INTERNET-DRAFT 5 ISSUES are highlighted like this.

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

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

~~February 2~~March 6, 2000

Internet Printing Protocol/1.1: **The '~~ipp-notify-poll~~ipp' Notification Polling Method**

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-notify-poll~~ipp' 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-notify-poll~~ipp' delivery method, it ~~queues~~ holds each Notification event~~s~~ ~~for a window of time for each Subscription object~~ for a certain length of time. The amount of time is called the "lease time" and it is the same for all events in a Printer. If a Notification Recipient does not want to miss events, the time between consecutive pollings of ~~Notification Recipients poll these~~ Subscription objects ~~at the rate specified by the time window~~must be less than the lease time. The Get-Notifications request indicates whether the client wants to receive all pending events Notifications for (1) any Subscription for which the client is the owner, (2) any Subscription associated with a Job or (~~2~~3) a particular Subscription object. ~~The Get-Notifications operation retrieves all pending Notifications that occurred for an interval of time in the past for the requested Subscription objects.~~ With the Get-Notifications operation, t~~T~~he Printer returns ~~the~~ all ~~pending~~ existing Notifications along with two time intervals. One specifies the length of the lease for all future events and the other specifies the recommended interval to wait to the next Get-Notifications operation. The second time interval is less than the first. ~~that~~

~~specify the next time window: one is the minimum interval that the client should wait before performing another Get-Notifications on the subscription-id and the other is the maximum interval that the Printer is guaranteed to keep any new Notifications associated with the subscription-id.~~

The Printer may keep the channel open if the ~~minimum~~ 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 lease time,~~will be making Get-Notification requests before the time window expires,~~ consecutive responses will normally contain some events that are identical. ~~the Printer will, on occasion, return the same event Notification in two successive responses.~~ The later ones in the previous response will become the earliest 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 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' 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-notify-poll~~ipp' 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-notify-poll~~ipp' delivery method, it ~~queues~~ holds each Notification event~~s~~ for a certain length of time ~~for a window of time for each Subscription object~~\. The amount of time is called the "lease time" and it is the same for all events in a Printer. If a Notification Recipient does not want to miss events, the time between consecutive pollings of Subscription objects must be less than the lease time. ~~Notification Recipients poll these Subscription objects at the rate specified by the time window.~~ The Get-Notifications request indicates whether the client wants to receive all pending events Notifications for (1) any Subscription for which the client is the owner, (2) any Subscription associated with a particular Job or (~~2~~3) a particular Subscription object. With the Get-Notifications operation, the Printer returns all existing Notifications along with two time intervals. One specifies the length of the lease for all future events and the other specifies the recommended interval to wait to the next Get-Notifications operation. The second time interval is less than the first. ~~The Get-Notifications operation retrieves all pending Notifications that occurred for an interval of time in the past for the requested Subscription objects. The Printer returns all pending Notifications along with two time intervals that specify the next time window: one is the minimum interval that the client should wait before performing a Get-Notifications on the subscription-id and the other is the maximum interval that the Printer is guaranteed to keep any new Notifications associated with the subscription-id~~.

The Printer may keep the channel open~~-~~ if the recommended interval is sufficiently short~~if the minimum 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 lease time, consecutive responses will normally contain some events that are identical. ~~will be making Get-Notification requests before the time window expires, the Printer will, on occasion, return the same event Notification in two successive responses.~~ The later ones in the previous response will become the earliest 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. ~~This~~ The former is useful if ~~you~~ someone is ~~are~~ 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], one or more Per-Job Subscriptions can be supplied in the Job Creation operation or OPTIONALLY as subsequent Create-Job-Subscription operations; 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" (uri) attribute. The "notify-recipient" attribute specifies both a single Notification Recipient that is to receive the Notifications when subsequent events occur and the method for Notification delivery that the IPP Printer is to use. For the '~~ipp-notify-poll~~ipp' Notification delivery method defined in this document, there is no notify-recipient because the Printer waits for one or more clients to ask for Notifications from a Subscription object rather than sending them. Rather, 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. Therefore, any Printer that supports the '~~ipp-notify-poll~~ipp' notification delivery method MUST ~~queue~~ hold each event Notification~~s~~ for the lease time that it advertises to clients ~~a sliding window of time for each Subscription object~~. Thus 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.

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. The client can indicate in the Get-Notifications request whether it wants to receive all pending Notifications for

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

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

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

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

If the client requests a persistent channel~~ and if the Printer has returned minimum intervals that are sufficiently short~~, then the Printer MAY keep~~s~~ the channel open. Either the client or the IPP Printer can disconnect the HTTP connection.

~~ISSUE 01: Should it be possible for a client to ask for the Per-Job Subscriptions for a particular job using a "job-id", instead of the subscription-id, which currently isn't returned by a Job Creation operation?~~

4 Get-Notifications operation

This REQUIRED operation allows the client to request that pending Notifications be delivered as a response to this request. The client MUST be the owner or have write-access rights of the Subscription objects that are involved and the delivery method specified when the Subscription objects were created MUST be ~~ipp-notify-poll~~ipp'. 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-notify-poll~~ipp' value for the “notify-recipient” operation attribute, the Printer returns the lease time for Events and the recommended time interval before the client to performs the next Get-Notifications operation.~~a minimum and maximum interval in the response.~~ The client SHOULD perform a Get-Notifications operation ~~after~~ at about the ~~minimum~~ recommended interval and if the Printer receives the Get-Notifications before the ~~maximum~~ lease time ~~interval~~ has elapsed, it MUST have all of the Notifications ~~that has occurred since the~~since the previous Get-Notification operation or the Subscription object ~~was created~~creation, whichever was most recent.

