INTERNET-DRAFT
<draft-ietf-ipp-indp-method-043.txt>
[Target Category: standards track]

Hugo Parra
Novell, Inc.
Tom Hastings
Xerox Corp.
August 29, 2000February 28, 2001

Internet Printing Protocol (IPP): The 'indp' Delivery Method for Event Notifications and Protocol/1.0

Copyright (C) The Internet Society (20010). 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 IPP 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 ‘indp’ Delivery Method and Protocol. This Delivery Method is a simple protocol consisting of a single operation: the Send-Notifications operation which uses the same encoding and transport as IPP. This document defines version ‘1.0’ of the protocol.

For this Delivery Method, when an Event occurs, the Printer immediately sends (pushes) an Event Notification via the Send-Notifications operation to the Notification Recipient specified in the Subscription Object. The Event Notification content consists of Machine Consumable attributes and a Human Consumable "notify-text" attribute. The Notification Recipient returns a response to the Printer.

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~~RFC2911]
- Internet Printing Protocol/1.1: Encoding and Transport [~~ipp-pro~~RFC2910]
- Internet Printing Protocol/1.1: Implementer's Guide [ipp-iig]
- Mapping between LPD and IPP Protocols [RFC2569]
- Internet Printing Protocol (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 a message body over HTTP 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 "Internet Printing Protocol (IPP): IPP Event Notification Specification" document defines the semantics for Subscription Creation Operations and the requirements for other Delivery Method documents to define a Delivery Method to carry an Event Notifications to a Notification Recipient.

Table of Contents

Tables

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 'indp' Delivery Method. This Delivery Method is a simple protocol consisting of a single operation: the Send-Notifications operation which uses the same encoding and transport as IPP. This document defines version '1.0' of the protocol.

For the 'indp' Delivery Method, an IPP Printer sends (pushes) a Send-Notifications operation request containing one or more Event Notifications to the Notification Recipient specified in the Subscription Object. The Event Notification content consists of Machine Consumable attributes and a Human Consumable "notify-text" attribute.

The Notification Recipient receives the Event Notification as a Send-Notifications operation, in the same way as an IPP Printer receives IPP operations. The Notification Recipient returns a response to the Printer.

2 Terminology

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

Terms such as attributes, keywords, and support. These terms have special meaning and are defined in the model terminology [~~ipp-mod~~RFC2911] section 12.2.

Capitalized terms, such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED NOT, and OPTIONAL, have special meaning relating to conformance as specified in RFC 2119 [RFC2119] and [~~ipp-mod~~RFC2911] section 12.1. These terms refer to conformance to this document, if this document is implemented.

Capitalized terms, such as Notification Recipient, Event Notification, Printer, etc., that are defined in [ipp-ntfy] with the same meanings and are not reproduced here.

Event Notification Attributes Group – The attributes group in a request that contains Event Notification Attributes in a request or response.

3 Model and Operation

See [ipp-ntfy] for the description of the Event Notification Model and Operation. This Delivery Method takes advantage of combining several Event Notifications into a single Compound Event Notification that is delivery by a single Send-Notification operation to a single Notification Recipient.

When creating each Subscription object, the client supplies the "notify-recipient" (uri) Subscription Template 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 Notification Delivery Method defined in this document, the notification method is 'indp' and the rest of the URI is the address of the Notification Recipient to which the IPP Printer will send the Send-Notifications operation.

The 'indp' Notification Delivery Method defined in this document uses a client/server protocol paradigm. The "client" in this relationship is the Printer described in [ipp-ntfy] while the "server" is the Notification Recipient. The Printer invokes the Send-Notifications operation to communicate IPP Event Notification contents to the Notification Recipient. The Notification Recipient only conveys information to the Printer in the form of responses to the operations initiated by the Printer.

Printers that implement the 'indp' Notification Delivery Method will need to include an HTTP client stack while Notification Recipients that implement this Delivery Method will need to support an HTTP server stack. See section 10.2 for more details.

