INTERNET-DRAFT 6 ISSUES are highlighted like this.
<draft-ietf-ipp-not~~ification~~-spec-00.doc>

S. Isaacson
Novell, Inc.
J. Martin
Underscore
R. deBry
Utah Valley State College
T. Hastings
Xerox Corporation
M. Shepherd
Xerox Corporation
R. Bergman
Dataproducts Corp.
~~July~~ August ~~22~~25, 1999

Internet Printing Protocol/1.0 & 1.1: **IPP Event Notification Specification**

Status of this Memo

Abstract

This document describes an extension to the IPP/1.0 & IPP/1.1 model that allows ~~end users~~a client to subscribe to printing related events. Subscriptions include "Per-Job Submission subscription" and "~~Explicit~~ Per-Printer subscriptions".

One or more Per-Job Submission subscriptions are specified by the ~~end-user~~client when submitting a job. ~~Explicit~~ One or more Per-Printer subscriptions are created by performing separate explicit ~~subscription~~ Create-Printer-Subscription operations on the Printer~~ or an identified Job object~~. Four other operations are defined for Per-Printer subscription objects: get attributes, get subscriptions, renew a subscription, and cancel a subscription.

A subscription includes:

_the names of Job and/or Printer events, ~~that are of interest to the subscriber~~

the ~~delivery methods and addresses to use for notifications (socket, email, etc.)~~Notification Recipient URL, the Notification content MIME type, possibly some opaque data and the charset and natural language. In addition, the Per-Printer subscription includes: the requested lease time and persistency for the subscription. ~~Job Submission Subscriptions: A simple method is provided for subscribing to printing related events when a job is submitted:~~

- ~~Two new subscription attributes are supplied by the client as part of an IPP create request (Print-Job, Print-URI, Create-Job, Validate-Job): "notify-recipients (1setOf uri)" and "notify-events (1setOf type2 keyword)".~~

~~An event is some occurrence (either expected or unexpected) within the printing system. Events can be classified along two dimensions:~~

- ~~Either as Job Events or Printer Events, and~~
- ~~Either as Errors, Warnings, or Reports~~

~~"Explicit Subscriptions": Three new Printer operations are specified: Subscribe-Job, Subscribe-Printer, Renew-Subscription, and Unsubscribe. The semantics of the two subscription Operation attributes ("notify-events" and "notify-recipients") used in create operations is preserved. These two subscription attributes are supplied by the client as operation attributes in this new Subscribe operation, along with a new "notify-lease-time" operation attribute.~~

~~A client can remove a subscription explicitly using the Unsubscribe operation or can let the lease time expire. A client can extend the lease time for a subscription using the Renew-Subscription operation.~~

When the event occurs, a notification is generated and delivered using the information specified in the ~~job's~~ subscription~~ which was submitted with the job~~.

The full set of IPP documents includes:

Design Goals for an Internet Printing Protocol [IPP-REQ]
Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [IPP-RAT]
Internet Printing Protocol/1.1: Model and Semantics [IPP-MOD]
Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]
Internet Printing Protocol/1.0: Implementer's Guide [IPP-IIG]
Mapping between LPD and IPP Protocols [IPP LPD]

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. Operator and administrator requirements are out of scope for version 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 specifications, and gives background and rationale for the IETF working group's major decisions.

The "Internet Printing Protocol/1.1: Model and Semantics", 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. It defines the encoding rules for a new Internet MIME media type called "application/ipp". This document also defines the rules for transporting over HTTP a message body whose Content-Type is "application/ipp". This document defines a new scheme named 'ipp' for identifying IPP printers and jobs. Finally, this document defines interoperability rules for supporting IPP/1.0 clients.

The "Internet Printing Protocol/1.0: Implementer's Guide" document gives insight and advice to implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.0 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 6
1.1 Notification Overview 6
2 Model for Per-Job and Per-Printer Subscription and Event Notification 8
2.1 Model for Per-Job Subscription and Notification 8
2.2 Model for Per-Printer Subscription and Notification 10
2.3 Relationship between the Printer object and the Notification Delivery Service 10
3 Terminology 12
3.1 Conformance Terminology 12
3.2 Other terminology 12
4 Object Model for Notification 14
4.1 Object relationships 15
5 Object attributes defined for Notification 16
5.1 "job-notify" (1setOf collection) Job Description attribute 16
5.2 Subscription object attributes 17
5.3 Common Per-Job and Per-Printer Subscription attributes 18
5.3.1 notify-recipient (uri) 18
5.3.2 notify-events (1setOf type2 keyword) 20
5.3.3 notify-content-type (mimeMediaType) 23
5.3.4 subscriber-user-data (octetString(63)) 24
5.3.5 notify-attributes-charset (charset) 24
5.3.6 notify-attributes-natural-languages (naturalLanguage) 24
5.3.7 request-id 25
5.4 Additional attributes for the Subscription object *only*....25
5.4.1 subscription-id (integer (1:MAX)) 25
5.4.2 notify-lease-expiration-time (integer(0:MAX)) 25
5.4.3 printer-uri (uri) 26
5.4.4 subscriber-user-name (name(MAX)) 26
5.4.5 notify-printer-up-time (integer(1:MAX)) 26
5.5 Printer Description Attributes related to Notification 26
5.5.1 notify-schemes-supported (1setOf uriScheme) 27
5.5.2 notify-events-default (1setOf type2 keyword) 27
5.5.3 notify-events-supported (1setOf type2 keyword) 27
5.5.4 max-events-supported (integer(5:MAX)) 27
5.5.5 max-job-subscriptions-supported (integer(1:MAX)) 28
5.5.6 max-printer-subscriptions-supported (integer(0:MAX)) 28
5.5.7 notify-lease-time-supported (rangeOfInteger(0:MAX)) 28
5.5.8 notify-lease-time-default (integer(0:MAX)) 28
5.5.9 persistent-jobs-supported (boolean) 28
5.5.10 persistent-subscriptions-supported (boolean) 28
5.5.11 printer-state-change-time (integer(1:MAX)) 29
5.5.12 printer-state-change-date-time (dateTime) 29

6 Notification Content 29
6.1 Notification content MIME media type formats 29
6.2 Notification content attributes common to Job and Printer events 30
6.3 Additional Notification content attributes for Job events *only* 31
6.4 Additional Notification content attributes for Printer events *only* 32
7 Operations for Per-Job Subscriptions 36
7.1 Job Creation Operations (Create-Job, Print-Job, Print-URI) and Validate-Job 36
7.2 Get-Printer-Attributes operation 37
7.3 Get-Job-Attributes operation 37
7.4 Set-Job-Attributes operation 38
8 Per-Printer Subscriptions and the Subscription object 38
8.1 Create-Printer-Subscription operation 39
8.2 Get-Printer-Subscription-Attributes operation 41
8.3 Get-Printer-Subscriptions operation 41
8.4 Renew-Printer-Subscription operation 42
8.5 Cancel-Printer-Subscription operation 44
9 Comparison of Per-Job Submission Subscriptions versus Per-Printer Subscriptions 44
10 Conformance Requirements 55
11 IANA Considerations 55
12 Internationalization Considerations 55
13 Security Considerations 56
14 Status Codes 56
14.1 client-error-uri-notification-scheme-not-supported (0x04??) 56
14.2 server-error-too-many-subscriptions (0x04??) 57
14.3 server-error-too-many-events (0x04??) 57
15 References 57
16 Author's Addresses 58
A. Appendix - Summary of Notification Attribute Usage 60
17 Appendix C: Full Copyright Statement 65

1 Introduction

This IPP notification specification is an extension to IPP/1.0 [RFC2568, RFC2569]. This document in combination with the following documents is intended to meet the notification requirements described in [ipp-not-req]:

Internet Printing Protocol/1.0 & 1.1: "Collection Attribute Syntax" [ipp-coll]
Internet Printing Protocol/1.0 & 1.1: "Job Progress Attributes" [ipp-prog]
Internet Printing Protocol/1.0 & 1.1: "Notification Change History" [ipp-not-hist]

The rest of this document is laid out as follows:

- The rest of Section 1.1 is an overview of IPP Notification.
- Section 2 is the model for network entities that use IPP notification, including clients (desktop and servers), IPP Printers (servers and devices), and Notification Recipients.
- Section 3 is the terminology used throughout the document.
- Section 4 is the object model for notification, including Job, Printer, and Subscription objects.
- Section 5 defines the notification attributes for each of the Job, Printer, and Subscription objects.
- Section 6 defines the content of Human Consumable and Machine Consumable Event Notifications.
- Sections 7, 8, and 9 define the Per-Job and Per-Printer Subscription mechanisms.
- Section 10 and 11 define the conformance requirements and IANA requirements, respectively.
- Section 12 - 14 cover Internationalization, Security, and Status codes.

1.1 Notification Overview

A client can establish an event notification subscription so that when one of the specified events occurs, an asynchronous Notification is sent to a specified Notification Recipient.

One or more Per-Job Submission subscriptions are specified by the client when submitting a job. One or more Per-Printer subscriptions are created by performing separate explicit Create-Printer-Subscription operations on the Printer.

A Per-Job or Per-Printer subscription includes:

1. the names of Job and/or Printer events that are of interest to the Notification Recipient
2. the delivery method and address to use to deliver the notification to one Notification Recipient
3. whether Human Consumable or Machine Consumable notification content is to be sent
4. some opaque data that the subscriber wants to be send to the Notification Recipient in the Notification, perhaps to identify either the subscriber or the ultimate recipient
5. the charset to use in the Notification, if it is to be different than the one used in the request that created the subscription

6. the natural language to use in the Human Consumable Notification, if it is to be different than the one used in the request that created the subscription

In addition, a Per-Printer subscription includes:

7. the requested lease time in seconds for the subscription
8. whether or not the subscription is requested to be persistent across power cycles.

For Per-Job subscriptions, a client requests job and printer event notification using the "job-notify" operation attribute when creating a job with any of the Job Creation operation: Print-Job, Print-URI, and Create-Job. The "job-notify" operation attributes may be submitted to the Validate-Job in order to be validated. The "job-notify" operation attribute is copied to the Job object as a Job Description attribute and so may be queried using the Get-Job-Attributes and Get-Jobs operations (see [ipp-mod]). Also this "job-notify" Job Description attribute may be changed using the Set-Job-Attributes (see [ipp-set2]). The "job-notify" operation attribute contains one or more collection values, each consisting of a number of member attributes the specify a subscription, so that a Job can contain more than one Per-Job subscription. The 'collection' is a new attribute syntax (see [ipp-coll]). Table 1 summarizes the Per-Job Subscription operations and their salient input operation attributes. All operations are REQUIRED, except the Set-Job-Attributes operation.

Table 1 - Summary of Per-Job Subscription operations

Operation:	salient inputs besides printer-uri:
Print-Job, Print-URI, Create-Job	1setOf {recipient, [events,] [content-type,] [user-data,] [charset,] [natural-language]}
Validate-Job	1setOf {recipient, [events,] [content-type,] [user-data,] [charset,] [natural-language]}
Get-Job-Attributes	job-id, [requested-attributes]
Set-Job-Attributes	job-id, 1setOf {[recipient,] [events,] [content-type,] [user-data,] [charset,] [natural-language]}

For Per-Printer subscriptions, a client requests job and printer event notification using new operations independent of any job. The Printer keeps each subscription in a separate Subscription object which is similar to a Job object in its operations and security. The Get-Printer-Attributes (see [ipp-mod]) returns the supported notification capabilities supported. The Create-Printer-Subscription operation creates an instance of the Subscription object supplying these new operation attributes and returns a subscription-id (analogous to a job-id for a Job object). These operation attributes are copied to the Subscription object as Subscription Description attributes and so may be queried using the Get-Subscription-Attributes and Get-Subscriptions operations. The subscriber requests a lease time for each Per-Printer subscription which MAY be infinite. The Printer grants a lease time according to its configured policy. A client MUST renew the Subscription before the granted lease time expires using the Renew-Printer-Subscription operation. Table 2 summarizes the Per-Printer-Subscription operations and their salient input operation attributes.

Table 2 - Summary of Per-Printer Subscription operations

Operation:	salient inputs beside printer-uri:
Get-Printer-Attributes	[requested-attributes]
Create-Printer-Subscription	recipient, [events,] [content-type,][user-data,][charset,][natural-language,][lease-time-requested,] [persistence-requested]
Get-Printer-Subscription-Attributes	subscription-id, [requested-attributes]
Get-Printer-Subscriptions	[my-printer-subscriptions,] [requested-attributes]
Renew-Printer-Subscription	subscription-id, [lease time-requested]
Cancel-Printer-Subscription	subscription-id

There are two steps that IPP notification must take regarding each event – an internal event recording, and an external notification:

1) As an events occurs, the printer internally records in the job objects and the printer objects those events which are required to be supported by the system and those that are subscribed to by a notification recipient.

2) As an events occurs, ~~for each event~~ the Printer searches the set of subscriptions for any interest in that event. As the Printer finds that some notification recipient is interested in that event (the notification recipient is subscribed to the event), the "request-id" sequence number for that event is incremented and a notification is generated and delivered using the methods and target addresses identified in the subscription. The "request-id" sequence number permits a Notification Recipient to detect duplicate notifications due either to duplicate subscriptions or retries and to detect dropped notifications.

2 Model for Per-Job and Per-Printer Subscription and Event Notification

2.1 Model for Per-Job Subscription and Notification

~~"~~Per-Job ~~Submission~~ subscriptions~~"~~ are created by ~~the end-user~~a client (desktop or server acting as a client) ~~upon~~ as part of creation of the job in an IPP Printer (printing device or server).. ~~"Explicit Subscriptions" are accomplished by performing independent operations on the printer or a certain job.~~More than one subscription may be submitted with a job. The IPP Printer object delivers ~~the~~ a Notifications to the Notification Recipient~~(s)~~ supplied by the Client in each subscription. A Notification Recipient can be the Job submitter or a third party.

Figure 1 shows the Per-Job subscription notification model for a simple Client - Printer relationship.

```
embedded printer:
                                           output device or server
        desktop or server                     +---------------+
       ~~O~~   +--------+                         |  ###########  |
      ~~/|\~~  | client |---IPP job submission------># Printer #  |
      / \  +--------+  Per-Job subscription   |  # Object  #  |
      ---                                     |  #####|#####  |
       0   +---------+                        +-------|-------+
      /|\  |recipient|<-----Notifications-------------+
      / \  +---------+    (Job and/or Printer events)
```

Figure 1 - Client-Printer Per-Job ~~Submission~~Subscription~~"~~ and Notification Model

Figure 2 shows a (spooling or non-spooling) Server that implements two Printer objects (1 and 2) that represent two devices. The devices A and B ~~that~~ in turn each implement an IPP Printer object (3 and 4, respectively). ~~The server implements one IPP Printer object (1 and 2) for each such device.~~ The Server implementation has three choices for how to support Per-Job subscriptions to the client (and itself):

1. forward the Per-Job subscriptions to the down stream IPP Printer and let it perform the notification directly to the Notification Recipients supplied by the Client (Notifications(C)) and use Per-Printer Subscriptions for the Server's own purposes.

