

INTERNET-DRAFT

[ietf-ipp-notify@ietf.org](mailto:ietf-ipp-notify@ietf.org)

Robert Herriot

Xerox Corp.

Henrik Holst

i-data international a/s

Tom Hastings

Xerox Corp.

Carl-Uno Manros

Xerox Corp.

July 13, 2000

**Internet Printing Protocol (IPP):  
The 'mailto:' Notification 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/1id-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. 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 'mailto' delivery method.

For this Delivery Method, when an Event occurs, the Printer immediately sends an Event Notification via an email message to the Notification Recipient specified in the Subscription Object. The message body of the email consists of Human Consumable text and is not intended to be parsed by a machine.

The Notification Recipient receives the Event Notification in the same way as it receives any other email message.



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 (IPP): 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 also 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.

Herriot, et al.

Expires: January 13, 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. 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="#">7</a>
<a href="#">2</a>	Terminology .....	<a href="#">7</a>
<a href="#">3</a>	Model and Operation .....	<a href="#">7</a>
<a href="#">4</a>	General Information .....	<a href="#">9</a>
<a href="#">5</a>	Subscription Template Attributes .....	<a href="#">11</a>
<a href="#">5.1</a>	Additional Subscription Template Attributes .....	<a href="#">11</a>
<a href="#">5.1.1</a>	notify-mailto-text-only (boolean).....	<a href="#">11</a>
<a href="#">5.2</a>	Additional Information about Subscription Template Attributes .....	<a href="#">12</a>
<a href="#">5.2.1</a>	notify-recipient-uri (uri).....	<a href="#">12</a>
<a href="#">5.2.2</a>	notify-user-data (octetString(63)).....	<a href="#">12</a>
<a href="#">6</a>	Event Notification Content .....	<a href="#">13</a>
<a href="#">6.1</a>	Headers .....	<a href="#">13</a>
<a href="#">6.1.1</a>	'Date' header.....	<a href="#">13</a>
<a href="#">6.1.2</a>	'From' header.....	<a href="#">13</a>
<a href="#">6.1.3</a>	'Subject' header.....	<a href="#">14</a>
<a href="#">6.1.4</a>	'Sender' header.....	<a href="#">14</a>
<a href="#">6.1.5</a>	'Reply-to' header.....	<a href="#">15</a>
<a href="#">6.1.6</a>	'To' header.....	<a href="#">15</a>
<a href="#">6.1.7</a>	'Content-type' header.....	<a href="#">16</a>
<a href="#">6.2</a>	Message Body .....	<a href="#">16</a>
<a href="#">6.2.1</a>	Event Notification Content Common to All Events.....	<a href="#">17</a>
<a href="#">6.2.2</a>	Additional Event Notification Content for Job Events.....	<a href="#">19</a>
<a href="#">6.2.3</a>	Additional Event Notification Content for Printer Events...	<a href="#">20</a>
<a href="#">6.3</a>	Examples .....	<a href="#">21</a>
<a href="#">6.3.1</a>	Job Event Example.....	<a href="#">21</a>
<a href="#">6.3.2</a>	Printer Event Example.....	<a href="#">23</a>
<a href="#">6.3.3</a>	Printer Event Example (localized to Danish).....	<a href="#">24</a>
<a href="#">7</a>	Conformance Requirements .....	<a href="#">25</a>
<a href="#">8</a>	IANA Considerations .....	<a href="#">25</a>
<a href="#">9</a>	Internationalization Considerations .....	<a href="#">26</a>
<a href="#">10</a>	Security Considerations .....	<a href="#">26</a>
<a href="#">11</a>	References .....	<a href="#">27</a>
<a href="#">12</a>	Author's Addresses .....	<a href="#">28</a>
<a href="#">13</a>	Full Copyright Statement .....	<a href="#">29</a>



## Table of Tables

Herriot, et al.

