INTERNET-DRAFT There are 6 ISSUES highlighted like this.
<draft-ietf-ipp-indp-00.txt>

Hugo Parra
Novell, Inc.
Tom Hastings
Xerox Corporation
February 29, 2000

Internet Printing Protocol/1.1: **IPP Notification Delivery Protocol**

Status of this Memo

Abstract

The IPP event notification specification [ipp-ntfy] is an OPTIONAL extension to IPP/1.0 and IPP/1.1. [ipp-ntfy] which enables IPP clients to request notification of printer and job events. The IPP notification extension gives IPP Printers the flexibility to choose how many Subscriptions objects (individual requests for notification), what delivery methods, and what natural languages to support, among others. In practice, it’s the working environment where an IPP Printer is deployed what ultimately dictates the notification requirements for that printer. Notification Delivery Services exist to help event producers, such as IPP Printers, meet the varying notification needs of disparate environments. Specifically, an IPP Notification Delivery Service may extend the notification capabilities of IPP Printers and help customize the type of notification required in a highly specialized environment. This documents defines the IPP Notification Delivery Protocol (INDP), a protocol for IPP Printers to communicate with Notification Delivery Services using “application/ipp” as the encoding mechanism and HTTP as the transport. The definition of INDP lends itself nicely for use by IPP Printers and Notification Delivery Services for dispatching IPP Notifications to Notification Recipients as well.

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 (this document)
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]

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: 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.

Table of Contents

1 Introduction

IPP Printers that support the OPTIONAL IPP event notification extension [ipp-ntfy] either a) accept, store, and use notification Subscriptions to generate notification reports and implement one or more delivery methods for notifying interested parties, or b) support a subset of these tasks and farm out the remaining tasks to a Notification Delivery Service. The IPP Notification Delivery Protocol (INDP) specified in this document is a request/response protocol that may be used in a variety of notification scenarios. Its primary intended use is for IPP Printers to engage the assistance of Notification Delivery Services for storing notification Subscriptions, generating human-readable notifications in various languages, and implementing additional delivery methods. Moreover, IPP Printers and Notification Delivery Services may use INDP to send (push) event notifications to Notification Recipients.

2 Terminology

This document uses terms such as "attributes", "keywords", and "support". These terms have special meaning and are defined in the model terminology [ipp-mod] 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. These terms are defined in [ipp-mod] section 12.1 on conformance terminology, most of which is taken from RFC 2119 [RFC2119].

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

REQUIRED: if an implementation supports the extensions described in this document, it MUST support a REQUIRED feature.
OPTIONAL: if an implementation supports the extensions described in this document, it MAY support an OPTIONAL feature.
Event Notification (Notification for short) - See [ip-ntfy]
Notification Source - See [ipp-ntfy]
Notification Recipient - See [ipp-ntfy]
Subscription object - See [ipp-ntfy]
Ultimate Notification Recipient - See [ipp-ntfy]

3 Model and Operation

In the IPP Notification Model [ipp-ntfy], print clients request an IPP Printer for event notification by causing a Subscription object to be created at the printer. [ipp-ntfy] specifies a number of ways in which Subscription objects can be created. Each Subscription object lists the events of interest, the delivery method to be employed, and the address to which notifications should be dispatched, among others. When an event occurs, the printer is responsible for notifying each Notification Recipient that has registered interest in that event, using delivery method requested by that Notification Recipient. IPP Printers may employ the assistance of Notification Delivery Services to accomplish some or all of these tasks.

IPP Printers with support for Notification Delivery Services must support a new printer description attribute, “notification-delivery-services-uri-supported” (1SetOf uri). This attribute needs to be populated with the uri’s of the Notification Delivery Services the printer is configured to use. Whether IPP Printers dynamically discover Notification Delivery Services on the network or need to be configured by a system administrator it implementation dependant.

3.1 Notification Delivery Service Model

The INDP 1.0 model defines objects of type Server and Subscription. Each object definition includes a set of attributes that describe the state and workings of a Notification Delivery Service. An IPP Printer interact with instances of these object types by issuing INDP operations. This section describes the attributes that compose the Server and Subscription objects with their corresponding attribute syntaxes and values that are part of the Notification Delivery Service Model. Each attribute is uniquely identified in this document using a “keyword” as defined in the IPP/1.1: Model and Semantics document [ipp-mod]. INDP 1.0 defines The Notification Delivery Service

3.1.1 Server Object

The Server object represents the state and capabilities of a Notification Delivery Service. It implements the server-side of INDP. In version 1.0 of INDP, the Server object contains information about the capabilities of a Notification Delivery Service that are of interest to an IPP Printer.

The following attributes comprise the Server object. Their description and intended use follow.

- notify-natural-languages-supported
- notify-uri-schemes-supported

3.1.1.1 notify-natural-languages-supported (1setOf naturalLanguage)

MANDATORY {To be added}

3.1.1.2 notify-uri-schemes-supported (1setOf uri)

MANDATORY {To be added}

3.1.2 Subscription Object

The Subscription object represents a request for notification. Subscription Objects are contained by a Server object and are created as a result of an IPP Printer issuing a Create-Subscription operation. The syntax and semantics of a Subscription object exactly mirror those of the Subscription object defined in the IPP Notification spec [ipp-ntfy].