2. save the client-supplied Per-Job subscription on the Job object in the server and substitute its own Per-Job subscription with the Server as the Notification Recipient (Notifications(B)). Then the Server relays Notifications to the client-supplied Notification Recipients (Notifications(A)).

3. A combination of 1 and 2 in which the Server adds its own Per-Job subscriptions to those supplied by the client. Thus the IPP Job that goes to Printer object 4 has a combination of subscription information from both the Client and the Server. This latter approach is sometimes called "piggy-backing" because the Server is adding its Per-Job subscription information to that supplied by the client. Piggy-backing is especially useful, if device B also accepts (IPP or non-IPP) requests from other servers. Then when all the jobs from Server S have been completed by device B, there will be no more Job events sent to Server S. (Server S could still maintain a long term Per-Printer subscription with Printer D to that Server S can have Printer B's state track (shadow) that of Printer D or Server S could poll Printer D when queried about Printer B).

```

                                                          device A
                           server S                    +------------+
                        +------------+                 | ###########|
+--------+   IPP Job    | ###########|    IPP Job      | # Printer #|
| client |--submission--->#  Printer #|---submission----->#  Object 3#|
+--------+   Per-Job    | # Object 1#|    Per-Job      | ###########|
           subscription | ###########|   subscription  +------------+
                        |            |                    device B
+--------+   IPP Job    | ###########|                 +------------+
| client |--submission--->#  Printer #|    IPP Job      | ###########|
+--------+   Per-Job    | # Object 2#|---submission----->#  Printer #|
           subscription | ###|#######|    Per-Job      | # Object 4#|
                        +----|---^---+   subscription  | ####|#|####|
+--------+                   |   |                     +-----|-|----+
|Notific-|<-Notifications(A)-+   +--- Notifications(B)-------+ |
|ation Re|<----------------Notifications(C)--------------------+
|cipient |
+--------+
```

Figure 2 - Client-Server-Printer Per-Job Subscription and Notification Model

2.2 Model for Per-Printer Subscription and Notification

Per-Printer subscriptions are created by a client (an end user, an operator, or a server acting as a client) using a Create-Printer-Subscription operation that is independent of Job Submission. The Printer object (printing device or server) creates a Subscription object to hold the attributes supplied by the subscriber. The client creates separate Per-Printer subscriptions if more than one Notification Recipient is desired. The Printer delivers Notifications to the Notification Recipient specified by each Per-Printer subscription. A Notification Recipient may be the subscriber or a third party. Figure 3 shows the Per-Printer subscription notification model for the Client - Printer relationship where the client may be an end user, an operator, or a server acting as a client.

```
       desktop or server                    server or printing device
                                              +---------------+
           +--------+                         |  ###########  |
           | client |---IPP job submission------># Printer #  |
           +--------+  Per-Job subscription   |  # Object  # |
                                              |  #####|#####  |
           +---------+                        +-------|-------+
           |recipient|<-----Notifications-------------+
           +---------+    (Job and/or Printer events)
```

Figure 3 - Client-Printer Per-Printer Subscription and Notification Model

2.3 Relationship between the Printer object and the Notification Delivery Service

~~Note: This~~ The IPP Notification model does not mandate that the IPP Printer object implement the full semantics of subscription, report generation, and multiple delivery methods itself. An ~~simple (embedded)~~ implementation may be configured to use some other notification service. Figure 4 shows this partitioning.

```
           Create Request with              ###########
       ----Subscriptions------------------->#  IPP    #
                                            # Printer #
                                            # Object  #
                                            ###########
                                                |
                                         *******|**********
                                       *    Subscriptions   *
                                     *       & Events        *
                                   *            |
                                  *        +----v---------+
                                 *         | notification |
       <---Job and Printer ------*---------| service      |
           Notifications         *         +--------------+
                                 *
                                 *

     *** = Implementation configuration opaque boundary
```

Figure 4 - Opaque Use of a Notification Service

If an IPP Printer does use some other notification service, it MAY do so in either of two ways:

1. The IPP Printer forwards the Subscriptions and the Events to the other notification service which then sends the Notifications to the proper Recipients.

2. The IPP Printer keeps the subscriptions and uses them to send Notifications to the other notification service which forwards them to Recipients that have also subscribed directly with the other notification service. Examples of this latter approach are paging and immediate messaging services.

In any case, the interface between the IPP Printer and the other notification service is outside the scope of this document and is intended to be transparent to this specification.

~~If "Explicit Subscription" is implemented, the IPP Printer MUST support three new Printer operations: Subscribe-Printer, Renew-Subscription, and Unsubscribe. These operations allow a client that is not submitting a job to subscribe, renew a subscription, and to unsubscribe independently from job submission. In addition, the Subscribe-Job operation permits a client to add a subscription to an already submitted job.~~

~~A REQUIRED operation attribute, "notify-lease-time", specifies the length of time in seconds that the subscription is to be in effect, after which the subscription is automatically canceled.~~

~~A REQUIRED operation attribute, "subscription-id", allows a client to maintain a reference to the applicable subscriptions made and know which notifications were generated as a result of this subscription.~~

~~The subscription created by a Subscribe-Job or Subscribe-Printer operation is called an "Explicit Subscription". Just like Job Submission Subscriptions, these subscriptions can include job and/or printer events. Subscriptions for Job Events in Subscribe-Job operation apply to that job only, while Job events in~~

~~the Subscribe-Printer operation apply to all jobs. For the Subscrbe-Printer subscriptions, it is most likely that the subscribing entity is either a server-based device management application submitting jobs to a device which has IPP embedded in it or an accounting application collecting accounting data.~~

3 Terminology

This section defines terminology used throughout this document.

3.1 Conformance Terminology

Capitalized terms, such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED NOT, and OPTIONAL, have special meaning relating to conformance to this specification. These terms are defined in [ipp-mod section 13.1 on conformance terminology, most of which is taken from RFC 2119 [RFC2119].

Since support of this entire notification specification is OPTIONAL for conformance to IPP/1.0 or IPP/1.1, the use of the term REQUIRED in this document means "REQUIRED if this notification specification is implemented". Likewise, the term RECOMMENDED means "RECOMMENDED if this notification specification is implemented".

READ-ONLY - indicates an attribute ~~the~~ that MUST NOT be settable using the Set-Job-Attributes or Set-Printer-Attributes operations (see [ipp-set2]).

3.2 Other terminology

Job Submitting End User - A human end user who submits a print job to an IPP Printer. This person may or may not be within the same security domain as the Printer. This person may or may not be geographically near the printer.

Administrator - A human user who established policy for and configures the print system.

Operator - A human user who carries out the policy established by the Administrator and controls the day to day running of the print system.

Job Submitting Application - An application (for example, a batch application), acting on behalf of a Job Submitting End User, which submits a print job to an IPP Printer. The application may or may not be within the same security domain as the Printer. This application may or may not be geographically near the printer.

Security Domain - The set of network components which can communicate without going through a proxy or firewall. A security domain may be geographically very large, for example - anyplace within IBM.COM.

IPP Client (or client) - The software component (desktop or server) that sends an IPP operation request to an IPP Printer object (server or printing device) and accepts the resulting operation response from the IPP Printer object.~~on the client system which implements the IPP protocol~~.

Job Recipient - A human who is the ultimate consumer of the print job. In many cases this will be the same person as the Job Submitting End User, but need not be.

Example: If I use IPP to print a document on a printer in a business partner's office, I am the Job Submitting End User, while the person I intend the document for in my business partner's office is the Job Recipient. Since one of the goals of IPP is to be able to print near the Job Recipient of the printed output, we would normally expect that person to be in the same security domain as, and geographically near, the Printer. However, this may not always be the case. For example, I submit a print job across the Internet to a Kinko's print shop. I am both the Submitting End User and the Job Recipient, but I am neither near nor in the same security domain as the Printer.

Job Recipient Proxy - A human acting on behalf of the Job Recipient. In particular, the Job Recipient Proxy physically picks up the printed document from the Printer, if the Job Recipient cannot perform that function. The Proxy is **by definition** geographically near and in the same security domain as the printer.

Example: I submit a print job from home to be printed on a printer at work. I'd like my secretary to pick up the print job and put it on my desk. In this case, I am acting as both Job Submitting End User and Job Recipient. My secretary is acting as a Job Recipient Proxy

Notification Subscriber (or **Subscriber**) - A client that requests the IPP Printer to send Event Notifications to one or more Notification Recipients. A Notification Subscriber may be:

1. a Job Submitting End User or Job Submitting Application (desktop or server) that is submitting a job or
2. an End User, an Operator, or an Administrator that is not submitting a job.

Subscription - A request by a Notification Subscriber to the IPP Printer to send Event Notifications to a specified Notification Recipient when the event occur. A Subscription is represented as a ~~The~~ set of attributes that indicate the "what, where, who, and how" for notification. Notifications are generated for certain events (what) and delivered using various delivery methods (how) to certain addresses (where and who).

Per-Job ~~Submission~~ Subscription - A Subscription that a client specifies as part of a create job operation (Print-Job, Print-URI, Create-Job), or a~~and~~ Validate-Job operation~~)~~.

~~Explicit~~ Per-Printer Subscription - A Subscription that a client specifies using an explicit ~~operation, Subscribe-Job, Subscribe-Printer~~Create-Printer-Subscription operation, that is attached to a ~~Job or~~ Printer object~~, respectively~~.

Notification Source - The entity that sends Event Notifications.

Notification Recipient - ~~Any~~ The entity identified as a recipient within a subscription that receives IPP Notifications about Job and/or Printer events. ~~Some~~ A Notification Recipient~~s~~ ~~are~~ may be a: Job Submitting End User, ~~s~~ Job Submitting Application (desktop or server), Job Recipient, Job Recipient Proxy, Operator, or Administrator, etc., and their representative or log file or usage statistics gathering application or other active or passive entities. ~~and others are interested third parties, such as the Job Recipient or Job Recipient Proxy.~~

Notification Recipient Agent - A program which receives Notifications on behalf of the Notification Recipient. The agent may take some action on behalf of the recipient, forward the notification to the recipient via some alternative means (for example, page the recipient), or queue the notification for later retrieval by the recipient.

Event - An event is some occurrence (either expected or unexpected) within the printing system of a change of state, condition, or configuration of a Job or Printer object. A property of an event is that it only occurs at one instant in time and does not span the time the physical event takes place. For instance, jam-occurred and jam-cleared are two distinct events. The jam-occurred event is reported only when the jam initially occurs~~. Each event is recorded internally when it occurs. In addition the event is reported externally~~ and only if there is one or more event subscriptions outstanding for that event.

Events can be classified along two dimensions:

- Either as Job Events or Printer Events, and

- Either as Errors, Warnings, or Reports

A Job event is some interesting state change in the Job object, and a Printer event is some interesting change in the Printer object.

A report event is purely informational, such as 'job-completed' or 'accepting-jobs'. A warning is not serious and processing continues. An error is serious and either the job is aborted or the printer stops. These are typical uses of the terms report, warning, and error, although the actual usage is implementation dependent.

An event occurs for a job or printer whether any entity ~~is registered~~has subscribed to be notified for that event or not. ~~The most recent event(s) of all possible events are recorded in a job object and a printer object, and~~ A notification is only generated depending on the set of subscriptions outstanding.

Notification - When an event occurs, a Notification is generated that fully describes the event (what the event was, where it occurred, when it occurred, etc.). Notifications are delivered to each ~~all the~~ Notification Recipient~~s~~ that has a subscription ~~are subscribed to~~ that includes the event, if any. The Notification is delivered to the address of the Notification Recipient using the notification delivery method defined in the subscription. However, a Notification is sent ONLY if there is a corresponding subscription.

Notification Delivery Method (or **Delivery Method** for short**)** - Notifications are delivered using a method, such as email, TCP/IP, etc.

Immediate Notification - Notifications that are delivered using a delivery method which is not store-and-forward (e.g. TCP connection, UDP datagram). This can be on the order of several minutes subject to network latency.

Store and Forward ~~**Queued**~~ **Notification -** **A** Notification which are not necessarily delivered to Notification Recipients immediately, but is queued for delivery by some intermediate network application, for later retrieval. Email is an example of a store and forward notification delivery method.~~Notifications that are delivered using a delivery method which has some sort of store-and-forward mechanism (e.g., email).~~

Human Consumable Notification - Notifications that are intended to be consumed by human End Users only. ~~(Further define)~~They are simple text that has been localized for the Notification Recipient as specified in the subscription. Programs are not intended to parse Human Consumable Notification, since it is localized and the content depends on implementation. There is no standardized format.

Machine Consumable Notification - Notifications that are intended for consumption by a program only**.** ~~(Further define)~~They use the encoding of an IPP response. The Notification Recipient must localize the contents, if displaying it to a human.

~~**Mixed Format Notification** - A mixed notification may contain both human consumable and machine consumable information.~~

4 Object Model for Notification

This section describes the notification object model that adds a REQUIRED Subscription object which together with the Job and Printer object provide the complete notification semantics. ~~The Subscription~~

~~object is OPTIONAL. The Subscription object MUST be supported if Per-Printer subscriptions are supported and vice-versa.~~

Per-Job subscriptions do not use the Subscription object. Instead, a Job can have one or more Per-Job subscription which is represented entirely as a single Job attribute~~s~~. Each Per-Job subscription is represented as a collection value of the "job-notify" Job attribute.~~A "notify-event-exclude-mask" attribute allows some of the specified events to be excluded from some of the specified Notification Recipients in the Job subscription.~~

The object relationships can be seen pictorially as:

```
Subscription objects (Per-Printer Subscriptions)     Printer object
+----+                                                +------------+
| s1 |<---------------------------------------------->|            |
+----++                                               |            |
 | s2 |<--------------------------------------------->|     p1     |
 +----++                                              |            |
  | s3 |<-------------------------------------------->|            |
  +----+                                              +------------+
                 Job objects
                  +---------+
                  |         |
                  |   j1    |
                  |         |
                  +----+    |
                  | s4 |    |      s4 is a Per-Job subscription
                  ++---+----++
                   |         |
                   |   j2    |
                   |         |
                   +----+----|
                   | s5 | s6 |      s5 and s6 are~~is a~~ Per-Job subscriptions
                   ++---+----++
                    |         |
                    |   j3    |
                    |         |
                    |         |
                    +---------+
```

s1, s2, and s3 are Per-Printer Subscription objects and can identify Printer and/or Job events.

s4, s5, and s6~~5~~ are Per-Job subscriptions (not objects) and can identify Printer and/or Job events.