Expires: January 13, 2001

[page 5]

Table 1 - Information about the Delivery Method.....	<a href="#">9</a>
Table 2 - Printer Name in Event Notification Content.....	<a href="#">18</a>
Table 3 - Event Name in Event Notification Content.....	<a href="#">19</a>
Table 4 - Job Name in Event Notification Content.....	<a href="#">19</a>
Table 5 - Job State in Event Notification Content.....	<a href="#">20</a>
Table 6 - Printer State in Event Notification Content.....	<a href="#">21</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 'mailto' delivery method.

For this Delivery Method, when an Event occurs, the Printer immediately sends an Event Notification via an email message to the Notification Recipient specified in the Subscription Object. The message body of the email consists of Human Consumable text and is not intended to be parsed by a machine. The 'mailto' Delivery Method is a 'push' Delivery Method as defined in [[ipp-ntfy](#)].

The Notification Recipient receives the Event Notification in the same way as it receives any other email message.

## **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](#)].

For 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 contains the scheme "mailto", the client is requesting that the Printer use the 'mailto' Delivery Method for Event

Notifications generated from the new Subscription Object.

Herriot, et al.

Expires: January 13, 2001

[page 7]

For this Delivery Method, the "notify-recipient-uri" attribute value MUST consist of a "mailto" scheme followed by a colon, and then followed by an address part (e.g. 'mailto:smith@abc.com'). See [section 5.2.1](#) for the syntax of the "notify-recipient-uri" attribute value for this Delivery Method.

A Printer MUST support SMTP [[RFC821](#)], and it MAY support other email protocols. A Printer MAY use additional services, such as SMTP delivery status notification [[RFC1891](#)] or S/MIME encryption [[RFC2633](#)].

If the client wants the Printer to send Event Notifications via the 'mailto' Delivery Method, the client MUST choose a value for "notify-recipient-uri" attribute which conforms to the rules of [section 5.2.1](#). To avoid denial-of-service attacks, a client SHOULD NOT use distribution lists as the Notification Recipient.

When an Event occurs, the Printer MUST immediately:

1. Find all pertinent Subscription Objects P according to the rules of section 9 of [[ipp-ntfy](#)], AND
2. Find the subset M of these Subscription Objects P whose "notify-recipient-uri" attribute has a scheme value of 'mailto', AND
3. For each Subscription Object in M, the Printer MUST
  - a) generate an email message as specified in [section 5.2.2](#) AND
  - b) send the email message to the Notification Recipient specified by the address part of the "notify-recipient-uri" attribute value (see [section 5.2.1](#)).

If the Printer supports only SMTP, it MUST send the email message via SMTP. If the Printer supports additional email protocols, it MUST determine the protocol from the address part of the "notify-recipient-uri" attribute value and then send the email message via the appropriate email protocol.

When a Subscription Object is listening to a frequently occurring Event, such as 'job-progress', the Printer MUST moderate the sending of Event



Notifications caused by such an Event. It is implementation dependent as to how a Printer moderates Events and how a human controls the moderation.

#### **4 General Information**

If a Printer supports this Delivery Method, the following are its characteristics.

Table 1 - Information about the Delivery Method

Document Method Conformance Requirement	Delivery Method Realization
1. What is the URL scheme name for the Delivery Method?	mailto
2. Is the Delivery Method REQUIRED, RECOMMEND, or OPTIONAL for an IPP Printer to support?	RECOMMENDED
3. What transport and delivery protocols does the Printer use to deliver the Event Notification Content, i.e., what is the entire network stack?	support other email protocols.  A Printer MUST support SMTP. It MAY
4. Can several Event Notifications be combined into a Compound Event Notification?	A Printer implementation MAY combine several Event Notifications into a single email message.
5. Is the Delivery Method initiated by the Notification Recipient (pull), or by the Printer (push)?	This Delivery Method is a push.
6. Is the Event Notification content Machine Consumable or Human Consumable?	Human Consumable
7. What section in this document answers the following question? For a Machine	<a href="#">Section 6</a>



Consumable Event Notification,

Herriot, et al.