~~ISSUE 02: Is there anything useful that we could define for the rest of the "notification-recipient" (uri) attribute, since there is no recipient address needed after the 'ipp-notify-poll://' since the recipient(s) poll?~~

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 02.1: 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 notification events as they occur. The question is whether http servers or proxies would behave in this manner.

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" (url):

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.

If the client does not supply this attribute, the "jobs-ids" attribute and the "subscription-ids" attribute, 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.

If a client supplies this attribute and the "subscription-ids" attribute, the Printer returns event Notifications for all Subscription objects specified by both attributes. If the "subscription-ids" attributes contains values that are implied by the "job-id", the Printer MAY remove duplicates.

Note: this attribute allows a subscribing client to pick urls that are unique, e.g. the client’s own url or a friends url, which in both cases is likely the url of the person’s host. If a client uses such a url as the value of this attribute, the client gets events 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 events without knowing job-ids or subscription-ids.

ISSUE 02.5: 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.

If the client does not supply this attribute the "job-ids" attribute and the "notification" attribute, 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-notify-poll~~ipp'. It is not an error if there are currently no Subscription objects for this client; the response then contains no Notifications.~~.~~

"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 return in the Unsupported Attributes. It is not an error if there are no Subscription objects for a Job.

If the client does not supply this attribute, the "subscription-ids" attribute and the "notification-recipients" attribute, 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 02.6: Does the above paragraph describe a useful option that notification-recipient 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 attribute describe duplicate notification events, the Printer MAY remove them.

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.

"~~minimum~~recommended-time-interval" (integer(0:MAX)):

The value of this attribute is the ~~minimum~~ recommended number of seconds that SHOULD elapses before the client performs this operation again for these subscription-ids. A client MAY perform this operation at any time, and a Printer MUST respond with all ~~pending~~ 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 "lease-time-interval" in order to give a client plenty of time to perform another Get-Notifications operation before the lease of the oldest events expire.

"~~maximum~~lease-time-interval" (integer(0:MAX)):

The value of this attribute is the minimum number of seconds that the Printer will retain all future events. Thus this number is the maximum number of seconds that ~~SHOULD~~ elapses before this client SHOULD issue this operation again for these subscription-ids. 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 ~~pending~~ existing Notifications. If a Printer receives this operation after this time interval, it ~~SHOULD~~ MAY have discarded some Notifications since the last response.

~~ISSUE 04 - Or MUST the Printer discard events that occurred earlier than the sliding time window specified by the difference between these two values? Otherwise, the clients may get back a lot of duplicate events on subsequent requests.~~

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: Notification Attributes

The Printer object responds with one event Notification per Group for each pending Notification that meets the criteria specified by the subscription-ids attribute and requesting user name.(see [ipp-ntfy]).

5 Extension 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 “job-notify” attribute and the “notify-recipient” is '~~ipp-notify-poll~~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-notify-poll~~ipp', the response contains two additional Operation Attributes that pertain to subscriptions.

Group 1: Operation Attributes

"~~minimum~~recommended-time-interval" (integer(0:MAX)):

The value of this attribute is the ~~minimum~~ recommended number of seconds that ~~SHOULD~~ elapses before the client SHOULD perform~~s~~ the Get-Notification operation for the first time with any subscription-ids returned with this job. A client MAY perform the Get-Notification operation at any time, and a Printer MUST respond with all ~~pending~~ 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 "lease-time-interval" in order to give a client plenty of time to perform another Get-Notifications operation before the lease of the oldest events expire.

~~ISSUE 05: if we don’t want to have Job Creation operations return subscription id’s, then allow a "job-ids" operation attribute in the Get-Notifications request in addition to the "subscription-ids" operation attribute.~~

"~~maximum~~lease-time-interval" (integer(0:MAX)):

The value of this attribute is the minimum number of seconds that the Printer will retain all future events. Thus this number is the maximum number of seconds that ~~SHOULD~~ elapses before a ~~Printer~~ client SHOULD ~~receives~~ perform the Get-Notification operation for the first time with any subscription-ids 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 pending 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".

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

Instead of defining a new object attribute tag, a Generic Object attributes tag is defined that is used for all new objects, such as Subscription objects, etc. Then this one new tag can also be used for the Get-Notifications response Group 3 tag and subsequent groups in section 4.2:

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

7 IANA Considerations

~~IANA will be asked to register this 'ipp-notify-poll' notification delivery scheme.~~There is nothing to register.

8 Internationalization Considerations

With the '~~ipp-notify-poll~~ipp' method defined in this document, the client cannot request the Human Consumable form by supplying the "notify-~~text-~~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 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/1.0: Model and Semantics", <draft-ietf-ipp-model-v11-04.txt>, June, 1999.

[ipp-ntfy]
Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-01.txt>, October 14, 1999.

[ipp-pro]
Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and Transport", draft-ietf-ipp-protocol-v11-03.txt, June, 1999.

[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 Author's Addresses

Robert Herriot
3400 Hillview Ave., Bldg #1
Palo Alto, CA 94304

Phone: 650-813-7696
Fax: 650-813-6860
Email: rherriot@pahv.xerox.com

12 Full Copyright Statement