthermore, if there is a single mechanism of "notification-recipient" for a client to specify Event Notifications, a Printer can better optimize event-leases because it knows which events will be accessed together. If client can specify subscription-ids, each request can contain a different mix of subscription-ids.~~

4.2 Get-Notifications Response

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

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] sections 13 and 3.1.6.

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod] section 3.1.4.2. The Printer uses the values of "notify-charset" and "notify-natural-language", respectively, from one of the Subscription objects associated with the Event Notifications in this response.

Normally, the values of these attributes is the same in all Subscriptions. If they are not, the Printer picks the values from one Subscription object to put in 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 operation level natural-language. The Printer’s choice of charset is critical because a bad choice may leave it unable to send some 'text' and 'name' values accurately.

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

The value of this attribute is the ~~recommended~~ suggested number of seconds that SHOULD elapse before the client performs this operation again for these Subscription objects. A client MAY perform this operation at any time, and a Printer MUST respond with all existing Notifications. A client observes this value in order to be a "good network citizen". The value that a Printer returns for this attribute MUST NOT exceed 80% of the "event-lease-time-interval" in order to give a client plenty of time to perform another Get-Notifications operation before the event-lease of the oldest Event Notifications expire.

"event-lease-time-interval" (integer(0:MAX)):

The value of this attribute is the minimum number of seconds until the event-lease expiration time for all future Event Notifications associated with the Subscription objects generating the requested Event Notifications. Thus this number is the maximum number of seconds that elapses before this client SHOULD issue this operation again for these Subscription objects. A Printer MUST preserve all Notifications that occur for the number of seconds specified by this attribute starting at the time it is sent in a response. A client MAY perform this operation at any time, and a Printer MUST respond with all existing Event Notifications. If a Printer receives this operation after this time interval, it MAY have discarded some Notifications since the last response.

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

The value of this attribute is the Printer’s "printer-up-time" attribute. Because each Event Notification also contains the value of this attribute when the event occurred, the value of this attribute lets a client know when each Event Notification occurred relative to the time of this response.

Group 2: Unsupported Attributes

See [ipp-mod] 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 object responds with one Event Notification per Group for each supplied Event Notification. Each Event Notification MUST ~~that~~ meets the criteria specified by the request.(see [ipp-ntfy]). Each Event Notification Group MUST start with an 'event-notification-attributes-tag', which is the tag that begins an Event Notification Group (see the section "Encodings of Additional Attribute Tags" in [ipp-ntfy]).

This group includes the following attributes from the section on " Content of Machine Consumable Event Notifications" in [ipp-ntfy]. They are encoded using the IPP rules for encoding attributes [ipp-pro] and they may be encoded in any order. Note: the Get-Jobs response acts as a model for encoding multiple groups of attributes.

Table 1 and Table 3 contains the following information

a) Attribute: the name of the attribute to include from the section on " Content of Machine Consumable Event Notifications" in [ipp-ntfy].

b) Condition: the condition for the attribute to be present. The value

   i) " means that the attribute MUST be present in all Event Notifications.

   ii) 'conditional' means the attribute MUST be present if the Printer supports the attribute

   iii) 'progress' means the attribute MUST be present for 'job-progress' or 'job-completed' events only.

For a Event Notification for job and printer events, the Printer includes the following attributes.

Table 1 – REQUIRED Attributes in all IPP Event Notification Content

Attribute	Condition
subscription-request-id (integer (0:MAX))	
notify-text (text)	
notify-text-format (mimeMediaType)	
printer-uri (uri)	
trigger-event (type2 keyword)	
printer-up-time (integer(MIN:MAX))	
printer-current-time (dateTime)	conditional

For Event Notification for job events, the Printer includes the following additional attributes.

Table 2 – REQUIRED Attributes in all IPP Event Notification Content

Attribute	Condition
job-id (integer(1:MAX))	

Attribute	Condition
job-state (type1 enum)	
job-state-reasons (1setOf type2 keyword)	
job-impressions-completed (integer(0:MAX))	progress

For Event Notification for printer events, the Printer includes the following additional attributes.

-**Table 3 – REQUIRED Attributes in all IPP Event Notification Content**

Attribute	Condition
printer-state (type1 enum)	
printer-state-reasons (1setOf type2 keyword)	
printer-is-accepting-jobs (boolean)	

5 Extensions to Print-Job, Print-URI, Create-Job, Create-Printer-Subscription and Create-Printer-Subscription

5.1 Response

When Print-Job, Print-URI or Create-Job contains a "notify-recipient-uri" attribute and the scheme in its value is 'ipp-get', the response contains two additional Operation Attributes that pertain to subscriptions.

When Create-Job-Subscription or Create-Printer-Subscription operation contains a "notify-recipient-uri" value whose scheme is 'ipp-get', the response contains two additional Operation Attributes that pertain to subscriptions.

~~When Print-Job, Print-URI or Create-Job contains a "job-notify" attribute and the "notify-recipient" is 'ipp', the response contains two additional Operation Attributes that pertain to subscriptions.~~

~~When Create-Job-Subscription or Create-Printer-Subscription operation contains a "notify-recipient" that is 'ipp', the response contains two additional Operation Attributes that pertain to subscriptions.~~

Group 1: Operation Attributes

" suggested-ask-again~~recommended~~-time-interval" (integer(0:MAX)):
The value of this attribute is the suggested ~~recommended~~ number of seconds that SHOULD elapse before the client SHOULD perform the Get-Notification operation for the first time with any Subscription objects returned with this job. A client MAY perform the Get-Notification operation at

any time, and a Printer MUST respond with all existing Notifications. A client observes this value in order to be a "good network citizen". The value that a Printer returns for this attribute MUST NOT exceed 80% of the "event-lease-time-interval" in order to give a client plenty of time to perform another Get-Notifications operation before the event-lease of the oldest events expire.

"event-lease-time-interval" (integer(0:MAX)):

The value of this attribute is the minimum number of seconds until the event-lease expiration time for all future Event Notifications associated with the Subscription objects generating the requested Event Notifications. Thus this number is the maximum number of seconds that elapses before a client SHOULD perform the Get-Notification operation for the first time with any Subscription objects returned with this job. A Printer MUST preserve all Notifications that occur for the number of seconds specified by this attribute starting at the time it is sent in a response. A client MAY perform the Get-Notification operation at any time, and a Printer MUST respond with all existing Event Notifications. If a Printer receives a Get-Notification operation after this time interval, it may have discarded some Notifications since the last response.

6 Encoding

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

0x00??

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

ISSUE: what is the value?

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

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

7 IANA Considerations

There is nothing to register.

8 Internationalization Considerations

With the '~~ipp~~ipp-get' method defined in this document, the client cannot request the Human Consumable form by supplying the "notify-format" operation attribute (see [ipp-ntfy]). The only supported value for this delivery method is "application/ipp". Therefore, the IPP Printer does not have to perform any localization with this notification delivery method. However, the client when it receives the Get-Notifications response 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.

9 Security Considerations

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

10 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]
Isaacson, S., Martin, J., deBry, R., Hastings, R. Herriot, T., Shepherd, M., Bergman, R., "Internet Printing Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-03.txt>, May 11, 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.

11 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

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

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

12 Full Copyright Statement