Expires: January 13, 2001

[page 9]

what is the representation and encoding of values defined in section 9.1 of [[ipp-ntfy](#)] and the conformance requirements thereof? For a Human Consumable Event Notification, what is the representation and encoding of pieces of information defined in [section 9.2](#) of [[ipp-ntfy](#)] and the conformance requirements thereof?

8. What are the latency and reliability of the transport and delivery protocol? Same as the underlying SMTP (or other optional email transport)
9. What are the security aspects of the transport and delivery protocol, e.g., how it is handled in firewalls? Same as the underlying SMTP (or other optional email transport)
10. What are the content length restrictions? None
11. What are the additional values or pieces of information that a Printer sends in an Event Notification content and the conformance requirements thereof? None
12. What are the additional Subscription Template and/or Subscription Description attributes and the conformance requirements thereof? See [section 5.1.1](#) on "notify-mailto-text-only"
13. What are the additional Printer Description attributes and the conformance requirements thereof? None



## **5 Subscription Template Attributes**

### **5.1 Additional Subscription Template Attributes**

This Delivery Method introduces one additional Subscription Template Attribute.

#### **5.1.1 notify-mailto-text-only (boolean)**

When the Printer generates an Event Notification from a Subscription Object, this attribute specifies whether the Printer generates the Event Notification with only plain text (i.e. 'text/plain') or with Content-Types that the Printer chooses.

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

A client **MAY** supply this attribute. If a client does not supply this attribute, the Printer **MUST** populate this attribute with the value of 'false' on the Subscription Object. There is no "notify-mailto-text-only-default" attribute.

If the value of this attribute is 'true' in a Subscription Object, the message body of each Event Notification that the Printer generates from the Subscription Object **MUST** contain plain text only (i.e. 'text/plain' with the charset specified by the "notify-charset" Subscription Object attribute).

If the value of this attribute is 'false' in a Subscription Object, the message body of each Event Notification that the Printer generates from the Subscription Object **MUST** contain a 'multipart/alternative'. One message body of the 'multipart/alternative' **MUST** be the same as the 'text/plain' message body when this attribute has the value of 'true'. Each of the other message bodies of the 'multipart/alternative' **MAY** be any Content-Type (e.g. 'text/html', 'image/gif', 'audio/basic', etc.).

A Printer **MUST** support both values ('true' and 'false') of this attribute. There is no "notify-mailto-text-only-supported" attribute.



## **5.2 Additional Information about Subscription Template Attributes**

This section describes additional values for attributes defined in [ipp-notify].

### **5.2.1 notify-recipient-uri (uri)**

This section describes the syntax of the value of this attribute for the 'mailto' Delivery Method. The syntax for values of this attribute for other Delivery Method is defined in other Delivery Method Documents.

In order to support the 'mailto' Delivery Method, the Printer **MUST** support the following syntax for the 'mailto' Delivery Method when the Printer uses SMTP. The line below use [RFC 822](#) syntax rules and terms.

```
"mailto:" mailbox
```

Note: the above syntax allows 1 occurrence of 'mailbox'. The occurrence of 'mailbox' represents an email address of a Notification Recipient. This syntax

For SMTP, the phrase 'address part' of the "notify-recipient-uri" attribute value refers to the 'mailbox' part of the value.

The Printer **MAY** support other syntax for the 'address part' if it supports email protocols in addition to SMTP.

### **5.2.2 notify-user-data (octetString(63))**

This attributes has a special use for the 'mailto' Delivery Method. It specifies the email address of the Subscribing Client. It is primarily useful when the Notification Recipient is some person other than the Subscribing Client. Then the Notification Recipient has a way to reply to the Subscribing Client.

If a client specifies this Delivery Method in a Subscription Creation Operation, and the specified Notification Recipient is not associated with the same person as the client, the client **SHOULD** supply its email address as the value of the "notify-user-data" attribute. If the client does not supply this attribute, the Printer **MUST NOT** populate the Subscription Object with this attribute.