4.1 Object relationships

The object relationships can be stated as follows:

1. The Printer object contains zero or more Subscription objects (p1 contains s1-s3 Subscription objects).

2. Each Subscription object (s1, s2, and s3) is contained in one Printer object (p1) and represents one Per-Printer subscription.

3. Each "Per-Printer" Subscription object identifies one or more Job and/or Printer events. Such Job events are for *all* jobs on the Printer. Such Printer events are for *any* Printer event, no matter which job is processing and when no jobs are processing.

4. Each~~The~~ "Per-Job" subscription (not an object by itself, but is attributes of the Job object) identifies one or more Job and/or Printer events. Such Job events are only for *this* job (different than "per-Printer" Subscriptions). Such Printer events are for *any* Printer event, no matter which job and when no jobs are processing (same as for "per-Printer" Subscriptions).

5. A Job object contains zero or more~~one~~ Per-Job subscriptions (not an object). Job j1 contains Per-Job subscription s4, Job j2 contains Per-Job subscription s5 and s6, and Job j3 does not contain a Per-Job subscription.

6. A Subscription object cannot be contained in or associated with more than one Printer object.

7. A Per-Job subscription cannot be contained in or associated with more than one Job object.

5 Object attributes defined for Notification

The following notification attributes are defined for each of the Printer, Job, and ~~(new, but OPTIONAL)~~ Subscription objects. The ~~explanations about some of~~ definitions of the object attributes are ~~collected~~ specified here so that they can be referred to from the subsequence definitions of the operations that set them. ~~All Job, Printer, and Subscription object attributes are READ-ONLY, i.e., they MUST NOT be settable with the Set-Job-Attributes or Set-Printer-Attributes operations. Instead, all of them are set as a side effect of another operation or when an event notification is delivered. Attributes and objects not in the current, July 22, spec are labeled as "new":~~

5.1 "job-notify" (1setOf collection) Job Description attribute

In order to permit more than one Per-Job subscription, each subscription is represented as a collection value of the "job-notify (1setOf collection)" Job Description attribute. An implementation MUST support at least one collection value of the "job-notify" attribute. Table 3 lists the member attributes of the "job-notify" attribute. Section 5.3 defines each of the "job-notify" member attributes.

The "job-notify" Job Description attribute MAY be modified by the Set-Job-Attributes operation (see section 7.4) in which case all of its member attributes are replaced and the READ-ONLY "request-id" member attribute is reset with a value of '0' (see section 5.3.7).

Table 3 - Job object 'job-notify' colletion member attributes

Job object attributes:	Printer support	~~READ-ONLY, ~~set by:
notify-recipient~~s~~ (~~1setOf ~~uri)	R	job creation operation, Set-Job-Attributes
notify-events (1setOf type2 keyword)	R	job creation operation, Set-Job-Attributes
notify-content-type (mimeMediaType)	R	job creation operation, Set-Job-Attributes
~~notify~~subscriber-user-data (octetString(63))	R	job creation operation, Set-Job-Attributes
~~notify-exclude-event-masks (1setOf octetString(8))~~	~~R~~	~~job creation operation~~
notify-charset (charset)	O	job creation operation, Set-Job-Attributes
notify-natural-languages'' (~~1setOf ~~naturalLanguage)	O	job creation operation, Set-Job-Attributes

request-id (integer(0:MAX))	R	Printer - event occurring
~~job-trigger-event (type2 keyword)~~	~~R~~	~~Printer - event occurring~~
~~job-trigger-time (integer(MIN:MAX))~~	~~R~~	~~Printer - event occurring~~
~~job-trigger-date-time (dateTime)~~	~~O~~	~~Printer - event occurring~~
~~notify-request-ids (1setOf integer(0:MAX))~~	~~R~~	~~Printer - event occurring~~

5.2 Subscription object attributes

Each Per-Printer subscription is contained in a separate Subscription object. An implementation MUST support at least one Subscription object instance. Table 4 lists the Subscription attributes defined for the Subscription object. Sections 5.3 and 5.4 define each of the Subscription attributes.

All of the Subscription object attributes are READ-ONLY. They are set by other operations.

Table 4 - Subscription object attributes

Subscription object attributes:	Printer support	READ-ONLY, set by:
notify-recipient~~s~~ (~~1setOf~~ uri)	R~~C~~	client - Create-Printer-Subscription
notify-events (1setOf type2 keyword)	R~~C~~	client - Create-Printer-Subscription
notify-content-type (mimeMediaType)	R	client - Create-Printer-Subscription
subscriber~~notify~~-user-data (octetString(63))	R~~C~~	client - Create-Printer-Subscription
~~notify-exclude-event-mask (1setOf octetString(8))~~	~~C~~	~~client - Create-Printer-Subscription~~
notify-charset (charset)	O	client - Create-Printer-Subscription
notify-natural-languages (1setOf naturalLanguage)	O	client - Create-Printer-Subscription
~~notify-~~request-id (~~1setOf~~ integer(0:MAX))	R~~C~~	Printer - event occurring
~~delivery-failure-count (integer(0:MAX))~~	~~C~~	~~Printer - event occurring~~

Not in a Per-Job subscription:

subscription-id (integer(1:MAX))	R~~C~~	Printer - Create-Printer-Subscription
notify-lease-expiration-time (integer(0:MAX))	R~~C~~	Printer - Create-Printer-Subscription, Renew-Printer-Subscription
printer-uri (uri)	R	Printer - Create-Printer-Subscription
subscriber-user-name (name(MAX))	R~~C~~	Printer - Create-Printer-Subscription
notify-printer-up-time (integer(1:MAX))	R	Printer - returned by Get-Printer-Subscription-Attributes and Get-Printer-Subscriptions

~~ISSUE 4 - It isn't necessary to add a "persistence (boolean)" attribute to the Subscription object in order to record the value passed in the Create-Printer-Subscription operation and to allow it to be queried?~~

5.3 Common Per-Job and Per-Printer Subscription attributes

The following sub-sections define attributes that are in common to both Per-Job and Per-Printer subscriptions, i.e., are member attributes of the "job-notify" collection Job Description attribute and are Subscription object attributes. Section 5.4 defines additional attributes that are only Subscription object attributes.

~~5.2.1~~5.3.1 notify-recipient~~s~~ (~~1setOf~~ uri)

~~The client supplies this operation attribute in a create request in order to subscribe for job and/or printer events while this job is "not-completed". In order to claim conformance to this notification specification,~~

~~the Printer object MUST support this attribute.~~ This REQUIRED attribute describes both where (the address of the Notification Recipient~~s~~) and how (the delivery method) notifications are to be delivered to the Notification Recipient when any of the events specified in the "notify-events" attribute occur. ~~If the client does not supply this attribute in a create request, the Printer object MUST NOT provide any Job Submission Subscription notification for this job.~~

There are potentially many different notification delivery methods for IPP notifications, standardized as well as proprietary. This document does not define any of these delivery mechanisms; they will each be described in separate complementary documents.

Each of the notification delivery method documents must provide at least the following information:

1) The URI scheme used.

2) The supported and default delivery format, and if not one of the specified types in Section 5.3.3, description of the notification content.

3) Any content length restrictions imposed by the delivery protocol.

4) The latency of the delivery protocol used.

5) The reliability of the transport and delivery protocol used.

6) The security aspects of the transport and delivery protocol used, e.g. how it is handled in firewalls.

7) How the delivery protocol is initiated, e.g. does it have to be initiated by the receiving user (pull), or is it initiated by the notification service (push).

ISSUE 1: Once a number of delivery solutions have been developed and evaluated, we may want to make one or several of them REQUIRED for implementation to ensure a minimum set of interoperability. Which one or ones should be REQUIRED?

~~Unless specified otherwise, a delivery method may be used with any event and an even may be used with any delivery method.~~

~~ISSUE 2: Delete previous sentence that says that some delivery methods can defined to not use some events?~~

~~IPP Printer objects MUST support the **'ipp-tcp-notify'** and **'ipp-udp-notify'** delivery methods in order to conform to this notification specification. Support of the other methods is OPTIONAL.~~

~~ISSUE 1 - Which delivery method or methods is/are REQUIRED?~~

~~Standard uriScheme values are:~~

~~**'mailto'**: a 'multipart/report' [RFC1892] message is sent via email to the specified email address. It MUST consist of both a "text/plain" part for display to a user and an 'application/ipp' part which is a notification that a program can process (see Section 1). This delivery method ignores the supplied values of the "notify-events" attribute and implies the 'job-completed' event (new state is 'completed', 'aborted', or 'canceled'). The notification recipient does not acknowledge receipt of the mail message.~~

~~**'http'**: an IPP notification is sent using an HTTP POST to the indicated URL.~~

~~**'ipp-tcp-notify':** (REQUIRED) an IPP notification is sent via a TCP/IP socket that is opened by the Printer object on the IP address specified in the URI using the specified port using the "host:port" HTTP convention. For example:~~

~~ipp-tcp-notify://foo.com:6000~~

~~If the port is omitted, the default port is TBD. The "application/ipp" notification content format is used for this method (see Section 6.1).~~

~~The event recipient does not respond or acknowledge the notification.~~

~~ISSUE 3: For TCP/IP delivery, what about leaving the connection open versus having to reestablish a connection for each event? Who specifies: client in subscription, Printer implementation, Notification Recipient, Administrator?~~

~~**'snmpv1-notify':** a notification is sent as an SNMPv1 trap to the host specified as the address in the URI. The notification recipient does not acknowledge receipt of the notification (trap).~~

~~**'snmpv2-notify':** a notification is sent as an SNMPv2 inform to the host specified as the address in the URI. The notification recipient does acknowledge receipt of the notification (inform).~~

~~**'snmpv3-notify':** a notification is sent as an SNMPv3 inform to the host specified as the address in the URI. The notification recipient does acknowledge receipt of the notification (inform).~~

~~ISSUE 4 – Move SNMP trap delivery methods to another document? Which one?~~

~~**'ipp-udp-notify':** (REQUIRED) an IPP notification is sent via a UDP datagram that is opened by the Printer object on the IP address specified in the URI using the specified port using the "host:port" HTTP convention. For example:~~

~~ipp-udp-notify://bar.com:6000~~

~~If the port is omitted, the default port is TBD (see). The UDP datagram contains the "application/ipp" notification content format (see Section 1). The notification recipient does not acknowledge receipt of the notification.~~

~~If the client specifies a "notify-recipients" URI scheme that is not supported by the printer, the Printer MUST return the [new] 'client-error-notify-uri-scheme-not-supported' in the IPP response in reply to the create request.~~

~~ISSUE 5 - Which URL parameters should we mention (which like SLP) are removed before being used?~~

5.3.2 notify-events (1setOf type2 keyword)

~~The client OPTIONALLY supplies this operation attribute in a create request. In order to claim conformance to this notification specification, the Printer object MUST support this attribute. ~~This REQUIRED attribute identifies the job and/or printer events that are to be delivered to the Notification Recipient~~s~~ as Notifications as defined in section 6. If the client ~~does ~~did not supply this attribute ~~in a create request~~when supplying the subscription, ~~but does supply the "notify-recipients", ~~the Printer object populates this attribute with its "notify-events-default" attribute value (see section 5.5.2).

There are both job events and printer events. Each job and printer event is assigned a keyword to use in this attribute and in the Notification.

The ~~following ~~events are defined to be disjoint. ~~Therefore, the "job-trigger-event" attribute is single valued, since two events cannot be sent in the same notification content.~~

A Printer MUST support the events indicated as "REQUIRED".

The standard job event keyword values are:

'none': REQUIRED - no notifications of any events (an IPP object can use this value to indicate that it is configured not to support event notification; a client would not subscribe to this event).

'job-created': REQUIRED - the Printer object has accepted ~~the create~~ a job creation operation (Print-Job, Print-URI, or Create-Job) and the job's "~~job-~~time-at-creation" attribute value is set (see [ipp-mod] section 4.3.14.1). ~~(i.e., when the job is created no matter whether it puts~~ The Printer puts the job in the 'pending', ~~or~~ 'pending-held' or 'processing' states~~)~~. ~~The IPP Printer MUST record this event.~~

Note: This event is separate from the 'job-state-changed' event so that it can be subscribed to without having to get every job state change event~~,~~ for a Notification Recipient that is only interested in when the job is first created.

'job-completed': REQUIRED - the job has reached one of the completed states, i.e., the value of the job's "job-state" attribute has changed to: 'completed', 'aborted', or 'canceled'. The Job's "time-at-completed" and~~/or~~ "date-time-at-completed" (if supported) attributes are set (see [ipp-mod] section 4.3.14). ~~The IPP Printer MUST record this event.~~

Note: This event is separate from 'job-state-changed' so that it can be subscribed to without having to get every job state change event~~,~~ for a Notification Recipient that is only interested in when the job is completed.

~~ISSUE 19 - Should there be more job attributes in the 'job-completed' notification content, such as "impressions-completed" and "sheets-completed"?~~

'job-state-changed': REQUIRED - the job has changed from any state ~~or state-reason~~ to any other state and/or a value has been added or removed from the job's "job-state-reasons" attribute, except when the job is created or when the job moves to any of the "completed" job states~~, i.e., the value of the job's "job-state" attribute changes to any value, except~~ ('completed', 'aborted', or 'canceled').

This event also indicates that one or more values have been added to or removed from the Job's "job-state-reasons" attribute, such as 'job-queued' or 'job-printing', whether or not the job's state has changed. If job state reasons are added when the job is created, only the 'job-created' event is generated, in order to keep the events disjoint. If job state reasons are added or removed when the job is completed, only the 'job-completed' event is generated, in order to keep the events disjoint. ~~This event is REQUIRED to be recorded if the job-state-reason is an error.~~

A client that wants to subscribe to all job state changes, including creation and completion, includes the 'job-created', 'job-state-changed', and 'job-completed' in the notification subscription. When a job is finally removed from the Job History (see [ipp-mod] 4.3.7.1) no event is generated, i.e., neither a 'job-state-changed' event nor a 'job-purged' event is generated.