If the client wants the Printer to send Event Notifications via the ‘indp’ Delivery Method, the client MUST choose a value for “notify-recipient-uri” attribute which conforms to the rules of section 5.2.1.

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 ‘indp’, AND

3. For each Subscription Object in M, the Printer MUST

   a) generate a Send-Notifications request as specified in section 8.1.1 AND

   b) send the Send-Notifications request to the Notification Recipient specified by the address part of the “notify-recipient-uri” attribute value (see section 5.2.1).

If several events occur sufficiently close to one another for the same or different Subscription objects, but with the same Notification Recipient, the Printer MAY combine them into a single Send-Notifications request using a separate Event Notification Attributes group for each event (see section 8.1.1).

4 General Information

If a Printer supports this Delivery Method, Table 1 lists its characteristics.

194 **Table 1 - Information about the Delivery Method**

Document Method conformance requirement	'indp' realization
1. What is the URL scheme name for the Delivery Method?	indp
2. Is the Delivery Method is REQUIRED, RECOMMENDED, or OPTIONAL for an IPP Printer to support?	RECOMMENDED
3. What transport and delivery protocol does the Printer use to deliver the Event Notification content, i.e., what is the entire network stack?	A Printer MUST support a complete HTTP/1.1 stack [RFC2616]
4. Can several Event Notifications be combined into a Compound Event Notification?	A Printer implementation MAY combine several Event Notifications into a single Event Notifications request as separate Event Notification Attributes Groups, see section 8.1.1
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?	Machine Consumable with the "notify-text" attribute being Human Consumable
7. What section in this document answers the following question? For a Machine Consumable Event Notification, what is the representation and encoding of values defined in section 9.1 of [ipp-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?	The representation and encoding is the same as IPP. See section 8.1.1
8. What are the latency and reliability of the transport and delivery protocol?	Same as for IPP/1.0 or IPP/1.1 itself (see [~~ipp-mod~~RFC2911]).
9. What are the security aspects of the transport and delivery protocol, e.g., how it is handled in firewalls?	See section 15
10. What are the content length restrictions?	They are the same as for IPP/1.0 and IPP/1.1 itself (see [~~ipp-mod~~RFC2911]).
11. What are the additional values or pieces of information that a Printer sends in an Event Notification and the conformance requirements thereof?	A new Event Notifications attribute group (see section 10.1) and additional status codes for use in the response (see section 9)

Document Method conformance requirement	'indp' realization
12. What are the additional Subscription Template and/or Subscription Description attributes and the conformance requirements thereof?	None
13. What are the additional Printer Description attributes and the conformance requirements thereof?	None

The remaining sections of this document parallel the sections of [ipp-ntfy].

5 Subscription object attributes

This section defines the Subscription object conformance requirements for Printers.

5.1 Subscription Template Attribute Conformance

The 'indp' Delivery Method has the same conformance requirements for Subscription Template attributes as defined in [ipp-ntfy]. The 'indp' Delivery Method does not define any addition Subscription Template attributes.

5.2 Additional Information about Subscription Template Attributes

This section defines additional information about Subscription Template attributes defined in [ipp-ntfy].

5.2.1 notify-recipient-uri (uri)

This section describes the syntax of the value of this attribute for the ‘indp’ 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 ‘indp’ Delivery Method and Protocol, the Printer MUST support the following syntax:

> The 'indp://' URI scheme. The remainder of the URI indicates the host name ~~and~~ or host address (and optional path) of the Notification Recipient that is to receive the Send-Notification operation.

5.3 Subscription Description Attribute Conformance

The 'indp' Delivery Method has the same conformance requirements for Subscription Description attributes as defined in [ipp-ntfy]. The 'indp' Delivery Method does not define any addition Subscription Description attributes.

6 Printer Description Attributes

This section defines the Printer Description Attributes conformance requirements for Printers.