## **6 Event Notification Content**

This section describes the content of an Event Notification sent via the 'mailto' Delivery Method using the SMTP protocol. This document does not describe the content for other email protocols, but an implementation should use this section as a model.

When a Printer sends an email message via SMTP, the content MUST conform to [RFC 822](#). The following sections define the content that a Printer MUST send. A Printer MAY send additional content as long as the resulting content conforms to [RFC 822](#).

Each subsection below specifies the syntax that pertains to the subsection. The syntax rules and syntactic terms (e.g. 'date-time') in each subsection come from [RFC 822](#), except for the section on "Content-Type" which comes from [RFC 1521](#).

The Event Notification content has two parts, the headers and the message body. The headers precede the message body and are separated by a blank line (see [[RFC 822](#)]).

### **[6.1](#) Headers**

When a Printer sends an Event Notification via SMTP, it MUST include the following headers. [RFC 822](#) RECOMMENDS that the headers be in the order that they appear below.

#### **[6.1.1](#) 'Date' header**

Syntax: "Date" ":" date-time

This header contains the date and time that the Event occurred.

The Printer MUST include a "Date" header if and only if it supports the "printer-current-time" Printer attribute.

#### **[6.1.2](#) 'From' header**

Syntax: "From" ":" mailbox

where

mailbox = addr-spec / phrase route-addr





This header causes a typical email reader to show the email as coming from the Printer that is sending the Event Notification.

The Printer MUST include a "From" header whose syntax is specified above.

The Printer MUST use the second alternative of the syntax for 'mailbox' defined above (i.e. 'phrase route-addr'). The 'phrase' is the Printer's display name and it MUST be the value of the "printer-name" Printer attribute. The 'route-addr' MUST contain an email address (inside angle brackets) belonging to either an administrator or the output-device. This email address NEED NOT be capable of receiving mail. There is no Printer attribute to hold this email address, so that it cannot be configured using the IPP protocol without an implementation-defined attribute extension.

#### [6.1.3](#) 'Subject' header

Syntax: "Subject" ":" \*text

This header specifies the subject of the message and contains a short summary of the Event Notification.

The Printer MUST include a "Subject" header whose syntax is specified above.

The Printer MUST localize the '\*text' using the values of the "notify-charset" and "notify-natural-language" Subscription Object attributes.

For Printer Events, the '\*text' SHOULD start with the localized word "printer:", followed by the Printer name, and then followed by the localized Event name, e.g., in English: "printer: 'tiger' stopped" or in Danish: 'Printeren 'tiger' er standset'.

For Job Events, the '\*text' SHOULD start with the localized phrase "print job:", followed by the Job name, and then followed by the localized Event name, e.g., in English: "print job: 'financials' completed".

The wording is implementation dependent. A Notification Recipient MUST NOT expect to be able to parse this text. But an email filter might look for "printer" or "print job".

#### [6.1.4](#) 'Sender' header

Syntax: "Sender" ":" mailbox



This header causes a typical email reader to show the email as coming on behalf of the person associated with the Subscribing Client.

If the Subscription Object contains the "notify-user-data" attribute, and if its value satisfies the [RFC 822](#) syntax rules for 'mailbox', the Printer MUST include a "Sender" header whose syntax is specified above. Otherwise, the Printer MUST NOT include a "Sender" header.

For the "Sender" header, the 'mailbox' MUST be the value of the "notify-user-data" Subscription Object attribute. See [section 5.2.2](#) for details about the "notify-user-data" attribute.

#### [6.1.5](#) 'Reply-to' header

Syntax: "Reply-to" ":" mailbox

If the Notification Recipient replies to Event Notification email, this header causes a typical email reader to send email to the person acting as the Subscribing Client. The rules are identical to the "Sender" header.

If the Subscription Object contains the "notify-user-data" attribute, and if its value satisfies the [RFC 822](#) syntax rules for "mailbox", the Printer MUST include a "Reply-to" header whose syntax is specified above. Otherwise, the Printer MUST NOT include a "Reply-to" header.