~~ISSUE 20 - Should 'job-state-changed' event be REQUIRED to support, since 'printer-state-changed' is?~~

'job-config-changed': OPTIONAL – when the configuration of a job has changed, i.e., the value of the "job-message-from-operator" or any of the non-READ-ONLY Job attributes have changed, such as any of the job template attributes or the "job-name" attribute. Typically, such a change is the result of the user or the operator performing a Set-Job-Attributes operation (see [ipp-set2]) on the

Job object. The client ~~would have to~~ performs a Get-Job-Attributes to find out the new values of the changed attributes. This event ~~would be~~is useful for GUI clients and drivers to update the job information to the user.

'job-purged': OPTIONAL - when a 'not-completed' job (i.e., not 'completed', 'canceled', or 'aborted') was purged from the printer using the Purge-Jobs operation. No event, including this event, is generated when a job is aged out of the Job History or moved out explicitly with the Purge-Jobs operation.

'job-progress' - a sheet or copy has completed. See separate [ipp-prog] spec.

~~ISSUE 21 - any other job events to be defined and added?~~

~~ISSUE 22 - should any more events be REQUIRED to be supported?~~

The standard Printer~~-report~~ event keywords values are~~ include~~:

'none': - REQUIRED - no notification of any events (an IPP object can use this value to indicate that it is configured not to support event notification; a client would not subscribe to this event).

'printer-restarted': OPTIONAL - when the printer is powered up or the Restart-Printer operation is performed (see [ipp-set2~~adm~~]). ~~The IPP Printer MUST record this event.~~

Note: This event is separate from the 'printer-state-changed' event so that it can be subscribed to without having to get every printer state change event, for a Notification Recipient that is only interested in when the Printer first comes up.

'printer-shutdown': OPTIONAL - when the device is being powered down or the Shutdown-Printer operation has been performed with either power-off or standby options (see [ipp-set2~~adm~~]).

Note: This event is separate from 'printer-state-changed' so that it can be subscribed to without having to get every Printer state change event, for a Notification Recipient that is only interested in when the Printer is powered down or shutdown.

'printer-state-changed':~~-~~ REQUIRED - the Printer changed state, i.e., the value of the Printer's "printer-state", "printer-state-reasons" (whether "printer-state" changed or not), and/or "printer-is-accepting-jobs" attributes changed, except when the Printer starts up or is shutdown. ~~If the "printer-state-reasons" changed, then one or more values have been added to or removed from the Printer's "printer-state-reasons" attribute, such as 'moving-to-paused' or 'connecting-to-printer'.~~

~~This event also indicates that one or more values have been added to or removed from the Printer's "printer-state-reasons" attribute, such as 'moving-to-paused', 'toner-low', or 'media-jam', whether or not the Printer's state has changed.~~ If printer state reasons are added when the Printer is started up, only the 'printer-restarted' event is generated, in order to keep the events disjoint. If printer state reasons are added or removed when the printer is powered-down or shutdown, only the 'printer-shutdown' event is generated, in order to keep the events disjoint. ~~This event is REQUIRED to be recorded if the printer-state-reason is an error.~~

A client that wants to subscribe to all printer state changes, including restart and power-down/shutdown, includes the 'printer-restarted', 'printer-state-changed', and 'printer-shutdown' in the notification subscription.

'printer-media-changed': OPTIONAL – when the media loaded on a printer has been changed, i.e., the "media-ready" attribute has changed. This event includes both an actual media change and filling an empty input tray with the same or different media. The client must check the "media-ready" Printer attribute (see [ipp-mod] section 4.2.11) separately to find out what new media was loaded or filled.

'printer-config-changed': OPTIONAL – when the configuration of a Printer has changed, i.e., the value of the "printer-message-from-operator" or any non-READ-ONLY Printer attribute has changed, except for "media-ready" (which has its own event), whether through the Set-Printer-Attributes operation or by other means and whether initiated by a human or not. For example, any "xxx-supported", "xxx-default", "printer-message-from-operator", etc. values have changed. The client ~~would have to~~ has to perform a Get-Printer-Attributes to find out the new values of these changed attributes. This event ~~would be~~ is useful for GUI clients and drivers to update the available printer capabilities to the user.

'printer-queue-changed': OPTIONAL - the order of jobs in the Printer's queue has changed, so that an application that is monitoring the queue can perform a Get-Jobs operation to determine the new order. This event does not include when a job enters the queue (the 'job-created' event covers that) and does not include when a job leaves the queue (the 'job-completed' event covers that).

'printer-no-longer-full': OPTIONAL – when the Printer can now accept a Print-Job, Print-URI, Create-Job, Send-Document, or Send-URI request. This event is used when there is more than one client feeding a printer/server (fan-in), and the Printer may still be printing but has acquired more buffer space to accept jobs. This event only occurs when the Printer did not have room to accept jobs previously and rejected a Print-Job, Print-URI, Create-Job, Send-Document, or Send-URI operation.

'printer-almost-idle': OPTIONAL – when the Printer needs another Job in order to stay busy. This event is used when a spooler is feeding more than one printer/server (fan-out), and the spooler holds jobs until a Printer requests them, rather than committing jobs to IPP Printers before it is necessary. This event MAY be used by~~for~~ a Printer implementation to request a new job from any subscribers sufficiently ahead of time so that the device does not run out of work between jobs.

5.3.3 notify-content-type (mimeMediaType)

This REQUIRED attribute specifies the type of content that is to be sent in the notifications. Thus the ~~client~~ subscriber can control whether the event notification content is Human Readable or~~,~~ Machine Readable~~, or both~~. Most delivery methods support both forms of content type and MUST define one of them as the default. If a delivery method only supports one form of content type, then that form is also the default.

If the MIME media type registration permits a charset parameter, than such a specification ~~SHALL~~ MUST be used (instead of the "notify-charset" ~~member~~ attribute) in order to indicate the charset to be used in the notification content.

If the Subscriber did not supply this attribute when requesting the subscription, the Printer object populates this attribute from the default value defined for the delivery method.

Standard mimeMediaType values are:

~~'multi-part/alternative' - contains both human consumable notification content using the 'text/plain' MIME media type and machine consumable notification content using the 'application/ipp' MIME media type with the Get-Job-Attributes response encoding of the attributes listed in Table 3 or the Get-Printer-Attributes response encoding of the attributed listed in Table 4. This value SHALL be supported and is the default, if the client does not supply the "notify-content-type" member attribute.~~

~~ISSUE 15: Should we make this attribute 1setOf so that the additional values could specify which alternatives are to be used with 'multi-part/alternative'?~~

'application/ipp' - the Machine Consumable notification content using the 'application/ipp' MIME media type [ipp-mod] as defined in section 6~~with the Get-Job-Attributes response encoding of the attributes listed in or the Get-Printer-Attributes response encoding of the attributes listed in~~ .

~~'text/plain' - the Human Consumable notification content. If the charset is other than US-ASCII, the /charset parameter SHALL be included in the value of this attribute and in the event notification content. RFC 2046 indicates that the absence of the charset parameter SHALL mean US-ASCII rather than simply unspecified [RFC2046]. Examples:~~
~~'text/plain': A plain text document in US-ASCII [US-ASCII]~~
~~'text/plain; charset=US-ASCII': A plain text document in US-ASCII.~~
~~'text/plain; charset=ISO-8859-1': A plain text document in ISO 8859-1 (Latin 1) [ISO8859-1].~~
'text/plain; charset=utf-8': A plain text document in ISO 10646 represented as UTF-8 [RFC-2044] as defined in section 6.
~~'text/plain, charset=iso-10646-ucs-2': A plain text document in ISO 10646 represented in two octets (UCS-2) [ISO10646-1]~~

5.3.4 subscriber~~notify~~-user-data (octetString(63))

~~ISSUE 6 - Should we add a third operation attribute to Job Submission Subscriptions, Subscribe-Job, and Subscript-Printer that is an opaque octet-string that is passed to the Notification Recipient on every notification, as in Jini? This operation attribute would be OPTIONAL for the client to supply and the Printer to support.~~

This REQUIRED attribute holds opaque information being sent from the Subscriber to the Notification Recipient~~(s)~~, such as the identify of the Subscriber or a path or index to some Subscriber information. Or it MAY contain a key that the Notification Recipient needs in order to process the Notification, such as the ultimate recipient, if the Notification Recipient is a general application that in turn forwards notifications. ~~This information is the same for all Notification Recipients. Per-Notification Recipient data should be specified in the URL registration so that they may be sent as parameters in each URL.~~

5.3.5 notify-attributes-charset (charset)

This OPTIONAL attribute specifies the charset to be used in the ~~human readable part of the~~ Notification content ~~that is~~ sent to the notification recipient~~s that the client supplied in this same collection value~~. This attribute ~~SHALL~~ MUST NOT be used when the "notify-content-type" attribute value specifies the charset parameter in its MIME media type value.

~~If the "notify-attributes-charset" attribute is not supplied, the charset supplied in the "attributes-charset" Operation attribute SHALL be used, if the charset value is supported by the Printer, else the Printer object SHALL use the Printer's "charset-configured" value.~~

5.3.6 notify-attributes-natural-languages (naturalLanguage)

This OPTIONAL attribute specifies the natural language for the IPP object to use in the ~~human readable part of the~~ Notification content that is sent to the Notification Recipient~~s that the client supplied in this same collection value~~. ~~If this attribute is not supported or the supplied value is not supported, the IPP~~

~~Printer SHALL return the attribute in the Unsupported Attributes Group but still accept the operation, as with all create operations. If this attribute is not supplied or the attribute or value is not supported by the Printer object, the natural language supplied in the "attributes-natural-language" create operation attribute SHALL be used, if that natural language value is supported by the Printer, else the Printer object SHALL use the Printer's "natural-language-configured" value. See the Print-Job operation in [ipp-model].~~

5.3.7 request-id

This REQUIRED READ-ONLY attribute holds the most recent request-id~~s~~ sequence number delivered in a ~~Job~~ Notification content to the Notification Recipient. ~~for a Per-Job Subscription for this Job for each of the Job's Notification Recipients. In other words, the "notify-request-ids" is a parallel attribute to the Job's "notify-recipients (1setOf uri) attribute.~~ A value of 0 indicates that no Notifications have been sent for this subscription~~Job~~. The first request-id sent for a ~~job~~ subscription MUST be 1. ~~The reason to have a separate request-id for each recipient is so that when more than one Notification Recipient is specified in a subscription,~~ Each Notification Recipient ~~will have~~has its own monotonically increasing series~~set~~ of request-ids, i.e., no gaps, in order to be able to detect a missing notification.

For a Per-Job subscription, "request-id" is a READ-ONLY member attribute of each collection value. All of the other member attributes (see section 5.1) are not READ-ONLY and MAY be modified by a Set-Job-Attributes operation (see 7.4).

5.4 Additional attributes for the Subscription object *only*

The following sub-sections define additional attributes that are only for the Subscription object and have no counter part in Per-Job subscriptions.

5.4.1 subscription-id (integer (1:MAX))

This REQUIRED READ-ONLY Subscription object attribute uniquely identifies this Subscription object instance on this Printer. The Printer object, on acceptance of a Create-Printer-Subscription request, generates an ID which identifies the new Subscription object on that Printer. The Printer returns the value of the "subscription-id" attribute as part of the response to a Create-Printer-Subscription request. The 0 value is not included to allow for compatibility with "job-id" and with SNMP index values which also cannot be 0.

It is RECOMMENDED that Per-Printer Subscription objects be persistent. Then the Subscription objects including the subscription-id remains unique across power-cycles. Even if an implementation does not make Per-Printer subscription objects persist, the implementation SHOULD make every attempt not to re-use subscription ids that subscribers might still think are valid. In other words, the Printer SHOULD at least keep the next subscription-id to be assigned in non-volatile memory.

5.4.2 notify-lease-expiration-time (integer(0:MAX))

This REQUIRED READ-ONLY Subscription object attribute specifies the time in the future when the subscription lease will expire, i.e., the "printer-up-time" value at which the lease will expire. When the Printer object creates a Subscription object, it populates this attribute with the appropriate value. When the indicated time arrives, the Printer MUST delete the Per-Printer Subscription object.

A client is able to extend a lease using the Renew-Printer-Subscription operation (see section 8.4). A value of 0 indicates an infinite time, if such a policy is supported as indicated in the "notify-lease-time-supported" (integer(0:MAX)) Printer Description attribute (see section 5.5.7) and the subscriber is authorized to request an infinite lease.

Note: In order to compute the number of seconds remaining in a Per-Printer Subscription lease, a client can subtract the "notify-printer-up-time" Printer object attribute (see section 5.4.5) from the "notify-lease-time" Subscription object attribute.

5.4.3 printer-uri (uri)

This REQUIRED READ-ONLY Subscription object attribute identifies the Printer object that created this Subscription object. When a Printer object creates a Subscription object, it populates this attribute with the Printer object URI that was used in the create request. This attribute permits a client to identify the Printer object URI that was used to create this Subscription object, i.e., what security scheme was used.

5.4.4 subscriber-user-name (name(MAX))

This REQUIRED READ-ONLY Subscription object attribute contains the name of the user that created the Subscription object. The Printer object sets this attribute to the most authenticated printable name that it can obtain from the authentication service over which the IPP operation was received. This attribute is intended to help a human user determine for which Per-Printer Subscriptions they are the Subscriber. Only if such is not available, does the Printer object use the value supplied by the client in the "requesting-user-name" operation attribute of the create operation (see [IPP-MOD] Sections 4.4.2, 4.4.3, and 8).

ISSUE 2 - What is the value of the "subscriber-user-name" when the Subscriber is a server send a job to an IPP Printer? The original Job Submitting End User use name or some "user name" that identifies the Server?

Note: The Printer object needs to keep an internal originating user id of some form, typically as a credential of a principal, with the Subscription object. Since such an internal attribute is implementation-dependent and not of interest to clients, it is not specified as a Subscription Description attribute. This originating user id is used for authorization checks (if any) on all subsequent operations.

5.4.5 notify-printer-up-time (integer(1:MAX))

~~ISSUE 6 - Should we provide a way for the client to get the "printer-up-time" in a Get-Printer-Subscriptions operation, so that the client doesn't have to make two requests?~~

This REQUIRED READ-ONLY Subscription object attribute indicates the amount of time (in seconds) that the Printer implementation has been up and running. This attribute is an alias for the Printer's "printer-up-time" attribute" (see [ipp-mod] section 4.4.29), in an analogous way that the Job's "job-printer-up-time" is an alias for "printer-up-time" (see [ipp-mod] section 4.3.13.4).