6.1 Printer Description Attribute Conformance

The 'indp' Delivery Method has the same conformance requirements for Printer Description attributes as defined in [ipp-ntfy]. The 'indp' Delivery Method does not define any addition Printer Description attributes.

6.2 New Values for Existing Printer Description Attributes

This section defines additional values for existing Printer Description attributes.

6.2.1 notify-schemes-supported (1setOf uriScheme)

The following “notify-schemes-supported” value is added in order to support the new Delivery Method defined in this document:

'indp': - The IPP Notification Delivery Method defined in this document.

6.2.2 operations-supported (1setOf type2 enum)

Table 2 lists the “operation-id” value added in order to support the new operation defined in this document. The operation-id is assigned in the same name space as other operations that a Printer supports. However, a Printer MUST NOT include this value in its "operations-supported" attribute unless it can accept the Send-Notifications request.

Table 2 – Operation-id assignments

Value	Operation Name
0x001D	Send-Notifications

7 Attributes Only in Event Notifications

No additional attributes are defined only for use in Event Notifications besides those defined in [ipp-ntfy].

8 Operations for Notification

This section defines the operation for Event Notification using the 'indp' Delivery Method.

There is only one operation defined: Send-Notifications. Section 6.2.2 assigns of the “operation-id” for the Send-Notifications operation and the following section defined the operation.

8.1 Send-Notifications operation

This REQUIRED operation allows a Printer to send one or more Event Notifications to a Notification Recipient using HTTP.

The Printer composes the information defined for an IPP Notification [ipp-ntfy] and sends it using the Sent-Notifications operation to the Notification Recipient supplied in the Subscription object.

The Send-Notifications operations uses the operations model defined by IPP [RFC2566]. This includes, the use of a URI as the identifier for the target of each operation, the inclusion of a version number, operation-id, and request-id in each request, and the definition of attribute groups. The Send-Notifications operation uses the Operation Attributes group, but currently has no need for the Unsupported Attributes, Printer Object Attributes, and Job-Object Attributes groups. However, it uses a new attribute group, the Event Notification Attributes group.

The Notification Recipient MUST accept the request in any state. There is no state defined for the Notification Recipient for this Delivery Method.

Access Rights: Notification Recipient MAY enforce access rights. If the Printer receives a rejection with these status codes: 'client-error-forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-authorized’ status code , the Printer SHOULD cancel the subscription.

8.1.1 Send-Notifications Request

Every operation request MUST contains the following parameters (see [~~ipp-mod~~RFC2911] section 3.1.1):

- a "version-number" ‘1.0’ – the version of the ‘indp’ protocol is ‘1.0’.
- an "operation-id" - the value defined in Table 2
- a "request-id" - the request id (see [~~ipp-mod~~RFC2911] section 3.1.2).

The following groups of attributes MUST be part of the Send-Notifications Request:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes as defined in [~~ipp-mod~~RFC2911] section 3.1.4.1.

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 request.

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.

Target:

A copy of the Subscription object's "notify~~ication-~~recipient-uri" (uri) attribute which is the target of this operation as described in [~~ipp-mod~~RFC2911] section 3.1.5, i.e., the URI of the 'indp' Notification Recipient (see section 5.2.1).

Group 2 to N: Event Notification Attributes

In each group 2 to N, each attribute is encoded using the IPP rules for encoding attributes [~~ipp-pro~~RFC2910] and may be encoded in any order. Note: the Get-Jobs response in [~~ipp-mod~~RFC2911] acts as a model for encoding multiple groups of attributes.

Each Event Notification Group MUST contain all of attributes specified in [ipp-ntfy] section 9.1 (“Content of Machine Consumable Event Notifications”) with exceptions denoted by asterisks in the tables below.

The tables below are copies of the tables in [ipp-ntfy] section 9.1 (“Content of Machine Consumable Event Notifications”) except that each cell in the “Sends” column is a “MUST”.

For an Event Notification for all Events, the Printer sends the following attributes.