3.2 Notification Delivery Service Operation

The figure below illustrates four different configurations through which an IPP Printer may implement support for IPP notification. Each configuration is discussed in this section.

```
  O   +--------+            +-------------+
 /|\  | client | ---IPP---> | IPP Printer |
 / \  +--------+            +-------------+
           ^                        |
            +------------any--------+

                                                             Notification Dlvry. Svc.
                                                                 +---------------+
                                             +------INDPa------> |   Extended    |
  O   +--------+            +-------------+ /                +---+   Support     |
 /|\  | client | ---IPP---> | IPP Printer |                  |         for       |
 / \  +--------+            +-------------+              +---+       Delivery    |
           ^                                             |           Methods     |
            \                                            +-----------------------+
             \                                                        |
              +-------------------------any---------------------------+

                                                             Notification Dlvry. Svc.
                                                                 +---------------+
                                                                 |   Extended    |
  O   +--------+            +-------------+                  +---+   Support     |
 /|\  | client | ---IPP---> | IPP Printer | -----INDPb-----> |         for       |
 / \  +--------+            +-------------+              +---+       Natural     |
           ^                                             |          Languages    |
            \                                            +-----------------------+
             \                                                        |
              +-------------------------any---------------------------+

                                                             Notification Dlvry. Svc.
                                                                 +---------------+
                                                                 |   Extended    |
  O   +--------+            +-------------+                  +---+   Support     |
 /|\  | client | ---IPP---> | IPP Printer |                  |         for       |
 / \  +--------+            +-------------+ \            +---+     Subscription  |
           ^                                 +--INDPc--> |           Objects     |
            \                                            +-----------------------+
             \                                                        |
              +-------------------------any---------------------------+
```

Legend:

INDPx represents three different subsets of INDP operations the IPP Printer uses to communicate with the Notification Delivery Service to realize three different levels of support.

any represents any protocol, including INDP, that the IPP Printer or the Notification Delivery Service may support for notifying interested notification Recipients.

3.2.1 Notification without Notification Delivery Services

An IPP Printer working without the assistance of a Notification Delivery Service must implement on its own at least the minimum set of functionality required by the IPP Notification spec. This section gives a summary of the process a typical IPP Printer may employ to support IPP notifications on its own. The IPP Notification spec [ipp-ntfy] provides a detailed description of this process. Subsequent sections will describe how an IPP Printer may use INDP to indirectly accomplish some of the following tasks.

a) Creating a Subscription object. The IPP notification spec [ipp-ntfy] describes a number of mechanisms for IPP clients to request notification of an IPP printer. The end result, however, is that a Subscription object is instantiated at the IPP printer containing the information needed by the printer to know who to notify, how, and of what events.

b) Validating the Subscription object. At Subscription object instantiation time, the IPP printer validates its contents to make sure the requested events and delivery methods are supported. The IPP printer may also perform some validation on the recipient uri, requested natural language, and other information contained in the Subscription object.

c) Storing the Subscription object. The IPP printer provides persistent and non-persistent storage for Subscription objects until de object’s lease expires (in the case of per-printer subscriptions) or their associated print job is removed (in the case of per-job subscriptions). The IPP notification spec [ipp-ntfy] outlines the minimum number of Subscription objects a printer MUST be able to store. In practice, this requirement will vary widely depending on the administrative practices and usage patterns of the printer’s users.

d) Event condition. Normal printer operation as well as printer exception circumstances will cause event conditions to be raised.

e) Matching event with subscriptions. For each raised event condition the printer finds all the Subscription objects that request notification of that event. Rather than inspecting each Subscription object each time an event condition is raised, an IPP Printer may keep a list of the events the combined Subscription objects have requested to quickly discard event conditions no one is interested in.

f) Generating human-readable notification data. The IPP Printer examines each Subscription object found in step (e) to determine if it needs to generate human-readable notification information for it. IPP Printers with users of different language preferences may need to provide translation for multiple natural languages.

g) Dispatching the notification via the specified delivery method. The IPP Printer may need to generate slightly different Notification payloads for different delivery methods. With Notifications generated for each target Recipient, the IPP Printer uses its implementation of the delivery method specified in the corresponding Subscription object to dispatch the notification to its intended Recipient.

Though in this scenario the IPP Printer does not need to interact with a Notification Delivery Service, it may use INDP to dispatch Notifications encoded in “application/ipp” and transported over HTTP to interested notification Recipients. IPP Printers may use the Send-Notifications operation to accomplish this task.

3.2.2 Delivery method support extension (INDPa)

An IPP Printer may use a Notification Delivery Service simply to extend the list of delivery methods it supports. Doing so offloads a printer from having to implement all the common delivery methods its potential clients might require. It also enables a generic printer to support very specialized delivery methods implemented by a site’s Notification Delivery Service. Moreover, by using existing Notification Delivery Methods, an IPP Printer can take advantage of present, widely deployed notification infrastructure, standards-based or proprietary.

Using a Notification Delivery Service for the sole purpose of extending the notification delivering capabilities on and IPP Printer results in very small changes to the notification process described in the previous section. Specifically, the following changes apply.