Note: A client can request this attribute in a Get-Printer-Subscription-Attributes or Get-Printer-Subscriptions request and use the value returned in combination with the "notify-subscription-expiration-time" (see section 5.4.2) in order to display wall clock time equivalent to the user. The difference between this attribute and the 'integer' value of the "notify-subscription-expiration-time" attribute is the number of seconds in the future that the subscription will expire. A client can compute the wall-clock time at which the subscription will expire by adding this difference to the client's wall-clock time.

5.5 Printer Description Attributes related to Notification

This section defines the Printer Description attributes that are related to Notification.

Table 5 - Printer Description attributes associated with Notification

Printer object attributes:	Printer support	READ-ONLY, set by:
notify-schemes-supported (1setOf uriScheme)	R	Administrator/implementation
~~notify-content-types-supported (1setOf collection)~~	~~R~~	~~Administrator/implementation~~
notify-events-default (1setOf type2 keyword)	R	Administrator/implementation
notify-events-supported (1setOf type2 keyword)	R	Administrator/implementation
max-events-supported (integer(5:MAX))	R	Administrator/implementation
max-job-subscriptions-supported (integer(1:MAX))	R	Administrator/implementation
max-printer-subscriptions-supported (integer(0:MAX))	R~~C~~	Administrator/implementation
notify-lease-time-supported (rangeOfInteger(0:MAX))	R~~C~~	Administrator/implementation
notify-lease-time-default (integer(0:MAX))	R~~C~~	Administrator/implementation
persistent-jobs-supported (boolean)	O	Administrator/implementation
persistent-subscriptions-supported (boolean)	O	Administrator/implementation
printer-state-change-time (integer(1:MAX))	O	Administrator/implementation
printer-state-change-date-time (dateTime)	O	Administrator/implementation

5.5.1 notify-schemes-supported (1setOf uriScheme)

This REQUIRED Printer attribute describes the notification delivery methods supported by this Printer object. Standard values are defined in Section 5.3.1).

5.5.2 notify-events-default (1setOf type2 keyword)

This REQUIRED Printer attribute identifies the event values if the client does not supply the "notify-events" operation attribute in either a Job creation request or the Create-Printer-Subscription request. ~~All the~~ Any value~~s~~ in this attribute MUST also appear in the notify-events-supported attribute, i.e., be a supported event.

5.5.3 notify-events-supported (1setOf type2 keyword)

This REQUIRED Printer attribute identifies the events supported by this Printer object for both Per-Job and Per-Printer subscriptions which MUST be the same. ~~In order to claim conformance to this notification specification, the Printer object MUST support this Printer Description attribute.~~ Standard values are defined in Section 1.1.1.

5.5.4 max-events-supported (integer(5:MAX))

This REQUIRED Printer attribute specifies the maximum number of events that are supported in a single Per-Job or Per-Printer subscription which must be the same. A Printer MUST support at least 5 events per subscription, so that clients can depend on at least 5 events in a single subscription. If the number of events supplied by a client in a subscription exceed this number, the Printer rejects the request and returns the 'server-error-too-many-events (see section 14.3). If notification is not supported, this attribute MUST NOT be supported.

5.5.5 max-job-subscriptions-supported (integer(1:MAX))

This REQUIRED Printer attribute specifies the maximum number of Per-Job subscriptions that are supported in a single Job Creation request, i.e., the maximum number of collection values for the "job-notify" operation attribute. A value of 0 indicates no effective maximum. A Printer MUST support at least 1 Per-Job subscription. If the number of Per-Job subscriptions supplied by a client in a Job Creation request exceed this number, the Printer rejects the request and returns the 'server-error-too-many-subscriptions' (see section 14.2).

5.5.6 max-printer-subscriptions-supported (integer(0:MAX))

This REQUIRED Printer attribute specifies the maximum number of Per-Printer subscriptions that are supported by multiple Create-Printer-Subscription requests, i.e., the maximum number of un-expired Subscription objects that the Printer supports at a time. A value of 0 indicates no effective maximum. A Printer MUST support at least 1 Per-Printer subscription. If the number of Per-Printer subscriptions would exceed this number, the Printer rejects the Create-Printer-Subscription request and returns the 'server-error-too-many-subscriptions' (see section 14.2).

5.5.7 notify-lease-time-supported (rangeOfInteger(0:MAX))

This REQUIRED Printer attribute specifies the range of values in seconds that are supported for the "notify-lease-time-requested" operation attribute in a Create-Printer-Subscription or Renew-Printer-Subscription request. When the lease time expires for a Per-Printer Subscription without renewing, the Printer MUST delete the Subscription object. If the client requests a value outside this range, the Printer MUST grant a value that is in this range (see section 1.1.1). A value of 0 indicates an infinite lease, i.e., one that does not expire.

5.5.8 notify-lease-time-default (integer(0:MAX))

This REQUIRED Printer attribute specifies the value of the lease time that the Printer object has been configured to assume if the client does not supply a "notify-lease-time-requested" operation attribute in the ~~Subscribe~~Create-Printer-Subscription or Renew-Printer-Subscription ~~operations~~requests. ~~A Printer MUST support this attribute if it supports the Subscribe-Printer operation.~~

5.5.9 persistent-jobs-supported (boolean)

This OPTIONAL Printer attribute indicates whether or not the Printer supports persistent Jobs, i.e., Jobs object that are preserved across power cycles. If Jobs are persistent, then Per-Job Subscriptions MUST also be persistent, since they are part of the Job object. It is RECOMMENDED that Jobs (and Per-Job Subscriptions) be persistent.

5.5.10 persistent-subscriptions-supported (boolean)

This OPTIONAL Printer attribute indicates whether or not the Printer supports persistent Per-Printer Subscriptions, i.e., Subscription objects that are preserved across power cycles. When this value is 'true' the implementation MAY support some that are persistent and some that are not. If the value is 'false' or the attribute is not supported, Per-Printer Subscriptions MUST NOT be persistent. It is RECOMMENDED that Per-Printer subscriptions be persistent.

5.5.11 printer-state-change-time (integer(~~MIN~~1:MAX))

This OPTIONAL Printer attribute records the time, i.e., copy of the Printer's "printer-up-time" attribute, that the Printer's "printer-state" attribute was last changed. On power-up, the Printer populates the "printer-state-change-time" from its "printer-up-time" attribute, so that it always has a value.

5.5.12 printer-state-change-date-time (dateTime)

This OPTIONAL Printer attribute records the date and time, i.e., copy of the Printer's "printer-current-time" attribute, that the Printer's "printer-state" attribute was last changed. On power-up, the Printer populates the "printer-state-change-date-time" from its "printer-current-time" attribute, so that it always has a value.

6 Notification Content

This section defines the Notification content that is sent to a Notification Recipient when an event occurs. The Notification MAY be sent by the IPP Printer or a third party Notification Service (see section 2.3). There are two notification content types: Machine Consumable and Human Consumable. For most notification delivery methods both content types are defined. Some of the notification delivery methods ~~dictate~~ are defined to support only one content type. A Printer MUST support all content types defined for a notification delivery method, if it supports that notification delivery method.~~the notification content type to be used. For example, 'mailto' uses "multipart/report" and 'http' and 'ipp-tcp-notify' use "application/ipp".~~

6.1 Notification content MIME media type formats

~~Once a client's create job request, Subscribe-Job, or Subscribe-Printer is successful, the Printer adds the "notify-recipients" and "notify-events" attribute values to its subscription information on the Job, Job, or Printer object, respectively, and returns the appropriate response.~~ This section defines the Notification content that the Notification Source sends asynchronously to each Notification Recipient based on the subscription information stored with the subscription. ~~Subsequently, notifications are asynchronously sent based on the subscription information stored by the Printer. The following descriptions give more detail on how the notifications are formed for each type.~~

~~An implementation MUST NOT extend the contents of a notification.~~

Notifications are generated using the following content formats:

'application/ipp' - Machine Consumable notification content using the 'application/ipp' MIME media type [ipp-mod] using the Get-Job-Attributes response encoding for job events and Get-Printer-Attributes for printer events. The attributes listed in sections 6.2 and 6.3 are sent in an notification for Job events. The attributes listed in sections 6.2 and 6.4 are sent in an notification for Printer events. For any string in any notification, the charset and natural language rules that apply to all IPP operations apply to the notification strings as well, since they are represented as operation responses. The Unsupported Attributes Group in the response is not sent. If the values of any of the attributes sent in an notification content are not known, the value sent in the report content is the out-of-band 'unknown' value, rather than omitting the attribute (see the beginning of [ipp-mod] section 4.1).

An implementation MAY extend the contents of the Machine Consumable notification by adding additional attributes.

'text/plain; charset=utf-8' - Human Consumable notification content type [RFC2046]. The text message SHOULD include information about the attributes in sections 6.2 and 6.3 for job events or in sections 6.2 and 6.4 for printer events. This information is localized according to the information about natural language and charset in the subscription.~~If the charset to be used in the mail message is other than US-ASCII, the /charset parameter must be included in the value of this content-type header and in the notification content [RFC2046].~~

An implementation MAY extend the contents of a Human Consumable notification by adding additional information.

6.2 Notification content attributes common to Job and Printer events

This section lists the parameters and attributes that are included in both Job and Printer event Notifications. Some events do not include all of these attributes as shown in Table 6. Each notification content contains a single Job or Printer event, whether that event was subscribed using the Job Submission Subscription mechanism or the Per-Printer subscription mechanism. If either kind of subscription subscribed to both Job and Printer events, then they will be sent as separate Job notification content and Printer notification contents to the same Notification Recipient.

Table 6 - Common Job and Printer Notification content attributes

	Reference	Events	
Attributes		'job-progress'	all others
1. version-number (integer (0:32767))	mod 3.1.1	R	R
2. status-code (integer (0:32767))	mod 3.1.1	R	R
3. request-id (integer (0:MAX))	5.3.7 & mod 3.1.1	R	R
4. attributes-charset (charset)	5.3.5 & mod 3.1.4	R	R
5. attributes-natural-language (naturalLanguage)	5.3.6 & mod 3.1.4	R	R
6. printer-uri (uri)	5.4.3	R	R
7. printer-name (name(127))	mod 4.4.4		R
8. job-id (integer(1:MAX))	mod 4.3.2	R	R**
9. job-name (name(MAX))	mod 4.3.5		R**
10. trigger-event (type2 keyword)	5.3.2	R	R
11. trigger-time (integer(MIN:MAX))		R	R
12. trigger-date-time (dateTime)			O
13. subscription-id (integer(1:MAX))	5.4.1	R	R
14. subscriber-user-name (name(MAX))	5.4.4		R
15. subscriber-user-data (octetString(63))	5.3.4		R

Attribute Notes:

"status-code" - a value of 600 for a Job event and 601 for a Printer event.

"request-id" - the sequence number for this subscription, starting at 1 for each subscription.

"attributes-charset" - the value comes from the "notify-attributes-charset" in the "job-notify" Per-Job Subscription and the "notify-attributes-charset" in the Per-Printer Subscription object.

"attributes-natural-language" - the value comes from the "notify-attributes-natural-language" in the "job-notify" Per-Job Subscription and the "notify-attributes-natural-language" in the Per-Printer Subscription object.

"printer-uri" - the value comes from the "job-printer-uri" Job attribute for Per-Job subscriptions.

**"job-id" and "job-name" - included in Printer event Notifications only for Per-Job subscriptions.

"trigger-event" - the event that caused this Notification to be delivered.

"trigger-time" - the "printer-up-time" value when the event occurred.

"trigger-date-time" - the "printer-current-time" value when the event occurred - OPTIONAL to support.

"job-name" - SNMP delivery method can truncate to less than 255 octets, since the Notification needs to fit into 484 octets or so on some transports that SNMP is defined for.

"subscription-id" - for Per-Job subscriptions, the value is the index of the "job-notify" collection value, starting with 1 for the first collection value.

"subscriber-user-name" - comes from "requesting-user-name" Job attribute for Per-Job subscriptions. SNMP delivery method can truncate to less than 255 octets, since the Notification needs to fit into 484 octets or so on some transports that SNMP is defined for.

"subscriber-user-data" - opaque user data that may identify either the Subscriber and/or the ultimate Notification Recipient.

6.3 Additional Notification content attributes for Job events *only*

This section lists the additional attributes that are included only in Job event Notifications. Some events do not include all of these attributes as shown in Table 7.

Table 7 - Additional Notification content attributes for Job events *only*

		Events		
Attributes	Reference	'job-progress'	'job-completed'	all others
16. job-state (type1 enum)	mod 4.3.7		R	R
17. job-state-reasons (1setOf type2 keyword)	mod 4.3.8		R	R
18. job-k-octets-processed (integer(0:MAX))	mod 4.3.18.1	O	O	
19. job-impressions-completed (integer(0:MAX))	mod 4.3.18.2	O	O	

		Events		
Attributes	Reference	'job-progress'	'job-completed'	all others
20. job-media-sheets-completed (integer(0:MAX))	mod 4.3.18.3	O	O	
21. job-collation-type (type2 enum)	[ipp-prog]	R		
22. sheet-completed-copy-number (integer(-2:MAX))	[ipp-prog]	R		
23. sheet-completed-document-number(integer(-2:MAX))	[ipp-prog]	R		
24. impressions-interpreted (integer(-2:MAX))	[ipp-prog]	R		
25. impressions-completed-current-copy (integer(-2:MAX))	[ipp-prog]	R		

6.4 Additional Notification content attributes for Printer events *only*

Table 8 lists the additional attributes that are included only in Printer event Notifications.