Table 3 – 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 (MAX))	MUST	Event Notification
attributes from the “notify-attributes” attribute, if any ***	MUST ***	Printer
attributes from the “notify-attributes” attribute, if any ***	MUST ***	Job
attributes from the “notify-attributes” attribute, if any ***	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 and the Printer does not send any client-requested attributes.

For Event Notifications for Job Events, the Printer sends the following additional attributes shown in Table 4.

Table 4 – 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 5.

Table 5 – 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 sends the following additional attributes shown in Table 6.

Table 6 – 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

8.1.2 Send-Notifications Response

The Notification Recipient MUST return (to the client which is the Printer) the following sets of attributes as part of a Send-Notifications response:

Every operation response contains the following REQUIRED parameters (see [~~ipp-mod~~RFC2911] section 3.1.1}:

- a "version-number"
- a "status-code"
- the "request-id" that was supplied in the corresponding request

Group 1: Operation Attributes

Status Message:
As defined in [~~ipp-mod~~RFC2911].

The Notification Recipient can return any status codes defined in [~~ipp-mod~~RFC2911] and section 9.1 that applies to all of the Event Notification Attribute groups. The following is a description of the important status codes:

'successful-ok': the Notification Recipient received all of the Event Notification Attribute Groups and was expecting each of them.

'successful-ok-ignored-notifications': the Notification Recipient was able to consume some, but not all of the Event Notification Attributes Groups sent. The Event Notification Attributes Groups with a “notify-status-code” attribute are the ones that were ignored or are to be canceled.

'client-error-ignored-all-notifications': the Notification Recipient was unable to consume any of the Event Notification Attributes Groups sent. The Event Notification Attributes Groups with a “notify-status-code” attribute are the ones that were ignored or are to be canceled.

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes as defined in [ipp-modRFC2911] section 3.1.4.1.

Group 2 to N: Notification Attributes

These groups MUST be returned if and only if the "status-code" parameter returned in Group 1 is anything but the 'successful-ok' status code.

"notifyication-status-code" (type2 enum)

Indicates whether the Notification Recipient was able to consume the n-th Notification Report as follows:

'successful-ok' - this Event Notification Attribute Group was consumed

'client-error-not-found' - this Event Notification Attribute Group was not able to be consumed. The Printer MUST cancel the Subscription and MUST NOT attempt to send any further Event Notifications from the associated Subscription object.

'successful-ok-but-cancel-subscription' - the Event Notification Attribute Group was consumed, but the Notification Recipient wishes to cancel the Subscription object. The Printer MUST cancel the Subscription and MUST NOT attempt to send any further Event Notifications from the associated Subscription object.

9 Status Codes

This section lists status codes whose meaning have been extended and/or defined for returning in Event Notification Attribute Groups as the value of the "notifyication-status-code" operation attribute. The code values are allocated in the same space as the status codes in [ipp-modRFC2911].

9.1 Additional Status Codes

The following status codes are defined as extensions for Notification and are returned as the value of the “status-code” parameter in the Operation Attributes Group of a response (see [ipp-modRFC2911] section 3.1.6.1). Operations in this document can also return the status codes defined in section 13 of [ipp-modRFC2911]. The ‘successful-ok’ status code is an example of such a status code.

9.1.1 successful-ok-ignored-notifications (0x0004)

The Notification Recipient was able to consume some, but not all, of the Event Notifications Attributes Groups sent by the Printer in the Send-Notifications request. See section 8.1.2 for further details.

9.1.2 client-error-ignored-all-notifications (0x0416)

The Notification Recipient was unable to consume any of the Event Notification Attributes Groups sent by the Printer. The Event Notification Attributes Groups with a “notify-status-code” attribute are the ones that were ignored or are to be canceled.