1) Before accepting requests to create Subscription objects, step (a) above, the IPP Printer gets a list of the uri schemes the Notification Delivery Service supports and adds the values to its “notify-schemes-supported” attribute. To obtain this list, the IPP Printer uses the Get-Notify-Service-Attributes operation requesting the “notify-schemes-supported” attribute from the Notification Delivery Service. To an IPP client reading the printer’s “notify-schemes-supported" attribute, the entries with internal support and those supported via a Notification Delivery Service are indistinguishable.

2) During Subscription object validation, step (b) above, the IPP Printer may communicate with the Notification Delivery Service to validate a target uri requesting a delivery method implemented in the Notification Delivery Service. This IPP Printer accomplishes through the Validate-Notification-Uri operation.

3) For dispatching notifications that require a delivery method implemented in the Notification Delivery Service, step (g) above, the IPP Printer forwards the Notification on to the Notification Delivery Service through the Send-Notifications operation. The IPP Printer must provide the target uri and human-readable data, when the case requires it. The Notification Delivery Service is then responsible for creating a Notification payload suitable for the requested delivery method and for dispatching the notification to the specified Recipient.

3.2.3 Natural language support extension (INDPb)

An IPP Printer may use a Notification Delivery Service to generate human-readable notification data in addition to extending its delivery methods support. By using a Notification Delivery Service in this manner, an IPP Printer can dynamically support notifications in any number of natural languages, as long as the Notification Delivery Service being used supports them.

In addition to the modifications to the notification process listed in section 3.2, the following changes result from using a Notification Delivery Service to generate human-readable notification data.

1) Before accepting requests to create Subscription objects, step (a) above, the IPP Printer must communicate with the Notification Delivery Service to get a list of the natural languages it supports for human-readable message generation and add these values to its own “notify-natural-languages-supported” attribute.

ISSUE 01: Do we have any use for the printer description attribute “notify-natural-languages-supported”?

2) The IPP Printer no longer needs to perform steps (f) and (g) above. Instead it uses the Send-Notifications operation to send the Notification to the Notification Delivery Service along with the language specified in the corresponding Subscription object.

3.2.4 Subscription object management outsource (INDPc)

Through INDP an IPP Printer can employ the full services of a Notification Delivery Service, which includes storing and managing Subscription objects on behalf of the printer. Outsourcing this type of functionality greatly reduces the logic and resources requirements for an IPP Printer to support notification. Suitably hosted Notification Delivery Services can meet the notification needs of an environment without having to increase the capabilities of each printer in that environment. This section describes how an IPP Printer interacts with a Notification Delivery Service to accomplish this level of interaction.

This notification configuration requires the IPP Printer to establish a temporary registration with the Notification Delivery Service. Through a lease-based relationship, the Notification Delivery Service can keep track of what Subscription objects belong to what IPP Printer and generate the appropriate notifications when events are reported. This mechanism also enables the Notification Delivery Service to clean up orphaned Subscription objects. The IPP Printer uses the Register-Event-Producer operation to establish this type of relationship with the Notification Delivery Service. The model requires that an IPP Printer renew its lease periodically using the Renew-Registration operation.

When registering, an IPP Printer can specify a location for a Notification Delivery Service to store Subscription objects persistently. Subscription objects stored persistently in previous registrations are automatically re-instantiated when an IPP Printer registers with a Notification Delivery Service. The printer instructs the Notification Delivery Service what Subscription objects should be stored persistently and which one should be automatically disposed when the registration expires.

Once registered, the IPP Printer may forward requests to create Subscription objects on to the Notification Delivery Service. The IPP Printer uses the Create-Subscription operation to accomplish this task.

In this notification configuration an IPP Printer only needs to keep track of the superset of events requested by all the Subscription objects combined. The Notification Delivery Service assists the IPP Printer accomplish this task. First, in the response of a successful registration request, the Notification Delivery Service returns to the printer the list of events that it must generate to satisfy any Subscription objects that might have been reinstated from persistent storage. Then, in the response to every successful request to add or delete Subscription objects, the Notification Delivery Service returns to the printer a list of the new events needed and those to be discontinued as a result of the operation.

The following summarizes an IPP Printer’s process for handling notification when making full use of a Notification Delivery Service’s capabilities. For simplification, the description assumes that the IPP Printer supports these capabilities only via a Notification Delivery Service and not directly. However, for printers that implement some delivery methods internally and support others through a Notification Delivery Service, the notification process is a combination of the process outlined below and the one summarized in section 3.1.1.

a) Register with Notification Delivery Service. Early in its initialization process the IPP Printer should use the Register-Event-Producer operation to register with a Notification Delivery Service if configured to do so. It must indicate to the Notification Delivery Service the location of its persistent Subscription object storage, if applicable. The IPP Printer must store away the registration Id returned by the operation and remember any events listed in the response so it can start generating them.

b) Get Notification Delivery Service information. Right after registering with a Notification Delivery Service, the IPP Printer should query the Notification Delivery Service’s “notify-uri-schemes-supported” and “notify-natural-languages-supported” attributes. The printer must populate its “notify-uri-schemes-supported” and “notify-natural-languages-supported” attributes with the information obtained.