Table 8 - Additional Notification content attributes for Printer events *only*

		Events
Attributes	Reference	all printer events
26. printer-state (type1 enum)	mod 4.4.11	R
27. printer-state-reasons (1setOf type2 keyword)	mod 4.4.12	R
28. printer-is-accepting-jobs (boolean)	mod 4.4.23	R

~~6.2Basic Job notification content~~

~~This section lists the parameters and attributes that are included in the Basic Job notification content. Additional job event definitions can be registered which use the Basic Job Notification content. If additional attributes are needed, additional Job Notification content formats will be defined and assigned a new status code to be used in the report in order to distinguish each report format. For example of another Job Notification content format, see [ipp-prog] for 'job-progress-events'.~~

~~Each Basic Job notification content contains a single job event, whether that event was subscribed using the Job Submission Subscription mechanism or the Explicit Subscription (Subscribe-Job or Subscribe-Printer). If either kind of subscription subscribed to both Job and Printer events, then they will be send as separate Basic Job notification content and Basic Printer notification contents to the same Notification Recipients.~~

~~If notification is supported, the implementation MUST support the following REQUIRED job object attributes, MUST support the following CONDITIONAL job object attributes, if the condition is true, and MAY support the following OPTIONAL job object attributes. Any of the following Job Description attributes that are supported MUST be included in a notification. All job notifications MUST use the Get-Job-Attributes response syntax.~~

~~The Basic Job Notification MUST include the following response parameters and job object attributes. The Job Attributes MAY be in any order:~~

~~Job object parameter/attribute~~	~~REQUIRED IN REPORT?~~	~~reference~~
~~version-number (integer (0:32767))~~	~~REQUIRED~~	~~[ipp-mod] 3.1.1~~
~~status-code (with the value: basic-job-event(600))~~	~~REQUIRED~~	~~[ipp-mod] 3.1.1~~
~~request-id (used as 32-bit sequence number - see section 8.1~~	~~REQUIRED~~	~~[ipp-mod] 3.1.1~~
~~attributes-charset (charset)~~	~~REQUIRED~~	~~[ipp-mod] 3.1.4~~
~~attributes-natural-language (naturalLanguage)~~	~~REQUIRED~~	~~[ipp-mod] 3.1.4~~
~~[subscription-id (integer(1:MAX))]~~	~~REQUIRED~~	~~section~~
~~job-printer-uri (uri)~~	~~REQUIRED~~	~~[ipp-mod] 4.3.3~~
~~job-id (integer(1:MAX))~~	~~REQUIRED~~	~~[ipp-mod] 4.3.2~~
~~job-trigger-event (type2 keyword)~~	~~REQUIRED~~	~~9.3~~
~~job-trigger-message* (text(255))~~	~~CONDITIONAL~~	~~9.4~~
~~job-trigger-time (integer(1:MAX))~~	~~REQUIRED~~	~~9.5~~
~~job-trigger-date-time (dateTime)~~	~~RECOMMENDED~~	~~9.6~~
~~job-state (type1 enum)~~	~~REQUIRED~~	~~[ipp-mod] 4.3.7~~
~~previous-job-state (type1 enum)~~	~~REQUIRED~~	~~9.7~~
~~job-state-reasons (1setOf type2 keyword)~~	~~REQUIRED~~	~~[ipp-mod] 4.3.8~~
~~job-state-reasons-added (1setOf type2 keyword)~~	~~REQUIRED~~	~~9.8~~
~~job-state-reasons-deleted (1setOf type2 keyword)~~	~~REQUIRED~~	~~9.9~~
~~subscription-id** (integer(1:MAX))~~	~~CONDITIONAL~~	~~7~~

~~Figure 8 - Basic Job Notification Content~~

~~Conditional attributes in the notification:~~

~~*If "status-message" is supported as an Operation attribute in operation responses, then "job-trigger-message" MUST be supported in the notification content.~~

~~** If the Subscribe-Job or Subscribe-Printer operation is implemented and the notification is caused by an explicit subscription request, the "subscription-id" MUST be supplied in the notification content.~~

~~If the values of any of the attributes sent in an notification content are not known, the value sent in the report content is the out-of-band 'unknown' value, rather than omitting the attribute. See [ipp-mod] section 4.1.~~

~~ISSUE 17 - Should we copy in the [ipp-prog] Job Progress notification content into this Event Notification specification?~~

~~6.3Basic Printer notification content~~

~~This section lists the parameters and attributes that are included in the Basic Printer notification content. Additional printer events can be registered which use the Basic Job Notification content. If additional attributes are needed, additional Printer Notification content formats will be registered and assigned a new status code to be used in the report in order to distinguish each report format.~~

~~Each Basic Printer notification content contains a single printer event, whether that event was subscribed using the Job Submission Subscription mechanism or the Explicit Subscription (Subscribe-Job or Subscribe-Printer). If either kind of subscription subscribed to both job and printer events, then they will be send as separate Basic Job notification content (see section 8.2) and Basic Printer notification contents to the same Notification Recipients.~~

~~If notification is supported, the implementation MUST supported the following REQUIRED attributes, MUST supported the following CONDITIONAL job object attributes, if the condition is true, and MAY supported the following OPTIONAL Printer object attributes in any printer notification. Any of the following Printer Description attributes that are supported MUST be included in an notification. All printer notifications MUST use the Get-Printer-Attributes response syntax. The Basic Printer Notification MUST include the following response parameters and Printer object attributes. The Printer Attributes MAY be in any order:~~

~~Printer object paramater/attribute~~	~~REQUIRED IN REPORT?~~	~~reference~~
~~version-number~~	~~REQUIRED~~	~~[ipp-mod] 3.1.1~~
~~status-code (with the value: basic-printer-event(601))~~	~~REQUIRED~~	~~[ipp-mod] 3.1.1~~
~~request-id (used as sequence number see section 8.1~~	~~REQUIRED~~	~~[ipp-mod] 3.1.1~~
~~printer-uri-supported (uri)~~	~~REQUIRED~~	~~[ipp-mod] 4.4.1~~
~~printer-trigger-event (type2 keyword)~~	~~REQUIRED~~	~~10.1~~
~~printer-trigger-message*(text(255))~~	~~CONDITIONAL~~	~~10.2~~
~~printer-trigger-time (integer(1:MAX))~~	~~REQUIRED~~	~~10.3~~
~~printer-trigger-date-time(dateTime)~~	~~RECOMMENDED~~	~~10.4~~
~~printer-state (type1 enum)~~	~~REQUIRED~~	~~[ipp-mod] 4.4.10~~
~~previous-printer-state (type1 enum)~~	~~REQUIRED~~	~~10.5~~
~~printer-state-reasons (1setOf type2 keyword)~~	~~REQUIRED~~	~~[ipp-mod] 4.4.11~~
~~printer-state-reasons-added (1setOf type2 keyword)~~	~~REQUIRED~~	~~10.6~~
~~printer-state-reasons-deleted (1setOf type2 keyword)~~	~~REQUIRED~~	~~10.6~~
~~subscription-id** (integer(1:MAX))~~	~~CONDITIONAL~~	~~7~~

~~Figure 10 - Basic Printer Notification Content~~

~~Conditional attributes in the notification:~~

~~* If "status-message" is supported as an Operation attribute in operation responses, then "printer-trigger-message" MUST be supported in the notification content.~~

~~** If the Subscribe-Printer operation is implemented and the notification is caused by an explicit subscription request, the "subscription-id" MUST be supplied in the notification content.~~

~~If the values of any of the attributes sent in an notification content are not known, the value sent in the report content is the out-of-band 'unknown' value, rather than omitting the attribute. See [ipp-mod] section 4.1.~~

7 Operations for Per-Job Subscriptions~~ mechanism~~

This section ~~lists~~ defines the REQUIRED operation requests and responses that ~~have some relation to notification, i.e., that~~ are related to Per-Job subscriptions. Section 8 ~~lists~~ defines the ~~OPTIONAL~~ REQUIRED operation requests and responses associated with the ~~OPTIONAL~~ REQUIRED Per-Printer subscription and its Subscription object.

7.1 Job Creation Operations (Create-Job, Print-Job, Print-URI) and Validate-Job

~~As in the current notification spec (July 22, 1999),~~ The usual method for a client to associate one subscription with a Job is to specify the subscription when the job is created. For a Per-Job Subscription, the client supplies the "job-notify (1setOf collection)" operation attribute with the ~~following notification operation~~member attributes listed in Table 9 with any of the job creation operations (Create-Job, Print-Job, Print-URI), plus Validate-Job (which doesn't create a job or subscription). If the client does not supply the~~se~~ "job-notify" attribute~~s~~ in the create operation, there is no subscription made (either implicitly or explicitly). ~~However, a client can add a subscription to a previously created Job object using the Subscribe-Job operation (see section 7.4)~~

If a Printer does not support this notification specification, then it MUST ignore the "job-notify" operation attribute and return it in the response indicated as an attribute that is not supported. See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes.

Table 9 - Member attributes of the "job-notify" collection operation attribute

Member attribute of "job-notify" collection	Reference	REQUIRED in request	Printer support
notify-recipient~~s~~ (~~1setOf~~ uri)	5.3.1	yes	REQUIRED
notify-events (1setOf type2 keyword)	5.3.2	no	REQUIRED
notify-content-type (mimeMediaType)	5.3.3	no	REQUIRED
subscriber~~notify~~-user-data (octetString(63))	5.3.4	no	REQUIRED
~~notify-exclude-event-mask (octetString(8))~~			~~REQUIRED~~
notify-attributes-charset (charset)	5.3.5	no	OPTIONAL
notify-attributes-natural-language (naturalLanguage)	5.3.6	no	OPTIONAL

See the referenced sections for a definition of these operation attributes, since they are copied to the Job object as the Job Description attributes described in section 5.3.

The following rules apply to Per-Job subscriptions:

1. Any subscription can contain job events, printer events, or both.
2. The Job Submission Subscription is only valid while the job is "not-completed". The job is "not-completed" while it is in the 'pending', 'pending-held', 'processing', and 'processing-stopped' states. The job changes from being "not-completed" to "retained" when it is done processing and enters any of the 'completed', 'canceled', or 'aborted' states. The job becomes "not-completed" again when it is restarted using the Restart-Job operation (see [ipp-mod]).

3. Since no job is created for the Validate-Job operation, the only purpose of supplying the subscription operation attributes in the Validate-Job operation is to validate that the values are supported; the Printer object does not establish a notification subscription as a result of the Validate-Job operation.

4. Since a Job Submission Subscription is included within a job submission operation, any interest in job events is limited to "this job" only (the Job object created because of this job creation operation). There is no mechanism to subscribe to events for all jobs or specifically some job other than this job in a create operation. But see the ~~Subscribe-Printer~~Create-Printer-Subscription operation (section 8.1) for an explicit operation to subscribe ~~persistently~~ for job and/or printer events independently of any particular job submission.

5. Event reporting only occurs when a notification recipient has specified a subscription to any event(s).

6. The notification implementation MAY allow an administrator to configure a policy on what events may be dropped.

7. If the OPTIONAL "notify-attributes-charset" attribute is not supported or the supplied value is not supported, the IPP Printer MUST return the attribute in the Unsupported Attributes Group but still accept the operation, as with all Job create operations. In this case, the Printer MUST use the natural language supplied in the "attributes-charset" Job creation operation attribute, if that natural language value is supported by the Printer, else the Printer object MUST use the Printer's "charset-configured" value. See the Print-Job operation in [ipp-mod].

8. If the OPTIONAL "notify-attributes-natural-language" attribute is not supported or the supplied value is not supported, the IPP Printer MUST return the attribute in the Unsupported Attributes Group but still accept the operation, as with all Job create operations. In this case, the Printer MUST use the natural language supplied in the "attributes-natural-language" Job creation operation attribute, if that natural language value is supported by the Printer, else the Printer object MUST use the Printer's "natural-language-configured" value. See the Print-Job operation in [ipp-mod].

7.2 Get-Printer-Attributes operation

The REQUIRED (by [ipp-mod]) Get-Printer-Attributes can be used to request the additional Printer Description attributes that indicate the notification supported. See section 5.5.

7.3 Get-Job-Attributes operation

The REQUIRED (by [ipp-mod]) Get-Job-Attributes can be used to request the additional Job Description attributes that relate to notification. See section 5.5.

7.4 Set-Job-~~Subscription~~ Attributes operation

The OPTIONAL Set-Job-~~Subscription~~ Attributes operation (see [ipp-set2]) allows a client to change the Job attributes of the Job's Per-Job subscription, including to add them in the first place if they had not been supplied in the Job creation operation and to remove them from the Job object. This operation replaces all the collection values of the supplied "job-notify (1setOf collection) notification attribute~~s: "notify-recipients", "notify-events", "notify-user-data", and "notify-exclude-event-mask" Job attributes~~. ~~Any notification attributes not supplied are not affected.~~

In order to cancel all Per-Job subscriptions, the client supplies the "job-notify~~-events~~" attribute with an out-of-band 'none' value (see [ipp-coll]). ~~ISSUE 11 - or should canceling a Per-Job subscription be done by supplying the out-of-band 'no-value' value for "notify-recipients" in a Set-Job-Subscription operation, instead of 'none' to "notify-events", since "notify-recipients" is the attribute that indicates that there is notification in a job creation operation? In order to remove the In order to remove the user-data (octetString(63)), the client supplies an octetString with zero length. In order to remove the "notify-exclude-event-mask (octetString(8))", the client supplies a octetString with zero length.~~

Request:

Group 1: Operation Attributes		Printer support
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	"printer-uri" (uri)	R
	"requesting-user-name" (name(MAX))	RECOMMENDED
	"job-notify" (1setOf collection)	R
	~~["notify-recipients" (1setOf uri)]~~	~~R~~
	~~["notify-events" (1setOf type2 keyword)]~~	~~R~~
	~~["notify-user-data" (octetString(63))]~~	~~R~~
	~~["notify-exclude-event-mask" (1setOf octetString(8))]~~	~~R~~
	~~["notify-natural-languages" (1setOf naturalLanguage)]~~	~~O~~

Response:

Group 1: Operation Attributes		Printer support
	"status-code" (type2 enum)	R
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	["status-message" (text(255))]	O
	["detailed-status-message" (text(MAX))]	O
Group 2: Unsupported Attributes		R

The "job-notify" operation attribute is copied to the "job-notify" Job Description attribute. See section 5.1 for a definition of the "job-notify" Job Description attribute~~these operation attributes, since they are copied to the Job object as Job Description attributes~~.

8 Per-Printer Subscriptions and the Subscription object

This section lists the REQUIRED Per-Printer Subscriptions operation requests and responses and the Subscription object that MUST be supported~~ if the OPTIONAL Per-Printer Subscriptions are supported which REQUIRES that the Subscription object be supported and vice versa~~.

ISSUE 3 - Are these abbreviated operation definitions ok for understandability and conformance, or should we use the more verbose definition style of the [ipp-mod]?

8.1 Create-Printer-Subscription operation

The REQUIRED Create-Printer-Subscription operation creates a Per-Printer subscription by creating a Per-Printer Subscription object which is added to the Printer object. The client can specify one or more job and/or printer events to be delivered as notifications to one ~~or more~~ Notification Recipient~~s~~. The job events are for any job submitted to the Printer. The printer events are any events generated by that Printer. ~~If "Explicit Subscriptions" is implemented, this operation is REQUIRED to be supported by the Printer.~~

The Printer returns a subscription id and the length of time for which it has granted a lease for the subscription. The client must renew the subscription using the Renew-Printer-Subscription operation (see section 8.4) before the lease runs out in order to maintain the subscription. A client can unsubscribe using the ~~Unsubscribe~~Cancel-Printer-Subscription operation (section 8.5) and the subscription id.