9.2 Status Codes returned in Event Notification Attributes Groups

This section contains values of the “notify-status-code” attribute that the Notification Recipient returns in a Event Notification Attributes Group in a response when the corresponding Event Notification Attributes Group in the request:

1. was not consumed OR
2. was consumed, but the Notification Recipient wants to cancel the corresponding Subscription object

The following sections are ordered in decreasing order of importance of the status-codes.

9.2.1 client-error-not-found (0x0406)

This status code is defined in [~~ipp-mod~~RFC2911]. This document extends its meaning and allows it to be returned in an Event Notification Attributes Group of a response.

The Notification Recipient was unable to consume this Event Notification Attributes Group because it was not expected. See section 8.1.2 for further details.

9.2.2 successful-ok-but-cancel-subscription (0x0006)

The Notification Recipient was able to consume this Event Notification Attributes Group that the Printer sent, but wants the corresponding Subscription object to be canceled none-the-less. See section 8.1.2 for further details.

10 Encoding and Transport

This section defines the encoding and transport used by the 'indp' Delivery Method.

10.1 Encoding of the Operation Layer

The 'indp' Delivery Method uses the IPP operation layer encoding described in [~~ipp-pro~~RFC2910] and the ~~following~~ Event Notification Attributes Group tag allocated by [ipp-ntfy] as shown in Table 7:

Table 7 – The "event-notification-attributes-tag" value

Tag Value (Hex)	Meaning
0x07	“event-notification-attributes-tag”

10.2 Encoding of Transport Layer

The 'indp' Notification Delivery Method uses the IPP transport layer encoding described in [~~ipp-pro~~RFC2910].

It is REQUIRED that an 'indp' Notification Recipient implementation support HTTP over the IANA assigned Well Known Port assigned to the 'indp' Delivery Method as its default port by IANA (see section 13), though a Notification Recipient implementation MAY support HTTP over some other port as well.

11 Conformance Requirements

This section defines conformance requirements for Printers and Notification Recipients.

11.1 Conformance Requirements for Printers

The ‘indp’ Delivery Method is RECOMMENDED for a Printer to support.

IPP Printers that conform to this specification:~~If the Printer supports the ‘indp’ Delivery Method, the Printer MUST:~~

1. MUST meet the conformance requirements defined in [ipp-ntfy].
2. MUST support the conformance requirements for Subscription object attributes defined in section 5, including the syntax for the “notify-recipient-uri” Subscription Object attribute defined in section 5.2.1.
3. MUST support the conformance requirements for Printer Description object attributes defined in section 6.
4. MUST support the ‘indp’ protocol by sending Event Notifications using the Send-Notifications operation defined in section 8.1.

5. MUST send INDP URLs (e.g., in the "notify-recipient-uri" attribute in 'Send-Notifications') that conform to the ABNF specified in section 12.5 of this document;

6. MUST send INDP operations via the port specified in the INDP URL (if present) or otherwise via IANA assigned well-known port [TBD];

7. MUST convert INDP URLs to their corresponding HTTP URL forms by the same rules used to convert IPP URLs to their corresponding HTTP URL forms (see section 5 'IPP URL Scheme' in [RFC2910]).

~~8.support sending Event Notification via email with the content specified in section 8.1.1.~~

11.2 Conformance Requirements for INDP Notification Recipients

INDP Notification Recipients that conform to this specification:

1. ~~A Notification Recipient~~ MUST accept Send-Notifications requests and return Send-Notifications responses as defined in sections 8 and 9.

2. SHOULD reject received INDP URLs in "application/ipp" request bodies (e.g., in the "notify-recipient-uri" attribute in 'Send-Notifications') that do not conform to the ABNF for INDP URLs specified in section 12.5 of this document;

3. MUST listen for INDP operations on IANA-assigned well-known port [TBD], unless explicitly configured by system administrators or site policies;

4. SHOULD NOT listen for INDP operations on any other port, unless explicitly configured by system administrators or site policies.