c) Create Subscription objects. When the IPP Printer receives a client request to create a new Subscription object, it must forward the request to the Notification Delivery Service using the Create-Subscription operation. This results in the Notification Delivery Service instantiating and validating a Subscription object. If the operation to create a new Subscription object succeeds, its response portion will tell the IPP Printer what, if any, new events it must generate to satisfy the new request. As with print jobs Subscription objects do not become active while the job is in “job-pending” state, the IPP Printer would not send a request to create a new Subscription object to the Notification Delivery Service until just before the job changes states from “job-pending”. For these types of notification requests, the IPP Printer may instead issue the Validate-Subscription operation to request that the Notification Delivery Service simply validate the request, thus allowing the printer to return an accurate status code to IPP operations requesting per-job notifications.

d) Event condition. The IPP Printer uses the consolidated list of events it maintains with the help of the Notification Delivery Service to know what events are of interest.

e) Send event report. When the IPP Printer raises an event condition, it reports the event to the Notification Delivery Service using the Send-Notification operation. At that point the IPP Printer is finished processing the event condition. The Notification Delivery Service is responsible for matching the event with the Subscription objects that requested it, generating any human-consumable data in the natural language specified in the Subscription object, and dispatching the appropriately formatted Notification using the requested delivery method.

4 Notification Operations

INDP makes extensive use of 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. INDP operations use the Operation Attributes group, but currently have no need for the Unsupported Attributes, Printer Object Attributes, and Job-Object Attributes groups. However, it uses a new attribute group, the Notification Attributes group.

The following operations form version 1.0 of INDP. All operations are targeted at the Server object. This section formally defines each INDP 1.0 operation.

- Get-Notify-Service-Attributes
- Validate-Notify-Target-Uri,
- Send-Notifications
- Register-Notification-Source
- Cancel-Notification-Source-Registration
- Renew-Notification-Source-Registration
- Create-Subscription
- Validate-Subscription
- Cancel-Subscription
- Renew-Subscription
- Get-Subscriptions

4.1 Get-Notify-Service-Attributes

This REQUIRED operation allows an IPP Printer to request the values of attributes of a Server object. In the request, the IPP Printer supplies the set of Server attribute names it’s interested in. In the response, the Service object returns a corresponding attribute set with the appropriate attribute values filled in.

4.1.1 Get-Notify-Service-Attributes Request

The following sets of attributes are part of the Get-Service-Attributes Request:

Group 1: Operation Attributes

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section 3.1.4.1.

“server-uri”:
The URI of the Notification Delivery Service.

“requested attributes” (1setOf keyword):

The IPP Printer OPTIONALLY supplies a set of attribute names in whose values the requester is interested. The Service object MUST support this attribute. If the IPP Printer omits this attribute, the Notification Delivery Service MUST respond with a list of all the attributes it supports and it respective values.

4.1.2 Get-Notify-Service-Attributes Response

The Server object returns the following sets of attributes as part of the Get-Notify-Service-Attributes Response:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566] section 3.1.4.1.

Group 2: Unsupported Attributes

A list of the attribute names requested by the IPP Printer but not supported by the Service object. See [rfc 2566] section 3.1.7 for details on returning Unsupported Attributes. As in version 1.0 of INDP all defined Service object attributes are mandatory, this group is a forward-looking feature when new OPTIONAL attributes may be defined.

Group 3: Server Object Attributes

This is the set of requested attributes and their current values. The Server object ignores any requested attribute that is not supported. The Service object MAY respond with a subset of the supported attribute and valued, depending on the security policy in force. However, the Service object MUS respond with the ‘unknown’ value for any supported attribute for which the Service object does not know the value. For a description of “out-of-band” values see [rfc 2566] section 4.1.

4.2 Validate-Notify-Target-Uri Operation

This REQUIRED operation allows an IPP Printer to request that the Notification Delivery Service validate a notification target uri. The Service object successfully validates the uri if the Notification Delivery Service implements the delivery method implied by the uri scheme or the target uri. The Service object is free to perform extended analysis on the validity of the recipient’s address provided in the uri is the semantics of the delivery method so allow.

4.2.1 Validate-Notify-Target-Uri Request

The following sets of attributes are part of the Validate-Notify-Target-Uri Request:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section 3.1.4.1.

“server-uri”:
The URI of the Notification Delivery Service.

“notify-target-uri” (uri):
The IPP Printer MUST supply this attribute. The Notification Delivery Service MUST support this attribute. It is the uri to be validated by the Server object.

4.2.2 Validate-Notify-Target-Uri Response

The Server object returns the following set of attributes as part of the Validate-Notify-Target-Uri Response:

Group 1: Operation Attributes

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566] section 3.1.4.1.

“validation-code” (boolean):
The Server object MUST return this attribute with a value of TRUE if the notify-target-uri was validates successfully; FALSE otherwise.

4.3 Send-Notifications Operation

This REQUIRED operation allows an IPP Printer to send one or more Notifications to a Notification Delivery Service. The Send-Notification operation can be used to transport Notification data in all four notification configurations described in section 3.2. Different attributes will be required depending on whether the operation is being used a) by an IPP Printer or Notification Delivery Service to send Notifications directly to a notification Recipient, b) by an IPP Printer to Send a localized Notification to a Notification Delivery Service (INDPa), c) by an IPP Printer to Send a Notification to be localized and dispatched by the Notification Delivery Service (INDPb), or d) by an IPP Printer to send a target-less notification using an established registration to a Notification Delivery Service (INDPc).

Both Machine-Consumable and Human-Consumable notifications may be included in the Send-Notification operation.

