

INTERNET-DRAFT  
<[draft-ietf-ipp-notify-poll-01.txt](#)>

Robert Herriot  
Xerox Corp.  
Tom Hastings  
Xerox Corp.  
Carl-Uno Manros  
Xerox Corp.  
Harry Lewis  
IBM, Corp.  
July 7, 2000

Internet Printing Protocol (IPP):  
The 'ipp-get' Delivery Method

Copyright (C) The Internet Society (2000). All Rights Reserved.

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of \[rfc2026\]](#). Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed as  
<http://www.ietf.org/shadow.html>.

## 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 'ipp-get' delivery method.

The 'ipp-get' Delivery Method is a 'pull' Delivery Method. That is, the Printer saves Event Notification for a period of time and expects the Notification Recipient to fetch the 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 again.



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: 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 '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.

Herriot, et al.

Expires: January 6, 2001

[page 3]

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

<a href="#">1</a>	Introduction .....	<a href="#">6</a>
<a href="#">2</a>	Terminology .....	<a href="#">6</a>
<a href="#">3</a>	Model and Operation .....	<a href="#">7</a>
<a href="#">4</a>	Get-Notifications operation .....	<a href="#">8</a>
<a href="#">4.1</a>	GET-NOTIFICATIONS REQUEST.....	<a href="#">11</a>
<a href="#">4.2</a>	GET-NOTIFICATIONS RESPONSE.....	<a href="#">11</a>
<a href="#">5</a>	<b>Extensions to Print-Job, Print-URI, Create-Job, Create-Printer-Subscription and Create-Printer-Subscription.....</b>	<a href="#">16</a>
<a href="#">5.1</a>	RESPONSE.....	<a href="#">16</a>
<a href="#">6</a>	Encoding .....	<a href="#">17</a>
<a href="#">7</a>	IANA Considerations .....	<a href="#">17</a>
<a href="#">8</a>	Internationalization Considerations .....	<a href="#">17</a>
<a href="#">9</a>	Security Considerations .....	<a href="#">17</a>
<a href="#">10</a>	References .....	<a href="#">18</a>
<a href="#">11</a>	Authors' Addresses .....	<a href="#">18</a>
<a href="#">12</a>	Full Copyright Statement .....	<a href="#">19</a>





## **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 'ipp-get' delivery method.

The 'ipp-get' Delivery Method is a 'pull' Delivery Method. That is, the Printer saves Event Notification for a period of time and expects the Notification Recipient to fetch the 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 again.

## **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 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" attribute has the scheme "ipp-get", the client is requesting that the Printer use the 'ipp-get' Delivery Method for the Event Notifications associated with the new Subscription Object. The client **MUST** choose a value for the address part of the "notify-recipient-uri" attribute that uniquely identifies the Notification Recipient.

When an Event occurs, the Printer **MUST** generate an Event Notification and **MUST** assign it an 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** 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.

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.

Herriot, et al.

Expires: January 6, 2001

[page 7]

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 whose "notify-recipient-uri" attribute has the same value as the "notify-recipient-uri" Operation attribute of the request. To avoid 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. 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 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 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.

#### **4 General Information**

According to the notification extension document [[ipp-ntfy](#)], this document MUST contain the following information:

- 1.The URL scheme name for the Delivery Method is: 'ipp-get'
- 2.Printer support for this delivery method is OPTIONAL.
- 3.For Event Notification content, a Printer MUST IPP with one new operation.



4. Several Event Notifications MAY be combined into a compound Event Notification. See [section 5](#).
5. The Notification Recipient MUST initiate the Delivery Method
6. The Delivery Method sends Machine Consumable Event Notifications.
7. The representation and encoding for each value MUST be the same as for IPP (see [section 5](#)).
8. In the Event Notification content, a Printer MUST send all attributes specified in [section 5](#).
9. Frequently occurring Events NEED NOT be moderated because the Delivery Method is a 'pull' Delivery Method. An implementation of the Get-Notifications operation SHOULD consider how often it recommends a Notification Recipient to poll again.
- [10](#). This Delivery Method has the same latency and reliability as the underlying HTTP transport.**
- [11](#). This Delivery Method has the same security aspects as the underlying HTTP transport.**
- [12](#). This Delivery Method has no content length restrictions.**
- [13](#). There are no additional values that a Printer MUST send in a Notification content.**
- [14](#). There are no additional Subscription Template and/or Subscription Description attributes.**
- [15](#). There are no additional Printer Description attributes.**

## **[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 'ipp-get' 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 wait for additional Events to occur before sending a response.

The 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: 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](#)] [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](#)] 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](#)] [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](#).

"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.

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.

### **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](#)] sections [13](#) and [3.1.6](#).

Herriot, et al.

Expires: January 6, 2001

[page 11]

The Printer can return any status codes defined in [[ipp-mod](#)]. 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.

#### 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 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.

#### "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 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.

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

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

Herriot, et al.

Expires: January 6, 2001

[page 12]

after the Printer sends the Get-Notifications response. The Printer MUST discard an Event Notification when its Event Notification 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, the Notification Recipient MUST expect to lose some Event Notifications.

"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](#)] [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 matched Event Notifications are all un-expired Event Notification associated with the matched Subscription Objects. 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](#)] and may be encoded in any order. Note: the Get-Jobs response in [[ipp-mod](#)] 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 1 - 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
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 2 - 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 3.

Table 3 - 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 4 - Additional Attributes in Event Notification Content for Printer Events

Source Value	Sends	Source Object
printer-state (type1 enum)	MUST	Printer
printer-state-reasons (1setOf type2 keyword)	MUST	Printer
printer-is-accepting-jobs (boolean)	MUST	Printer

## **6 Extensions to Subscription Creation Operations**

### **6.1 Response**

When a Subscription Creation Operation contains a "notify-recipient-uri" attribute and the scheme in its value is 'ipp-get', 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 'ipp-get'.

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

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 'ipp-get'.



## **7 Encoding**

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

```
0x001C
```

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:

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

## **8 IANA Considerations**

There is nothing to register.

## **9 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.

## **10 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 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.

## **11 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.

## **12 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](mailto:robert.herriot@pahv.xerox.com)

Tom Hastings

Herriot, et al.

Expires: January 6, 2001

[page 18]

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](mailto: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](mailto: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](mailto:harryl@us.ibm.com)

### **13 Full Copyright Statement**

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK

FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT

Herriot, et al.

Expires: January 6, 2001

[page 19]

INTERNET-DRAFT

IPP: The 'ipp-get' Delivery Method

July 7, 2000

LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT  
INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR  
FITNESS FOR A PARTICULAR PURPOSE.