12 INDP URL Scheme

12.1 INDP URL Scheme Applicability and Intended Usage

This section is intended for use in registering the "indp" URL scheme with IANA and fully conforms to the requirements in [RFC2717]. This document defines the "indp" URL (Uniform Resource Locator) scheme for specifying the location of an INDP Notification Recipient object which implements IPP Notification Delivery Protocol (INDP) specified in this document.

The intended usage of the "indp" URL scheme is COMMON.

12.2 INDP URL Scheme Associated INDP Port

All INDP URLs which do NOT explicitly specify a port MUST be used over IANA-assigned well-known port [TBD] for the INDP protocol.

See: IANA Port Numbers Registry [IANA-PORTREG].

12.3 INDP URL Scheme Associated MIME Type

All INDP protocol operations (requests and responses) MUST be conveyed in an "application/ipp" MIME media type as registered in [IANA-MIMEREG]. INDP URLs MUST refer to INDP Notification Recipient objects which support this "application/ipp" MIME media type.

See: IANA MIME Media Types Registry [IANA-MIMEREG].

12.4 INDP URL Scheme Character Encoding

The INDP URL scheme defined in this document is based on the ABNF for the HTTP URL scheme defined in HTTP/1.1 [RFC2616], which is derived from the URI Generic Syntax [RFC2396] and further updated by [RFC2732] and [RFC2373] (for IPv6 addresses in URLs). The INDP URL scheme is case-insensitive in the host name or host address part; however the path part is case-sensitive, as in [RFC2396]. Code points outside [US-ASCII] MUST be hex escaped by the mechanism specified in [RFC2396].

12.5 INDP URL Scheme Syntax in ABNF

This section is intended for use in registering the "indp" URL scheme with IANA and fully conforms to the requirements in [RFC2717]. This document defines the "indp" URL (Uniform Resource Locator) scheme for specifying the location of an INDP Notification Recipient object which implements IPP Notification Delivery Protocol (INDP) specified in this document.

The intended usage of the "indp" URL scheme is COMMON.

The IPP protocol places a limit of 1023 octets (NOT characters) on the length of a URI (see section 4.1.5 'uri' in [RFC2911]). An INDP Notification Recipient MUST return 'client-error-request-value-too-long' (see section 13.1.4.10 in [RFC2911]) when a URI received in a request is too long.

Note: INDP Notification Recipients ought to be cautious about depending on URI lengths above 255 bytes, because some older client or proxy implementations might not properly support these lengths.

INDP URLs MUST be represented in absolute form. Absolute URLs always begin with a scheme name followed by a colon. For definitive information on URL syntax and semantics, see "Uniform Resource Identifiers (URI): Generic Syntax and Semantics" [RFC2396]. This specification adopts the definitions of

"port", "host", "abs_path", and "query" from [RFC2396], as updated by [RFC2732] and [RFC2373] (for IPv6 addresses in URLs).

The INDP URL scheme syntax in ABNF is as follows:

```
  indp_URL = "indp:" "//" host [ ":" port ] [ abs_path [ "?" query
  ]]
```

If the port is empty or not given, IANA-assigned well-known port [TBD] is assumed. The semantics are that the identified resource (see section 5.1.2 of [RFC2616]) is located at the INDP Notification Recipient listening for HTTP connections on that port of that host, and the Request-URI for the identified resource is 'abs_path'.

Note: The use of IP addresses in URLs SHOULD be avoided whenever possible (see [RFC1900]).

If the 'abs_path' is not present in the URL, it MUST be given as "/" when used as a Request-URI for a resource (see section 5.1.2 of [RFC2616]). If a proxy receives a host name which is not a fully qualified domain name, it MAY add its domain to the host name it received. If a proxy receives a fully qualified domain name, the proxy MUST NOT change the host name.

12.5.1 INDP URL Examples

The following are examples of valid INDP URLs for Notification Recipient objects (using DNS host names):

```
indp://abc.com
indp://abc.com/listener
```

Note: The use of IP addresses in URLs SHOULD be avoided whenever possible (see [RFC1900]).