For the "Reply-to" header, the "mailbox" MUST be the value of the "notify-user-data" Subscription Object attribute. See [section 5.2.2](#) for details about the "notify-user-data" attribute.

#### [6.1.6](#) 'To' header

Syntax: "To" ":" 1#mailbox

See [[RFC 1521](#)] for the syntax.

This header specifies the Notification Recipient(s).

The Printer MUST include a "To" header whose syntax is specified above.

The '1#mailbox' MUST be the '1#mailbox' part of the value of the "notify-recipient-uri" Subscription attribute, i.e. the part after the "mailto:".



### [6.1.7](#) 'Content-type' header

Syntax: "Content-Type" ":" type "/" subtype \*(";"parameter)

See [[RFC 1521](#)] for the syntactic terms (e.g. 'type').

This header specifies the format of the message body.

The Printer MUST include the "Content-Type" header.

If the value of the "notify-mailto-text-only" Subscription Object attribute is 'true', the 'type' MUST be "plain", the 'subtype' MUST be "text" and the 'parameter' MUST be ' "charset=" XXX' where XXX is the value of the "notify-charset" Subscription Object attribute, e.g. 'text/plain;charset=UTF-8'.

If the value of the "notify-mailto-text-only" Subscription Object attribute is 'false', the 'type' MUST be "multipart", the 'subtype' MUST be "alternative" and the 'parameter' MUST include the boundary string. Each header of a body part of a multipart entity also has a Content-Type and its value of 'type', 'subtype' and 'parameter' MUST be values allowed by [RFC 1521](#) or some registered MIME type. That is, a Printer MAY send any format it wishes in each body part of a multipart entity, e.g. 'text/html', 'image/gif', or 'audio/basic'.

## [6.2](#) Message Body

This document describes a message body that is plain text. The content of all other Content-Types is implementation dependent. A Printer MUST include a plain text message even when it sends other Content-Types in a 'multipart/alternative'.

When a Printer sends a plain text message, it MUST localize the text using the values of the "notify-charset" and "notify-natural-language" Subscription Object attributes.

Section 9.2 in [[ipp-ntfy](#)] specifies the information that a Delivery Method MUST specify and a Printer SHOULD send. This section contains the information from section 9.2 in [[ipp-ntfy](#)] and changes "Printer SHOULD send" to "Printer MUST send".

A Printer MUST send the following localized information in the message body. The specific wording of this information and its layout are

implementation dependent.

Herriot, et al.

Expires: January 13, 2001

[page 16]

- a) the Printer name (see Table 2)
- b) omitted (see below).
- c) for Printer Events only:
  - i) the Event (see Table 3) and/or Printer state information (see Table 6)
- d) for Job Events only:
  - i) the job identity (see Table 4)
  - ii) the Event (see Table 3) and/or Job state information (see Table 5)

Item b) in the above list is omitted because the Printer sends the time of the Event as an email header (see [section 6.1.1](#) on the 'Date' header).

The subsections of this section specify the attributes that a Printer MUST use to obtain this information.

The Printer MAY send additional information, depending on implementation.

Notification Recipients MUST NOT expect to be able to parse the message.

The next three sections define the attributes in Event Notification Contents that are:

- a) for all Events
- b) for Job Events only
- c) for Printer Events only

#### **[6.2.1](#) Event Notification Content Common to All Events**

The Printer MUST send the following information.

There is a separate table for each piece of information. Each row in the table represents a source value for the information and the values are listed in order of preference, with the first one being the preferred one. An implementation SHOULD use the source value from the earliest row in each table. It MAY use the source value from another row instead, or





it MAY combine the source values from several rows. An implementation is free to determine the best way to present this information.

The tables in this section and following contain the following columns for each piece of information:

- a) Source of Value: the name of the attribute that supplies the value for the Event Notification
- b) Sends: if the Printer supports the value (column 1) on the Source Object (column 3) the Delivery Method MUST specify
  - MUST: that the Printer MUST send the value.
  - SHOULD: either that the Printer MUST send the value or that the value is incompatible with the Delivery Method.
  - MAY: that the Printer MUST, SHOULD, MAY, MUST NOT, SHOULD NOT, or NEED NOT send the value. The Delivery Method specifies the level of conformance for the Printer.
- c) Source Object: the object from which the source value comes.