Two ~~Subscribe~~Create-Printer-Subscription operations with the same events and same Notification Recipient~~s~~ MUST be kept as distinct subscriptions and be assigned distinct subscription ids. A Printer ~~MAY suppress sending duplicate notifications to the same Notification Recipient, but the printer~~ MUST allow such duplicate subscriptions such that ~~Unsubscribe~~Cancel-Printer-Subscription doesn't unsubscribe both subscriptions and MUST send the Notifications twice to the Notification Recipient, since the "request-id" is supposed to count monotonically for each subscription.

If the Printer has a bounded set of concurrent subscriptions (for the specified events or any events), the printer rejects the operation and returns the 'server-error-too-many-subscriptions' status code. The client SHOULD try again later.

Access Rights: The authenticated user (see [IPP-MOD] section 8.3) performing this operation MUST ~~be an~~have operator or administrator access rights for ~~of~~ the Printer object (see [IPP-MOD] sections 1 and 8.5). Otherwise the IPP object MUST reject the operation and return: client-error-forbidden, client-error-not-authenticated, and client-error-not-authorized as appropriate.

```
Request:
```

Group 1: Operation Attributes		Printer support
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	"printer-uri" (uri)	R
	["requesting-user-name" (name(MAX))]	RECOMMENDED
	["notify-recipient~~s~~" (~~1setOf~~ uri)]	R
	["notify-events" (1setOf type2 keyword)]	R
	["notify-content-type" (mimeMediaType)]	R
	["subscriber~~notify~~-user-data" (octetString(63))]	R
	~~["notify-exclude-event-mask" (1setOf octetString(8))]~~	~~R~~
	["notify-attributes-charset" (charset)]	O
	["notify-attributes-natural-language~~s~~" (~~1setOf~~ naturalLanguage)]	O
	["notify-lease-time-requested" (integer(0:MAX))]	R
	["notify-persistence-requested" (boolean)]	O

```
Response:
```

Group 1: Operation Attributes		Printer Support
	"status-code" (type2 enum)	R
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	["status-message" (text(255))]	O
	["detailed-status-message" (text(MAX))]	O
	"subscription-id" (integer(1:MAX))	R
	"notify-lease-expiration-time" (integer(0:MAX))	R
	"notify-lease-time-granted" (integer(0:MAX))	R
	"notify-persistence-granted" (boolean)	R
Group 2: Unsupported Attributes		R

Attribute Notes:

"notify-recipient" (uri) - the client MUST supply this attribute in order to have a subscription.

"notify-lease-time-requested" (integer(0:MAX) - the number of seconds requested for the Per-Printer subscription lease. A value of 0 indicates a request that the Per-Printer Subscription lease never expire. Supplying a 0 value MAY require authentication in order to be used, if 0 is supported at all. If the client does not supply this attribute, the Printer uses its "notify-lease-time-default" value (see section 5.5.8).

"notify-persistence-requested" (boolean) - whether or not the Per-Printer Subscription is to be persistent, i.e., saved across power cycles. Note: Persistent trap registrations is a client option in SNMPv3 [RFC2573]. ~~ISSUE 12 - Ok to have added the OPTIONAL "notify-persistence" (boolean) operation attribute to the Create-Printer-Subscription operation?~~

"notify-lease-expiration-time" (integer(0:MAX)) - The Printer object MUST return this attribute which is the time in the future at which the subscription lease will expire, i.e., the "printer-up-time" value (in time ticks - ~~)~~ see [ipp-mod] section 4.4.29) at which the ~~lease will expire~~Printer will delete the Subscription. A value of 0 indicates that the lease subscription will never expire.

"notify-lease-time-granted" - The Printer object MUST return this attribute which is the number of seconds of time actually granted. The value of "notify-lease-time-granted" may be less than the requester requested in the "notify-lease-time-requested" if it was greater than the MAX supported or more than the requester requested if it was less than the MIN supported, as indicated in the Printer's "notify-lease-time-supported" (rangeOfInteger(0:MAX)) attribute.

"notify-persistence-granted" - whether or not this Per-Printer Subscription object instance is persistent. This attribute MUST be returned whether "notify-persistence-requested" is supported or not, so that the client knows which.

8.2 Get-Printer-Subscription-Attributes operation

The REQUIRED Get-Printer-Subscription-Attributes returns the requested attributes of the identified Per-Printer Subscription object. See sections 5.3 and 5.4.

Request:

Group 1: Operation Attributes		Printer support
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	"printer-uri" (uri)	R
	["requesting-user-name" (name(MAX))]	RECOMMENDED
	"subscription-id" (integer(1:MAX))	R
	["requested-attributes" (1setOf type2 keyword)]	R

Response:

Group 1: Operation Attributes		Printer support
	"status-code" (type2 enum)	R
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	["status-message" (text(255))]	O
	["detailed-status-message (text(MAX))]	O
Group 2: Unsupported Attributes		R
Group 3: <the requested Subscription object attributes>		R

This operation is similar to the Get-Printer-Attributes operation. If the client omits the "requested-attributes" operation attribute, the Printer MUST respond as if the client had supplied the 'all' value, i.e., return all of the attributes supported for the Subscription object.

8.3 Get-Printer-Subscriptions operation

The REQUIRED Get-Printer-Subscriptions operation returns Per-Printer subscriptions, i.e., Subscription objects.

Request:

Group 1: Operation Attributes		Printer support
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	"printer-uri" (uri)	R
	["requesting-user-name" (name(MAX))]	RECOMMENDED
	["limit" (integer(1:MAX))]	R
	["requested-attributes" (1setOf type2 keyword)]	R
	["my-subscriptions" (boolean)]	R

Response:

Group 1: Operation Attributes		Printer support
	"status-code" (type2 enum)	R
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	["status-message" (text)]	O
	["detailed-status-message" (text(MAX))]	O
Group 2: Unsupported Attributes		R
Group 3 to N:<the requested Subscription Attributes for each Subscription object in a separate group>		R

Notes: This operation is similar to the Get-Jobs operation (see [ipp-mod]). If the client wants any attributes returned, including the "subscription-id", it must include the attribute keyword name in the "requested-attributes" operation attribute. If the "requested-attributes operation attribute is omitted, the Printer MUST respond as if the client supplied the value: 'subscription-id'. If the client supplies the "my-subscriptions" with a 'false' value or omits it, the Printer returns all subscriptions, subject to the security policy in force.

8.4 Renew-Printer-Subscription operation

The REQUIRED Renew-Printer-Subscription operation permits a client to request the IPP Printer to extend the lease on a Per-Printer Subscription object instance.

Access Rights: The authenticated user (see [IPP-MOD] section 8.3) performing this operation MUST either be the owner of the Subscription object or ~~be an~~have operator or administrator access rights for ~~of~~ the Printer object (see [IPP-MOD] sections 1 and 8.5). Otherwise the IPP object MUST reject the operation and return: client-error-forbidden, client-error-not-authenticated, and client-error-not-authorized as appropriate.

```
Request:
```

Group 1: Operation Attributes		Printer support
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	"printer-uri" (uri)	R
	["requesting-user-name" (name(MAX))]	RECOMMENDED
	"subscription-id" (integer(1:MAX))	R
	["notify-lease-time-requested" (integer(0:MAX))]	R

```
Response:
```

Group 1: Operation Attributes		Printer support
	"status-code" (type2 enum)	R
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	["status-message" (text(255))]	O
	["detailed-status-message" (text(MAX))]	O
	"notify-lease-expiration-time" (integer(0:MAX))	R
	"notify-lease-time-granted" (integer(0:MAX))	R
Group 2: Unsupported Attributes		R

Attribute Notes:

"notify-lease-time-requested" (integer(0:MAX) - the number of seconds requested for the Per-Printer subscription lease. Same as Create-Printer-Subscriptions (see section 8.1).~~A value of 0 indicates a request that the Per-Printer Subscription lease never expire. Supplying a 0 value MAY require authentication in order to be used, if 0 is supported at all.~~

"notify-lease-expiration-time" - ~~The Printer object MUST return the "notify-lease-time-granted" which is the number of seconds remaining in the lease and the "notify-lease-time" which is~~ the time in the future when the subscription will expire. Same as for the Create-Printer-Subscription operation (see section 7.1).~~, i.e., the value of the Printer's "printer-up-time" (time-ticks) that the notification will be canceled by the Printer.~~

"notify-lease-time-granted" - the lease time (number of seconds) actually granted. Same as for the Create-Printer-Subscription operation (see section 7.1).

Note: There is no way to change any of the Subscription object attributes, except the "notify-lease-expiration-time" attribute. In order to change other attributes, a client can create a new Subscription object and then use the Cancel-Printer-Subscription operation to cancel the old one (or do this in the other order, in case there is a limit on the number of Subscription object instances, as long as a short window with no Notifications is ok).

~~ISSUE 13 - What happens if the client supplies a 0 for the "notify-lease-time-requested" operation attribute? Does that result in a canceling of the Subscription? Is that ok? Ok to keep Cancel-Printer-Subscription operation too?~~

~~ISSUE 14 - Ok that Renew-Printer-Subscription doesn't allow any modification of the Subscription object attributes, except the "notify-lease-time" attribute? If we need a way to change other attributes would we add it to this same operation and rename it to Set-Printer-Subscription? Having a Set-Printer-Subscription operation would allow a client/server to set the "notify-events" to 'none' and thereby reserve a slot using the lease time, but not have any Notifications until a subsequence Set-Printer-Subscription operation was done that set some events.~~

Note: There is no need to renew a Per-Job Subscription, since it is effectively the time that the Job is active (see section 7.1)~~has document~~.

8.5 Cancel-Printer-Subscription operation

The REQUIRED Cancel-Printer-Subscription operation allows a client to remove a Per-Printer Subscription object from the Printer. No more Notifications are delivered for that Subscription. Once performed, there is no way to use that Subscription in the future. Subscription-ids should not be reused immediately, so that a stale reference situation is not created. Same as for Cancel-Job and job-ids.

Access Rights: The authenticated user (see [IPP-MOD] section 8.3) performing this operation MUST either be the owner of the Subscription object or ~~be an~~have operator or administrator access rights for ~~of~~ the Printer object (see [IPP-MOD] sections 1 and 8.5). Otherwise the IPP object MUST reject the operation and return: client-error-forbidden, client-error-not-authenticated, and client-error-not-authorized as appropriate.

```
Request:
```

Group 1: Operation Attributes		Printer support
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	"printer-uri" (uri)	R
	["requesting-user-name" (name(MAX))]	RECOMMENDED
	"subscription-id" (integer(1:MAX))	R

```
Response:
```

Group 1: Operation Attributes		Printer support
	"status-code" (type2 enum)	R
	"attributes-charset" (charset)	R
	"attributes-natural-language" (naturalLanguage)	R
	["status-message" (text(255))]	O
	["detailed-status-message" (text(MAX))]	O
Group 2: Unsupported Attributes		R

9 Comparison of Per-Job Submission Subscriptions versus ~~Explicit~~ Per-Printer Subscriptions

Per-Job and Per-Printer subscriptions are quite similar. However, there are some semantic differences between Per-Job ~~submission~~ subscriptions and ~~explicit~~ Per-Printer subscriptions. A ~~"~~Per-Job Submission Subscription~~"~~ is established by the client when submitting a job~~between the client and the device when a job create request is made~~. ~~The attributes as specified in this document are sent to the device and the response includes attributes telling the client if the subscription was successful or not.~~ An Per-Printer ~~"Explicit~~ Subscription~~"~~ is established between a client and a ~~device~~ Printer using the ~~four~~ new Create-Printer-Subscription operation~~s: Subscribe-Job, Subscribe-Printer, Unsubscribe, and Renew-Subscription~~. Some specific differences are:

1. Either type of subscription can ~~register~~ subscribe to ~~for~~ Job events, Printer events, or both.

2. For ~~"~~Per-Job ~~Submission~~ Subscriptions~~" and the Subscribe-Job operation~~, the subscription is only valid while the job is "not-complete" (see sections 1 and 1).

3. For the Per-Printer Subscriptions~~Subscribe-Printer operation~~, the subscription is valid until it is explicitly canceled with ~~an Unsubscribe~~a Cancel-Printer-Subscription operation or the time (in seconds) that the Printer returned ~~was specified~~ in the "notify-lease-time-granted" operation attribute expires, whichever occurs first.
4. Job Events in a "Per-Job ~~Submission~~ Subscription~~"~~ ~~and Subscribe-Job operation~~ apply only to "one job" (the Job created by the job creation ~~specified within the create-job or subscribe-job~~ operation).
5. Job Events in a Per-Printer-Subscription ~~operation~~ apply to ALL jobs contained in the IPP Printer object.
6. Printer Events in ~~all~~both kinds of subscriptions apply no matter which job is being processed, if any.