The following literal IPv4 addresses:

```
192.9.5.5                    ; IPv4 address in IPv4 style
186.7.8.9                    ; IPv4 address in IPv4 style
```

are represented in the following example INDP URLs:

```
indp://192.9.5.5/listener
indp://186.7.8.9/listeners/tom
```

The following literal IPv6 addresses (conformant to [RFC2373]):

```
::192.9.5.5                  ; IPv4 address in IPv6 style
::FFFF:129.144.52.38         ; IPv4 address in IPv6 style
2010:836B:4179::836B:4179    ; IPv6 address per RFC 2373
```

are represented in the following example INDP URLs:

```
indp://[::192.9.5.5]/listener
indp://[::FFFF:129.144.52.38]/listener
indp://[2010:836B:4179::836B:4179]/listeners/tom
```

12.5.2 INDP URL Comparisons

When comparing two INDP URLs to decide if they match or not, an INDP Client SHOULD use a case-sensitive octet-by-octet comparison of the entire URLs, with these exceptions:

- A port that is empty or not given is equivalent to the well-known port for that INDP URL (port [TBD]);
- Comparisons of host names MUST be case-insensitive;
- Comparisons of scheme names MUST be case-insensitive;
- An empty 'abs_path' is equivalent to an 'abs_path' of "/".

Characters other than those in the "reserved" and "unsafe" sets (see [RFC2396] and [RFC2732]) are equivalent to their ""%" HEX HEX" encoding.

For example, the following three URIs are equivalent:

```
indp://abc.com/~smith/listener
indp://ABC.com/%7Esmith/listener
indp://ABC.com:/%7esmith/listener
```

13 IANA Considerations

IANA is requested to register the indp URL scheme as defined in section 12.

IANA is requested to assign a default system port (less than 1024) for use with the indp URL as defined in section 12.

The rest of this section contains the exact information for IANA to add to the IPP Registries according to the procedures defined in RFC 2911 [RFC2911] section 6.

Note to RFC Editors: Replace RFC NNNN below with the RFC number for this document, so that it accurately reflects the content of the information for the IANA Registry.

~~The 'indp' URL scheme for the 'indp' Delivery Method and Protocol will be registered with IANA. IANA will assign a default port to use with the 'indp' Delivery Method and Protocol~~

13.1 Operation Registrations

The operations defined in this document will be published by IANA according to the procedures in RFC 2911 [RFC2911] section 6.4 with the following path:

ftp.isi.edu/iana/assignments/ipp/operations/

The registry entry will contain the following information:

Operations:	Ref.	Section:
Send-Notifications operation	RFC NNNN	8.1

13.2 Additional values of existing attributes

13.2.1 Additional values for the “notify-schemes-supported” Printer attribute

The “notify-schemes-supported” uriScheme attribute value defined in this document will be published by IANA according to the procedures in RFC 2911 [RFC2911] section 6.1 with the following path:

ftp.isi.edu/iana/assignments/ipp/attribute-values/notify-schemes-supported/

The registry entry will contain the following information:

	Ref.	Section:
indp	RFC NNNN	6.2.1

13.2.2 Additional values for the “operations-supported” Printer attribute

The “operations-supported” type2 enum attribute value defined in this document will be published by IANA according to the procedures in RFC 2911 [RFC2911] section 6.1 with the following path:

ftp.isi.edu/iana/assignments/ipp/attribute-values/operations-supported/

The registry entry will contain the following information:

	Value	Ref.	Section:
Send-Notifications	0x001D	RFC NNNN	6.2.1

13.3 Status code Registrations

The status codes defined in this document will be published by IANA according to the procedures in RFC 2911 [RFC2911] section 6.6 with the following path:

ftp.isi.edu/iana/assignments/ipp/status-codes/

The registry entry will contain the following information:

Status codes:	Ref.	Section:
successful-ok-ignored-notifications (0x0004)	RFC NNNN	9.1.1
client-error-ignored-all-notifications (0x0416)	RFC NNNN	9.1.2

14 Internationalization Considerations

When the client requests Human Consumable form by supplying the "notify-text-format" operation attribute (see [ipp-ntfy]), the IPP Printer (or any Notification Service that the IPP Printer might be configured to use) supplies and localizes the text value of the "human-readable-report" attribute in the Notification according to the charset and natural language requested in the notification subscription.