In all tables of this section, all rows contain a "MAY" in order to state that the Delivery Method specifies the conformance.

Table 2 lists the source of the information for the Printer Name. The "printer-name" is more user-friendly unless the Notification Recipient is in a place where the Printer name is not meaningful. For example, an implementation could have the intelligence to send the value of the "printer-name" attribute to a Notification Recipient that can access the Printer via value of the "printer-name" attribute and otherwise send the value of the "notify-printer-uri" attribute.

Table 2 - Printer Name in Event Notification Content

Source Value	Sends	Source Object
printer-name (name(127))	MAY	Printer



Source Value	Sends	Source Object
notify-printer-uri (uri)	MAY	Subscription

Table 3 lists the source of the information for the Event name. A Printer MAY combine this information with state information described for Jobs in Table 5 or for Printers in Table 6.

Table 3 - Event Name in Event Notification Content

Source Value	Sends	Source Object
notify-subscribed-event (type2 keyword)	MAY	Subscription

#### [6.2.2](#) Additional Event Notification Content for Job Events

This section lists the source of the additional information that a Printer MUST send for Job Events.

Table 4 lists the source of the information for the job name. The "job-name" is likely more meaningful to a user than "job-id".

Table 4 - Job Name in Event Notification Content

Source Value	Sends	Source Object
job-name (name(MAX))	MAY	Job



Source Value	Sends	Source Object
job-id (integer(1:MAX))	MAY	Job

Table 5 lists the source of the information for the job-state. If a Printer supports the "job-state-message" and "job-detailed-state-message" attributes, it SHOULD use those attributes for the job state information, otherwise, it should fabricate such information from the "job-state" and "job-state-reasons". For some Events, a Printer MAY combine this information with Event information.

Table 5 - Job State in Event Notification Content

Source Value	Sends	Source Object
job-state-message (text(MAX))	MAY	Job
job-detailed-status-messages (1setOf text(MAX))	MAY	Job
job-state (type1 enum)	MAY	Job
job-state-reasons (1setOf type2 keyword)	MAY	Job

### **6.2.3 Additional Event Notification Content for Printer Events**

This section lists the source of the additional information that a Printer MUST send for Printer Events.

Table 6 lists the source of the information for the printer-state. If a Printer supports the "printer-state-message", it SHOULD use that attribute for the job state information, otherwise it SHOULD fabricate such information from the "printer-state" and "printer-state-reasons". For some Events, a Printer MAY combine this information with Event information.



Table 6 - Printer State in Event Notification Content

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

### [6.3](#) Examples

This section contains three examples. One is a Job Event and the other two are Printer Events, the latter in Danish.

A Printer implementation NEED NOT generate Event Notification content that is identical or even similar to these examples. In fact it would be unfortunate if every implementation copied these example as is. These examples merely show some possibilities and are not necessarily the best way to convey information about an Event.

#### [6.3.1](#) Job Event Example

This section contains an example of an Event Notification of a Job Event.

A Subscribing Client Mike Jones (who works for xyz Corp.) performs a Subscription Creation Operation as part of the Print-Job operation on Printer "ipp://tiger@abc.com". Mike Jones specifies that the "job-name" is "financials". Mike is printing the Job for Bill Smith at abc Corp. The Subscription Object then has the following attributes:





Attribute Name	Attribute Value
notify-recipient-uri	mailto:bsmith@abc.com
notify-events	job-completed
notify-user-data	mjones@xyz.com
notify-mailto-text-only	true
notify-charset	us-ascii
notify-natural-language	en-us
notify-subscription-id	35692
notify-sequence-number	0
notify-printer-up-time	34593
notify-printer-uri	ipp://tiger@abc.com
notify-job-id	345
notify-subscriber-user-name	mjones

When the Job completes, the Printer generates and sends the following email message:

Date: 17 Jul 00 1632 PDT  
From: tiger <printAdmin@abc.com>  
Subject: print job: 'financials' completed  
Sender: mjones@xyz.com  
Reply-to: mjones@xyz.com  
To: bsmith@abc.com  
Content-type: text/plain

printer: tiger  
job: financials  
job-state: completed

The reader should note that the phrases are not identical to IPP keywords. They have been localized to English.