4.3.1 Send-Notifications Request

The following groups of attributes are 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 [rfc 2566] section 3.1.4.1.

Target:

The Target can a) The URI of the Notification Delivery Service if an IPP Printer is using Send-Notifications to dispatch notifications, or b) the URI of the Notification Recipient if the IPP Printer or the Notification Delivery Service are using the operation to dispatch notifications directly to a Notification Recipient.

“ultimate-target-uri”:

This attribute MUST be supplied by the IPP Printer when it uses the Send-Notifications operation to send notifications to a Notification Delivery Service without having registered as a Notification Source, i.e., configurations INDPa and INDPb above.

“registration-id”:

This attribute MUST be supplied by the IPP Printer when it uses the Send-Notifications operation to send notifications to a Notification Delivery Service after having registered a as a Notification Source, i.e., configuration INDPc above.

Group 2 to N: Notification Attributes

"human-readable-report" (text)

The Notification Source OPTIONALLY supports this attribute. This attribute is a text string generated by the IPP printer or Notification Delivery Service from the contents of the IPP Notification suitable for human consumption. If the Notification Source supports this attribute, it MUST supply this attribute if the Subscription object contains the "notify-text-format" (mimeMediaType) attribute. The text value of this attribute MUST be localized in the charset identified by the "notify-charset" (charset) attribute and the natural language identified by the notify-natural-language" (naturalLanguage) attribute supplied in the associated Subscription object that generates this event Notification. The format of the text value is specified by the value of the "notify-text-format" (mimeMediaType) supplied in the associated Subscription object.

“human-readable-report-format” (mime)

This attribute MUST be supplied by the Notification Source whenever the “human-readable-report” attribute is present. It indicates the format, e.g., text/plain, text/html, etc. of the “human-readable-report” attribute.

All of the REQUIRED attributes and any of the OPTIONAL attributes indicated in [ipp-ntfy] for a Push event Notification, including "notify-text-format-type" (mimeMediaType), if the "human-readable-report" (text) attribute is included, so that the Notification Recipient will know the text format of the "human-readable-report" (text) attribute value. These attributes communicate the same information as the notification attributes by the same name described in sections 7.4, 7.5, and 7.6 of [ipp-ntfy].

The rules that govern when each individual attribute MUST or MAY be included in this operation precisely mirror those specified in [ipp-ntfy] with the following exception: if the Send-Notifications operation is being used by an IPP Printer to communicate events to a Notification Delivery Service using a “registration-id”, Group 2 of this operation MUST only include the “trigger-event”, “trigger-time”, and “trigger-date-time” Notification attributes.

4.3.2 Send-Notifications Response

The target of the Send-Notifications operation, Notification Delivery Method or Notification Recipient, returns a status code for the entire operation and one for each Notification Report in the request if the operation's status code is other than "success-ok". If the Notification Recipient receives a Notification report that it can't pair up with a subscription it knows about, it can return an error status-code to indicate that events associated with that subscription should no longer be sent to it.

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section 3.1.4.1.

Group 2 to N: Notification Attributes

"notification-report-status-code" (type2 enum)

Indicates whether the intended target, i.e., Notification Delivery Service or Notification Recipient was able to consume the n-th Notification Report.

4.4 Register-Notification-Source Operation

This REQUIRED operation allows an IPP Printer to register itself as a Notification Source with a Notification Delivery Service. While registered, the Printer can add Subscription objects to the Server object. The Printer can then send Notifications to the Server object for the Server object to dispatch Notifications to all interested Recipients.

4.4.1 Register-Notification-Source Request

The following sets of attributes are part of the Register-Notification-Source Request:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section 3.1.4.1.

“server-uri”:

The URI of the Notification Delivery Service.

“registration-lease-time-requested” (integer(0:86,400)):

This REQUIRED attribute specifies the time in the future when the IPP Printer would like its registration lease to expire. When the Server object accepts a Registration request, it keeps track of this information. When the expiration time arrives, the Server object purges the registration.

An IPP Printer is able to extend its registration lease using the Renew-Notification-Source-Registration operation. The maximum value for a registration lease is one day.

“notification-source-name” (name(127)):

This REQUIRED attribute specifies the name of the IPP Printer. The Server object may use this information to organize current registrations. This information may also be useful to a Notification Delivery Service’s manager. Note: Management of a Notification Delivery Service is outside the scope of INDP v1.0.

“persistent-registration-storage-uri” (uri):

Through this OPTIONAL attribute an IPP Printer may communicate to the Service object where to retrieve persistent Subscriptions from previous registrations. The Service object also uses this location to store away future persistent Subscriptions. It the IPP Printer doesn’t provide this attribute, it will not be able to add Subscription objects that require persistent storage.

4.4.2 Register-Notification-Source Response

The Server object returns the following set of attributes as part of the Register-Notification-Source Response:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566] section 3.1.4.1.

“registration-id” (integer(0:MAX)):

The Server object MUST return the registration ID that the IPP Printer can use in subsequent calls such as Renew-Notification-Source-Registration, Create-Subscription, etc.

“notify-events” (1setOf type2 keyword):

If in this operation’s request the IPP Printer specifies a “persistent-registration-storage-uri” and as a result one or more Registrations are instantiated by the Server object during registrations, this attribute MUST contain the list of events that the printer must notify the Server object of to satisfy those Subscriptions.