15 Security Considerations

The IPP Model and Semantics document [~~ipp-mod~~RFC2911] discusses high level security requirements (Client Authentication, Server Authentication and Operation Privacy). Client Authentication is the mechanism by which the client proves its identity to the server in a secure manner. Server Authentication is the mechanism by which the server proves its identity to the client in a secure manner. Operation Privacy is defined as a mechanism for protecting operations from eavesdropping.

The Notification Recipient can cancel unwanted Subscriptions created by other parties without having to be the owner of the subscription by returning the 'successful-ok-but-cancel-subscription' status code in the Send-Notifications response returned to the Printer.

15.1 Security Conformance

Printers (client) MAY support Digest Authentication [RFC2617]. If Digest Authentication is supported, then MD5 and MD5-sess MUST be supported, but the Message Integrity feature NEED NOT be supported.

Notification Recipient (server) MAY support Digest Authentication [RFC2617]. If Digest Authentication is supported, then MD5 and MD5-sess MUST be supported, but the Message Integrity feature NEED NOT be supported.

Notification Recipients MAY support TLS for client authentication, server authentication and operation privacy. If a Notification Recipient supports TLS, it MUST support the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite as mandated by RFC 2246 [RFC2246]. All other cipher suites are OPTIONAL. Notification recipients MAY support Basic Authentication (described in HTTP/1.1 [RFC2616]) for client authentication if the channel is secure. TLS with the above mandated cipher suite can provide such a secure channel.

16 References

[ipp-iig]
Hastings, T., Manros, C., Kugler, K, Holst H., Zehler, P., “Internet Printing Protocol/1.1: draft-ietf-ipp-implementers-guide-v11-021.txt, work in progress, January 25, 2001May 9, 2000

[ipp-mod]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.1: Model and Semantics", <draft-ietf-ipp-model-v11-07.txt>, May 22, 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-064.txt>, January 24, 2001August 30, 2000.

[ipp-pro]
Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and Transport", draft-ietf-ipp-protocol-v11-06.txt, May 30, 2000.

[IANA-MIMEREG]
IANA MIME Media Types Registry. ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/

[IANA-PORTREG]
IANA Port Numbers Registry. ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers

[RFC1900]
B. Carpenter, Y. Rekhter. Renumbering Needs Work, RFC 1900, February 1996.

[RFC2026]
S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

[RFC2373]
R. Hinden, S. Deering. IP Version 6 Addressing Architecture, RFC 2373, July 1998.

[RFC2396]
Berners-Lee, T. et al. Uniform Resource Identifiers (URI): Generic Syntax, RFC 2396, August 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.

[RFC2617]
J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC2717]
R. Petke and I. King, “Registration Procedures for URL Scheme Names”, RFC 2717, November 1999.

[RFC2732]
R. Hinden, B. Carpenter, L. Masinter. Format for Literal IPv6 Addresses in URL's, RFC 2732, December 1999.

[RFC2910]
Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and Transport", RFC 2910, September 2001.

[RFC2911]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.1: Model and Semantics", RFC 2911, September 2001.

17 Author's Addresses

Hugo Parra
Novell, Inc.
1800 South Novell Place

Provo, UT 84606

Phone: 801-861-3307
Fax: 801-861-2517
e-mail: hparra@novell.com

Tom Hastings
Xerox Corporation
737 Hawaii St. ESAE 231
El Segundo, CA 90245

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

18 Full Copyright Statement

Copyright (C) The Internet Society (2001~~0~~). 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.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.