### **6.3.2 Printer Event Example**

This section contains an example of an Event Notification of a Printer Event.

A Subscribing Client Peter Williams, a Printer admin, performs a Create-Printer-Subscriptions operation on Printer "ipp://tiger@abc.com". The Subscription Object then has the following attributes:

Attribute Name	Attribute Value
notify-recipient-uri	mailto:pwilliams@abc.com
notify-events	printer-state-changed
notify-mailto-text-only	true
notify-charset	us-ascii
notify-natural-language	en-us
notify-subscription-id	4623
notify-sequence-number	0
notify-printer-uptime	23002
notify-printer-uri	ipp://tiger@abc.com
notify-lease-expiration-time	0
notify-subscriber-user-name	pwilliams

When the Printer jams, the Printer generates and sends the following email message:

```
Date: 29 Aug 00 0832 PDT
From: tiger <printAdmin@abc.com>
Subject: printer: 'tiger' has stopped
To: pwilliams@abc.com
Content-type: text/plain
```



Printer tiger has stopped with a paper jam.

The reader should note that the phrases are not identical to IPP keywords. They have been localized to English.

### **6.3.3 Printer Event Example (localized to Danish)**

This section contains an example of an Event Notification of a Printer Event localized to Danish.

A Subscribing Client Per Jensen, a Printer admin, performs a a Create-Printer-Subscriptions operation on Printer "ipp://tiger@def.dk". The Subscription Object then has the following attributes:

Attribute Name	Attribute Value
notify-recipient-uri	mailto:pjensen@def.dk
notify-events	printer-state-changed
notify-mailto-text-only	true
notify-charset	utf-8
notify-natural-language	da
notify-subscription-id	50225
notify-sequence-number	0
notify-printer-uptime	53217
notify-printer-uri	ipp://tiger@def.dk
notify-lease-expiration-time	0
notify-subscriber-user-name	pjensen

When the Printer jams, the Printer generates and sends the following email message:



Date: 29 Jan 00 0832 CET  
From: tiger <admin@def.dk>  
Subject: Printeren 'tiger' er standset  
To: pjensen@def.dk  
Content-type: text/plain; charset=utf-8

Printerens navn er 'tiger'.  
Printeren er standset.  
Aarsagen er papir stop.

## **7 Conformance Requirements**

The 'mailto' Delivery Method is RECOMMENDED for a Printer to support.  
If the Printer supports the 'mailto' Delivery Method, the Printer MUST:

- 1. meet the conformance requirements defined in [[ipp-ntfy](#)].**
- 2. support the "notify-mailto-text-only " Subscription Object attribute defined in [section 5.1.1](#).**
- 3. support the syntax for the "notify-recipient-uri" Subscription Object attribute defined in [section 5.2.1](#)**
- 4. support the use for the "notify-user-data" Subscription Object attribute defined in [section 5.2.2](#)**
- 5. support SMTP for sending Event Notifications.**
- 6. support the 'text/plain' Content-Type for the message body.**
- 7. support sending Event Notification via email with the content specified in [section 5.2](#).**

## **8 IANA Considerations**

Because the 'mailto' URL scheme is already defined in a standards track document [[RFC 2368](#)] and registered with IANA, this document does not require anything further of IANA.





## **9 Internationalization Considerations**

This Delivery Method presents no internationalization considerations beyond those covered in the [[ipp-ntfy](#)] document, and sections [6.1.3](#) and [6.2](#) of this document.

The Notification Recipient is expected to present the email as received because the Printer does all necessary localization to the Event Notification contents.