“registration-lease-expiration-time” (integer(0:86,400)):

This REQUIRED attribute specifies the time in the future when the registration lease will expire. If the Server object is not able to grant the lease-time requested by the IPP Printer, this attribute may contain a different value that the one provided in the request.

An IPP Printer is able to extend its registration lease using the Renew-Notification-Source-Registration operation. The maximum value for a registration lease is one day.

4.5 Cancel-Notification-Source-Registration Operation

This REQUIRED operation allows an IPP Printer to terminate a current registration to a Notification Delivery Service. This causes the Server object to saves all current persistent Subscriptions into the location specified for this purpose at registration time, if one was specified. The Server object then cleans up any data and processes associated with that registration. Notification Delivery Service implementations should consider periodically saving away persistent Subscription objects to reduce the risk of failing to save everything at deregistration time.

4.5.1 Cancel-Notification-Source-Registration Request

The following set of attributes is part of the Cancel-Notification-Source-Registration Request:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section 3.1.4.1.

“server-uri”:

The URI of the Notification Delivery Service.

“registration-id” (integer(0:MAX)):

The IPP Printer MUST specify this REQUIRED attribute using the registration-id it obtained from the Server object via the Register-Notification-Source operation.

4.5.2 Cancel-Notification-Source-Registration Response

The Server object returns the following set of attributes as part of the Cancel-Notification-Source-Registration Response:

Group 1: Operation Attributes

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566] section 3.1.4.1.

“notify-events” (1setOf type2 keyword):
The Server object MUST return in this attribute the list of events that the printer must discontinue as a result of ending its registration to the Notification Delivery Service. This feature may be useful to IPP Printers that implement some delivery methods internally and others via a Notification Delivery Service and those who may use more than one Notification Delivery Service simultaneously.

4.6 Renew-Notification-Source-Registration Operation

This REQUIRED operation allows an IPP Printer to renew its lease on an existing registration to a Notification Delivery Service. It MUST be issued before the lease-time specified in the Register-Notification-Source operation or the previous Renew-Notification-Source-Registration operation expires.

4.6.1 Renew-Notification-Source-Registration Request

The following set of attributes is part of the Renew-Notification-Source-Registration Request:

Group 1: Operation Attributes

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section 3.1.4.1.

“server-uri”:
The URI of the Notification Delivery Service.

“registration-id” (integer(0:MAX)):
The IPP Printer MUST specify this REQUIRED attribute using the registration-id it obtained from the Server object via the Register-Notification-Source operation.

“registration-lease-time-requested” (integer(0:86,400)):
This REQUIRED attribute specifies the time in the future when the IPP Printer would like the registration lease to expire.

4.6.2 Renew-Notification-Source-Registration Response

The Server object returns the following set of attributes as part of the Renew-Notification-Source-Registration Response:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566] section 3.1.4.1.

“registration-lease-expiration-time” (integer(0:86,400)):

This REQUIRED attribute specifies the time in the future when the registration lease will expire. If the Server object is not able to grant the lease-time requested by the IPP Printer, this attribute may contain a different value that the one provided in the request.

4.7 Create-Subscription Operation

This REQUIRED operation allows an IPP Printer to cause a Subscription object to be instantiated in a Server object to which it is currently registered as a Notification Source. The Server object is responsible for keeping track of all registrations until their corresponding IPP Printer removes them via the Cancel-Subscription operation or until the registration is terminated by the Printer or it expires. The Server object uses Subscription object to know who and how to notify when it receives Notifications specifying a registration-id.

4.7.1 Create-Subscription Request

The Request for this operation includes the union of all of the REQUIRED attributes and any of the OPTIONAL attributes indicated in [ipp-ntfy] for the Create-Job-Subscription and Create-Printer-Subscription operations, with the following chages:

a) The “printer-uri” operational attribute is replaced by “server-uir” and MUST contain the URI of the Notification Delivery Service.
b) The request MUST include the operational attribute “registration-id” (integer(0:MAX)) specifying the registration-id the IPP Printer obtained from the Server object via the Register-Notification-Source operation.

The rules that govern when each individual attribute MUST or MAY be included in this operation precisely mirror those specified in [ipp-ntfy] for the Create-Job-Subscription and Create-Printer-Subscription operations, but obviously not simultaneously. If the request contains a “job-id” the Server object enforces applies the validation rules defined for the Create-Job-Subscription operation. If the “job-id” is not present, the Server object enforces the validation rules defined for the Create-Printer-Subscription operation.

4.7.2 Create-Subscription Response

The Response for this operation is defined to be identical to the Response for the Create-Printer-Subscription operation as specified in [ipp-ntfy] except for the following changes:

a) The Response MUST include the operational attribute “notify-events” (1setOf type2 keyword) containing the list of events that the printer must notify the Server object of to satisfy the creatioin of the new Subscription object.

b) The “notify-printer-up-time” operational attribute … ???

ISSUE 02: What should be done with this attribute. Should it be called the “notify-server-up-time” and be populated with the Notification Delivery Server’s up time, or should it be filled in by the printer? There are other ramifications here.

c) The Response does not include the “Unsupported Attribute” Group.

The Response that results from creating a job-related Subscription object doesn’t include the “notify-lease-expiration-time” and “notify-server-up-time” attributes.

4.8 Validate-Subscription Operation

This REQUIRED operation allows an IPP Printer to request the Sever object to validate the contents of what could become a Subscription object without actually creating the object. It employs the same logic used by the Create-Subscription operation to validate a request.

4.8.1 Validate-Subscription Request

The Request for this operation is identical to the Create-Subscription operation Request.

4.8.2 Validate-Subscription Response

The Server object returns the following set of attributes as part of the Validate-Subscription Registration Response:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566] section 3.1.4.1.

4.9 Cancel-Subscription Operation

This REQUIRED operation allows an IPP Printer to cause the Server object to cancel a Subscription object currently associated with a given registration-id.

4.9.1 Cancel-Subscription Request

The following set of attributes is part of the Cancel-Subscription Request:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section 3.1.4.1.

“server-uri”:

The URI of the Notification Delivery Service.

“registration-id” (integer(0:MAX)):

The IPP Printer MUST specify this REQUIRED attribute using the registration-id it obtained from the Server object via the Register-Notification-Source operation.

“subscription-id” (integer(0:MAX)):

This REQUIRED attribute specifies the ID of the Subscription object to be cancelled. The IPP Printer must provide here the same “subscription-id” that it received back from the Create-Subscription or Get-Subscriptions operations.

4.9.2 Cancel-Subscription Response

The Server object returns the following set of attributes as part of the Cancel-Subscription Response:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566] section 3.1.4.1.

“notify-events” (1setOf type2 keyword):

The Server object MUST return in this attribute the list of events that the printer must discontinue as a result of canceling the Subscription object.

4.10 Renew-Subscription Operation

The REQUIRED Renew-Subscription operation permits an IPP Printer to request the Server object to extend the lease on a Subscription object instance. This operation is only valid for Subscription object that don’t specify a “job-id”, or Per-Printer Subscription objects as they are referred to in [ipp-ntfy].

4.10.1 Renew-Subscription Request

The following set of attributes is part of the Renew-Subscription Request:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section 3.1.4.1.

“server-uri”:

The URI of the Notification Delivery Service.

"registration-id" (integer(0:MAX)):
The IPP Printer MUST specify this REQUIRED attribute using the registration-id it obtained from the Server object via the Register-Notification-Source operation.

"subscription-id" (integer(0:MAX))

The IPP Printer MUST specify the ID of the Subscription object whose lease is being extended.

"notify-lease-time-requested" (integer(0:MAX))

The IPP Printer MUST specify the time by which it wishes to extend the Subscription object's lease.

4.10.2 Renew-Subscription Response

The Server object returns the following set of attributes as part of the Renew-Subscription Response:

Group 1: Operation Attributes

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as defined in [rfc 2566] section 3.1.4.1.

"subscription-lease-expiration-time" (integer(0:86,400)):
This REQUIRED attribute specifies the time in the future when the Subscription's lease will expire. If the Server object is not able to grant the lease-time requested by the IPP Printer, this attribute may contain a different value that the one provided in the request.

ISSUE 04: What shall be done with the "notify-printer-up-time" operational attribute?

4.11 Get-Subscriptions Operation

This REQUIRED operation allows an IPP Printer to get a list of the Subscription objects associated with a given registration ID.

4.11.1 Get-Subscriptions Request

The Request for this operation is defined to be identical to the Request for the Get-Subscriptions operation as specified in [ipp-ntfy], except for the following changes:

a) The "printer-uri" operational attribute is replaced by "server-uir" (uri) and MUST contain the URI of the Notification Delivery Service.
b) The request MUST include the operational attribute "registration-id" (integer(0:MAX)) specifying the registration-id the IPP Printer obtained from the Server object via the Register-Notification-Source operation.

4.11.2 Get Subscriptions Response

The Response for this operation is defined to be identical to the Response for the Get-Subscriptions operation as specified in [ipp-ntfy].

ISSUE 05: What shall be done with the Subscription object attribute “notify-printer-up-time”?

5 Encoding of the Operation Layer

INDP uses the same operation layer encoding model and syntax as IPP [ipp-pro] with the following extensions:

5.1 New attribute tag

A new notification attributes tag is defined:

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

5.2 New status codes

ISSUE 06 - Should we move the status codes into the Notification Model document in order to have the same status codes for any other delivery method that might be defined?

The following status codes are defined:

5.2.1 unknown-notification-recipient. (0xXXX)

The Notification Recipient returns this status code in order to indicate that the intended Ultimate Notification Recipient is not known to the Notification Recipient.

5.2.2 unable-to-delivery-notification-report (0xXXX)

The Notification Recipient returns this status code in order to indicate that it was unable to deliver the event Notification to the intended Ultimate Notification Recipient.

5.2.3 successful-ok-but-cancel-subscription (0xXXXX)

The Notification Recipient indicates that it no longer wants to receive Notifications for this Subscription object. Therefore, the Subscription object is canceled. Note: this status code allows the Notification Recipient to cancel a Subscription object without having to be the owner of the Subscription object. Only the owner of the Subscription object can cancel a Subscription object using the Cancel-Subscription operation.