## **10 Security Considerations**

The biggest security concern is that a Subscribing Client will cause unsolicited Event Notifications to be sent to third parties, potentially creating denial-of-service problems (i.e., spam). The problem is even worse if the third parties are distribution lists.

There exist scenarios where third party notification is required (see Scenario #2 and #3 in [[ipp-not-req](#)]). The fully secure solution would require active agreement of all persons before they can become Notification Recipients. However, requirement #9 in [[ipp-req](#)] ("There is no requirement for IPP Printer receiving the print request to validate the identity of an event recipient") argues against this. To minimize the risk, a Printer could disallow third party Notification Recipients (a traditional facsimile model).

The Delivery Method recommends that the Subscribing Client supply his or her email address as the value of the "notify-user-data" attribute in the Subscription Creation Operation when the Notification Recipient is a third party. To reduce the chance of spamming or identify the spammer, a Printer could disallow third party Notification Recipients if the Subscribing Client doesn't supply the "notify-user-data" attribute with a valid email address.

Some firewall administrators prevent mail attachments from being accepted into their organizations because of the problem of the attachments containing computer viruses. The 'mailto' Delivery Method allows the Subscribing Client to request that the Content-Type of a message body be 'text/plain'.



## **11 References**

### **[ipp-iig]**

Hastings, T., Manros, C., Kugler, K, Holst H., Zehler, P.,  
"Internet Printing Protocol/1.1: [draft-ietf-ipp-implementers-guide-v11-01.txt](#), work in progress, May 9, 2000

### **[ipp-mod]**

R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell,  
"Internet Printing Protocol/1.0: Model and Semantics", <[draft-ietf-ipp-model-v11-06.txt](#)>, March 1, 2000.

### **[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-04.txt](#)>, July  
13, 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.

### **[RFC821]**

Jonathan B. Postel, "Simple Mail Transfer Protocol", [RFC 821](#),  
August, 1982.

### **[RFC822]**

David H. Crocker, "Standard For The Format Of ARPA Internet Text  
Messages", [RFC 822](#), August 13, 1982.

### **[RFC1341]**

N. Borenstein, N. Freed, "MIME (Multipurpose Internet Mail  
Extensions): Mechanisms for Specifying and Describing the Format of  
Internet Message Bodies", [RFC 1341](#), June, 1992.

### **[RFC1521]**

N. Borenstein, N. Freed, "MIME (Multipurpose Internet Mail  
Extensions) Part One: Mechanisms for Specifying and Describing the  
Format of Internet Message Bodies", [RFC 1521](#), September 1993.

## [RFC1891]

K. Moore, "SMTP Service Extension for Delivery Status Notifications", [RFC 1891](#), January 1996

## [RFC2026]

S. Bradner, "The Internet Standards Process -- Revision 3", [RFC 2026](#), October 1996.

## [RFC2046]

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.

## [RFC2368]

P. Hoffman, L. Masinter, J. Zawinski, "The mailto URL scheme", [RFC 2616](#), July 1998.

## [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.

## [RFC2633]

B. Ramsdell, "S/MIME Version 3 Message Specification", [RFC 2633](#), June 1999.

## **[12](#) Author's Addresses**

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

Phone: 650-813-7696  
Fax: 650-813-6860  
Email: [robert.herriot@pahv.xerox.com](mailto:robert.herriot@pahv.xerox.com)

Henrik Holst  
i-data international a/s  
Vadstrupvej 35-43  
2880 Bagsvaerd, Denmark

Phone: +45 4436-6000  
Fax: +45 4436-6111  
e-mail: [hh@i-data.com](mailto:hh@i-data.com)

Tom Hastings  
Xerox Corporation

737 Hawaii St. ESAE 231

Herriot, et al.

Expires: January 13, 2001

[page 28]

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

737 Hawaii St. ESAE 231

El Segundo, CA 90245

Phone: 310-333-8273

Fax: 310-333-5514

e-mail: [manros@cp10.es.xerox.com](mailto:manros@cp10.es.xerox.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 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.