5.2.4 unknown-registration-id (0xXXX)

5.2.5 successful-ok-but-error-accessing-persistent-storage (0xXXXX)

5.3 Encoding

The encoding of INDP is based strictly on the encoding used by IPP. This specification, however, defines a new Group tag which is used it to encode multiple notifications in a Request. As multiple instances of the same group type have only been included in operation Responses in the past, this section describes the encoding of an operation that uses the new tag for illustration purposes.

The encoding for the Send-Notification Request consists of:

```
      -----------------------------------
      |        version-number            | 2 byte
      -----------------------------------
      |         operation-id             | 2 bytes
      -----------------------------------
      |          request-id              | 4 bytes
      -----------------------------------
      |     operation-attributes-tag     | 1 byte
      -----------------------------------
      |        attributes-charset        | u bytes
      -----------------------------------
      |    attributes-natural-language   | v bytes
      -----------------------------------
      |        target-attribute          |        w bytes
      ----------------------------------------------
      |   notification-attributes-tag    | 1 byte  |
      -----------------------------------          | - 1 or more
      |      notification-attr-list      | x bytes |
      ----------------------------------------------
      |      end-of-attributes-tag       | 1 byte
      -----------------------------------
```

Where:

version-number is made up of a major-version-number of %d1 and a minor-version-number of %d0 indicating the 1.0 version of the 'ipp-notify-send' event notification delivery method and protocol.

operation-id, in the 1.0 version of the protocol, can only be 0x00003, Send-Notification.

request-id is any 4 byte number provided by the notification source and must be matched by the notification recipient in the corresponding response to a request. It assists the notification source in associating operation responses with their corresponding requests. Note that this request id is independent of the request id embedded in the notification report, which is opaque to the delivery method but assists the notification recipient order and identity missing or duplicate notification reports.

operation-attribute tag, natural-language-attribute, *charset-attribute, target-attribute, and end-of-attributes-tag* have the same syntax and semantics as in [ipp-pro].

notification-attr-list contains a list of the attributes that make up a single notification (see section 2 above) encoded using the syntax specified in [ipp-pro].

The encoding for the Send-Notification Response consists of:

```
     ----------------------------------
     |         version-number         | 2 byte
     ----------------------------------
     |          status-code           | 2 bytes
     ----------------------------------
     |           request-id           | 4 bytes
     ----------------------------------
     |    operation-attributes-tag    | 1 byte
     ----------------------------------
     |       attributes-charset       | u bytes
     ----------------------------------
     |  attributes-natural-language   | v bytes
     ----------------------------------
     |        target-attribute        | w bytes
     -------------------------------------------
     |  notification-attributes-tag   | 1 byte  |
     ----------------------------------         | - 1 or more
     |        ntfy-status-code        | 2 bytes |
     -------------------------------------------
     |     end-of-attributes-tag      | 1 byte
     ----------------------------------
```

6 Encoding of Transport Layer

HTTP/1.1 [rfc2616] is the transport layer for this protocol.

The operation layer has been designed with the assumption that the transport layer contains the following information:

- the URI of the target INDP operation.

- the total length of the data in the operation layer, either as a single length or as a sequence of chunks each with a length.

It is REQUIRED that a Notification Delivery Service and a 'indp://' Notification Recipient implementation support HTTP over the IANA assigned Well Known Port XXX (INDP's default port), though a notification recipient implementation MAY support HTTP over some other port as well.

Each HTTP operation MUST use the POST method where the request-URI is the object target of the operation, and where the "Content-Type" of the message-body in each request and response MUST be "application/ipp-notify-send". The message-body MUST contain the operation layer and MUST have the syntax described in section 3, "Encoding of Operation Layer". An INDP client implementation (be it an IPP Printer or a Notification Delivery Service) MUST adhere to the rules for a client described for HTTP1.1

[rfc2616]. An INDP server implementation (be it a Notification Delivery Method or a notification Recipient) MUST adhere the rules for an origin server described for HTTP1.1 [rfc2616].

An INDP server implementation sends a response for each request that it receives. If it detects an error, it MAY send a response before it has read the entire request. If the HTTP layer of the INDP server implementation completes processing the HTTP headers successfully, it MAY send an intermediate response, such as "100 Continue", with no notification data before sending the notification response. The INDP client implementation MUST expect such a variety of responses. For further information on HTTP/1.1, consult the HTTP documents [rfc2616].

An INDP server implementation MUST support chunking for HTTP notification requests, and an INDP client implementation MUST support chunking for HTTP notification responses according to HTTP/1.1[rfc2616]. Note: this rule causes a conflict with non-compliant implementations of HTTP/1.1 that don't support chunking for POST methods, and this rule may cause a conflict with non-compliant implementations of HTTP/1.1 that don't support chunking for CGI scripts

INDP uses 'indp://' as its URI scheme.

7 IANA Considerations

IANA will be asked to register this 'ipp-notify-send' notification delivery scheme and protocol and will be asked to assign a default port.

8 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.

9 Security Considerations

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

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 Notification Source.

9.1 Security Conformance

Notification Sources (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.

10 References

[ipp-mod]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and Semantics", <draft-ietf-ipp-model-v11-04.txt>, June, 1999.

[ipp-ntfy]
Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-02.txt>, February 2, 2000.

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

[rfc2026]
S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

[rfc2616]
R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.

[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.

11 Author's Addresses

Hugo Parra
Novell, Inc.
122 E 1700 S
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

12 Full Copyright Statement