~~10Job Submission Subscription~~

~~"Job Submission Subscriptions" are specified by the end-user when submitting a job to the Printer using two new subscription operation attributes described in this section. A client subscribes to events by supplying these attributes in any create request (i.e., a Print-Job Request, Print-URI Request, Validate-Job Request, or a Create-Job Request). These attributes are multi-valued attributes; the client can supply more than one value. If the client does not supply these attributes in the create operation, there is no subscription made (either implicitly or explicitly). However, a client can add a subscription to a previously created Job object using the Subscribe-Job operation (see section 6.1).~~

~~The following rules apply:~~

~~1.Any subscription can contain job events, printer events, or both.~~

~~2.The Job Submission Subscription is only valid while the job is "not-completed". The job is "not-completed" while it is in the 'pending', 'pending-held', 'processing', and 'processing-stopped' states. The job changes from being "not-completed" to "retained" when it is done processing and enters any of the 'completed', 'canceled', or 'aborted' states. The job becomes "not-completed" again when it is restarted using the Restart-Job operation (see [ipp-mod]). Since no job is created for the Validate-Job operation, the only purpose of supplying the subscription operation attributes in the Validate-Job operation is to validate that the values are supported; the Printer object does not establish a notification subscription as a result of the Validate-Job operation.~~

~~). The UDP datagram contains the "application/ipp" notification content format (see Section). The notification recipient does not acknowledge receipt of the notification.~~

~~If the client specifies a "notify-recipients" URI scheme that is not supported by the printer, the Printer MUST return the [new] 'client-error-notify-uri-scheme-not-supported' in the IPP response in reply to the create request.~~

~~ISSUE 5 - Which URL parameters should we mention (which like SLP) are removed before being used?~~

~~7.1.2notify-events (1setOf type2 keyword)~~

~~The client OPTIONALLY supplies this operation attribute in a create request. In order to claim conformance to this notification specification, the Printer object MUST support this attribute. This attribute identifies the job and/or printer events that are to be delivered to the Notification Recipients as notifications. If the client does not supply this attribute in a create request, but does supply the "notify-recipients", the Printer object uses its "notify-events-default" attribute value (see section).~~

~~There are both job events and printer events. Each job and printer event is assigned a keyword to use in this attribute and in the notification.~~

~~Standard event keyword values are:~~

> ~~See the values of the "job-trigger-events" Job Description attribute and the "printer-trigger-events" Printer Description attribute.~~

~~ISSUE 6 - Should we add a third operation attribute to Job Submission Subscriptions, Subscribe-Job, and Subscript-Printer that is an opaque octet-string that is passed to the Notification Recipient on every notification, as in Jini? This operation attribute would be OPTIONAL for the client to supply and the Printer to support.~~

~~11Explicit Subscriptions~~

~~"Explicit Subscriptions" are created by performing explicit subscription operations on the Printer or an identified Job object. These operations are: Subscribe-Job, Subscribe-Printer, Renew-Subscription, and Unsubscribe. These operations are like any other IPP operation with respect to common operation attributes (target, charset, "requesting-user-name", natural language, unsupported attributes, etc.).~~

~~If a printer crashes or is rebooted, an attempt should be made for the Printer object to maintain all of the subscriptions currently active.~~

~~ISSUE 7 - Rather than adding the Subscribe-Job operation (and possibly corresponding Renew-Job-Subscription and Unsubscribe-Job-Subscription operations), why not go back to the original proposal that has a 1setOf collection for the create job operations. The collection contains the two simple "notify-recipients" (1setOf uri) and "notify-events (1setOf type2 keyword). We could say that only one collection value is sufficient for conformance to help the low end. Supporting additional values would give the same functionality as the Subscribe-Job operation, but would happen when the job was submitted.~~

~~8.1Subscribe-Job Operation~~

~~This OPTIONAL operation allows a client to add an explicit notification subscription to a Job object after the job has been created, in case the client did not supply a Job Submission Subscription when the job was submitted. Also this operation allows a client to set up multiple Job Subscriptions with different sets of events for different Notification Recipients. With the Job Submission Subscription (section 5) all Notification Recipients receive the same set of events. If this operation is supported, then Job Submission Subscription MUST be supported, as well as the other Explicit Subscription operations (Subscribe-Printer, Renew-Subscription, and Unsubscribe).~~

~~The Printer returns a subscription id so that a client can unsubscribe to the explicit subscription using the Unsubscribe operation (section 6.4) before the job completes. The Printer does not return a lease time, since the subscription lasts until the job completes.~~

~~Two Subscribe-Job operations with the same events and same Notification Recipients MUST be kept as distinct subscriptions and be assigned distinct subscription ids. A Printer MAY suppress sending duplicate notifications to the same Notification Recipient, but the printer MUST allow duplicate subscriptions such that Unsubscribe doesn't unsubscribe both subscriptions.~~

~~The subscription expires after the job completes (enters the 'completed', 'canceled', or 'aborted' job states) just as with the Job Submission Subscription. See section 5 for more detail.~~

~~*Access Rights:* The authenticated user (see [ipp-mod] section 8.3) performing this operation must either be the job owner of the current job or an operator or administrator of the Printer object (see [ipp-mod] sections 1 and 8.5). Otherwise, the IPP object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.~~

~~11.1.1Subscribe-Job Request~~

~~The following groups of attributes are part of the Subscribe Request:~~

~~Group 1: Operation Attributes~~

~~Natural Language and Character Set:~~

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

~~Target:~~

~~Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation attribute(s) which define the target for this operation as described in [ipp-mod] section 3.1.5.~~

~~Requesting User Name:~~

~~The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as described in [ipp-mod] section 8.3.~~

~~"notify-recipients" (1setOf uri):~~

~~The client MUST supply this attribute. The Printer object MUST support this attribute, if it supports this operation. This attribute describes both where (the address of the Notification Recipients) and how (the delivery method) notifications are to be delivered as notifications when any of the events specified in the "notify-events" attribute occur. See section 5.1.1.~~

~~"notify-events" (1setOf type2 keyword):~~

~~The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute, if it supports this operation. This attribute identifies the job and/or printer events that are to be delivered to the Notification Recipients as notifications (see section 5.1.2). If the client does not supply this attribute, then the Printer object uses its "notify-events-default" attribute value (see section 9.9).~~

~~8.1.2Subscribe-Job Response~~

~~The following groups of attributes are part of the Subscribe-Job Response:~~

~~Group 1: Operation Attributes~~

~~Status Message:~~

~~In addition to the REQUIRED status code returned in every response, the response OPTIONALLY includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation attribute as described in [ipp-mod] sections 13 and 3.1.6.~~

~~Natural Language and Character Set:~~

~~The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod] section 3.1.4.2.~~

~~"subscription-id" (integer(1:MAX)):~~

~~The Printer object MUST support this attribute and return it in the response. It contains an integer subscription identifier assigned by the Printer object for use in notifications and Unsubscribe operations. There is no need to renew, since the lease is the same as for Job Submission operations, i.e., until the Job is completed.~~

~~Group 2: Unsupported Attributes~~

~~See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes.~~

~~8.2Subscribe-Printer Operation~~

~~This OPTIONAL operation allows a client to add an explicit notification subscription to a Printer object at any time, whether or not the client has submitted a job to that Printer. The client can specify one or more job and/or printer events to be delivered as notifications to one or more Notification Recipients. The job events are for any job submitted to the Printer. The printer events are any events generated by that Printer. If "Explicit Subscriptions" is implemented, this operation is REQUIRED to be supported by the Printer.~~

~~The Printer returns a subscription id and the length of time for which it has granted a lease for the subscription. The client must renew the subscription using the Renew-Subscription operation (see section 6.3) before the lease runs out in order to maintain the subscription. A client can unsubscribe using the Unsubscribe operation (section 6.4) and the subscription id.~~

~~Two Subscribe-Printer operations with the same events and same Notification Recipients MUST be kept as distinct subscriptions and be assigned distinct subscription ids. A Printer MAY suppress sending duplicate notifications to the same Notification Recipient, but the printer MUST allow duplicate subscriptions such that Unsubscribe doesn't unsubscribe both subscriptions.~~

~~If the Printer has a bounded set of concurrent subscriptions (for the specified events or any events), the printer rejects the operation and returns the 'server-error-too-many-subscriptions' status code. The client SHOULD try again later.~~

~~Access Rights: The authenticated user (see [IPP-MOD] section 8.3) performing this operation must be an operator or administrator of the printer object (see [IPP-MOD] sections 1 and 8.5). Otherwise the IPP~~

~~object must reject the operation and return: client-error-forbidden, client-error-not-authenticated, and client-error-not-authorized as appropriate.~~

~~11.2.1Subscribe-Printer Request~~

~~The following groups of attributes are part of the Subscribe-Printer Request:~~

~~Group 1: Operation Attributes~~

~~Natural Language and Character Set:~~

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

~~Target:~~

~~The "printer-uri" (uri) operation attribute which is the target for this operation as described in [ipp-mod] section 3.1.5.~~

~~Requesting User Name:~~

~~The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as described in [ipp-mod] section 8.3.~~

~~"notify-recipients" (1setOf uri):~~

~~The client MUST supply this attribute. The Printer object MUST support this attribute, if is supports this operation. This attribute describes both where (the address of the Notification Recipients) and how (the delivery method) notifications are to be delivered as notifications when any of the events specified in the "notify-events" attribute occur. See section 5.1.1.~~

~~"notify-events" (1setOf type2 keyword):~~

~~The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute, if it supports this operation. This attribute identifies the job and/or printer events that are to be delivered to the Notification Recipients as notifications (see section 5.1.2). If the client does not supply the attribute, then the Printer object uses its "notify-events-default" attribute value (see section 9.9).~~

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

~~The client OPTIONALLY supplies this attribute. The printer object MUST support this attribute, if it supports this operation. This attribute requests a lease time for the subscription, i.e., the number of seconds before the subscription expires.~~

~~There is no way to query the outstanding subscriptions. Instead, every subscription has a lease time, so that no subscription can be carried too long and become a dead subscription because the Notification Recipient has gone away.~~

~~If the client does not supply this attribute in the request, the Printer object uses its "notify-lease-time-default" attribute value (see section 9.12).~~

~~"printer-message-from-operator":~~

~~If the client supplies this attribute, the Printer copies its value to the Printer's "printer-message-from-operator" Printer Description attribute.~~

~~8.2.2Subscribe-Printer Response~~

~~The following groups of attributes are part of the Subscribe-Printer Response:~~

~~Group 1: Operation Attributes~~

~~Status Message:~~

~~In addition to the REQUIRED status code returned in every response, the response OPTIONALLY includes a "status-message" (text) operation attribute as described in [ipp-mod] section 3.1.6.~~

~~Natural Language and Character Set:~~

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

~~"subscription-id" (integer(1:MAX)):~~

~~The Printer object MUST support this attribute and return it in the response. It contains an integer subscription identifier assigned by the Printer object for use in notifications and Renew-Subscription/Unsubscribe operations.~~

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

~~The Printer object MUST support this attribute and return it in the response. It contains the actual lease time granted, which MUST be in the range specified by the Printer's "notify-lease-time-supported" attribute. If the client supplied a value that is less than the minimum or greater than the maximum, the Printer sets the actual lease time to the minimum or maximum value, respectively, but otherwise accepts and performs the operation normally and no special status code is returned. It is up to the client to check this returned value, if it cares.~~

~~"initial-sequence-numbers" (1setOf integer(1:MAX)):~~

~~The Printer object MUST support this attribute and return it in the response. It contains the sequence number values that will be sent in the first notification for each of the events in the "notify-events" operation attribute in the request. The order of the values of this attribute correspond to the order of the values in the "notify-events" operation attribute. Since the event sequence numbers are associated with the event, not the subscription, there is a single sequence number attribute for the Printer for each event, which is counting monotonically for each event separately . Therefore, a Notification Recipient that is concerned about missing notifications, needs to know the first value to expect.~~

~~Note: The subscribing client needs to convey this value to the Notification Recipient, in case the Notification Recipient cares.~~

~~ISSUE 8 - Should the event sequence number be associated with the event (see section 7.1) as in Jini, or with the Subscription object? If the latter, then notification batching could happen for the same subscription. Then "xxx-trigger-event" could be changed back to 1setOf to agree with the~~

~~"notify-events" operation, Job Description, and Printer Description attributes. Then event sequence numbers always start with 1 for Explicit Printer Subscriptions, just like for Explicit Job and Job Submission Subscriptions and there is no need to return the starting sequence number for Printer events.~~

~~"detailed-status-message" (text(MAX))~~

~~The OPTIONAL "detailed-status-message" operation attribute provides additional more detailed technical and implementation-specific information about the operation. The "detailed-status-message" attribute's syntax is "text(MAX)". See [ipp-mod] section 3.1.6.3 for more information.~~

~~Group 2: Unsupported Attributes~~

~~See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes.~~

~~8.3 Renew-Subscription~~

~~This OPTIONAL operation allows a client to renew the lease time on an already established explicit subscription. The client supplies the "subscription-id" for this operation that was returned in the previous Subscribe-Job or Subscribe-Printer operation that created the subscription. If "Explicit Subscriptions" is implemented, this operation is REQUIRED to be supported by the Printer.~~

~~The Subscription-id MUST NOT change upon renewal of the lease.~~

~~11.3.1Renew-Subscription Request~~

~~The following groups of attributes are part of the Subscribe Request:~~

~~Group 1: Operation Attributes~~

~~Natural Language and Character Set:~~

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

~~Target:~~

~~The "printer-uri" (uri) operation attribute which is the target for this operation as described in [ipp-mod] section 3.1.5.~~

~~ISSUE 9 – A "printer-uri" plus "subscription-id" isn't enough to target an Explicit Job Subscription? Do we really need both Renew-Printer-Subscription and Renew-Job-Subscription, both which require a "subscription-id", but the former has a Printer object target and the latter has a Job object target?~~

~~Requesting User Name:~~

~~The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as described in [ipp-mod] section 8.3.~~

~~"subscription-id" (integer(1:MAX)):~~

~~The client MUST supply this attribute. The printer object MUST support this attribute. It identifies the subscription to be renewed, i.e., it is the integer subscription identifier returned by the Printer object in a previous Subscribe or Renew-Subscription operation.~~

~~The Printer returns a 'client-error-not-found' status code, if the subscription id is not found (or has expired).~~

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

~~See the description of "notify-lease-time" for the Subscribe operation request in Section .6.2.1.~~

~~8.3.2Renew-Subscription Response~~

~~The following groups of attributes are part of the Subscribe Response:~~

~~Group 1: Operation Attributes~~

~~Status Message:~~

~~In addition to the REQUIRED status code returned in every response, the response OPTIONALLY includes a "status-message" (text) operation attribute as described in [ipp-mod] section 3.1.6.~~

~~Natural Language and Character Set:~~

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

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

~~See the description of "notify-least-time" for the Subscribe operation response in Section 6.2.2.~~

~~Group 2: Unsupported Attributes~~

~~See section 3.1.7 for details on returning Unsupported Attributes.~~

~~*Access Rights*: The authenticated user (see [IPP-MOD] section 8.3) performing this operation must be an operator or administrator of the printer object (see [IPP-MOD] sections 1 and 8.5). Otherwise the IPP object must reject the operation and return: client-error-forbidden, client-error-not-authenticated, and client-error-not-authorized as appropriate.~~

~~ISSUE 10 - Can't we have authentication for subscriptions as we have for jobs? Then the owner of the subscription, i.e., the user that performed the Explicit Subscription can Renew or Unsubscribe.~~

~~8.4Unsubscribe~~

~~This OPTIONAL operation allows a client to unsubscribe explicit notification subscription. The "subscription-id" is required to perform this operation. Once a client performs an Unsubscribe operation, the Printer (or an associated notification service) immediately stops sending notifications to the associated Notification Recipient(s) for that subscription. If "Explicit Subscriptions" is implemented, this operation is REQUIRED to be supported by the Printer.~~

~~ISSUE 11 - Add way to the Unsubscribe operation to unsubscribe all subscriptions? all "my" subscriptions?~~

~~*Access Rights*: The authenticated user (see [IPP-MOD] section 8.3) performing this operation must be an operator or administrator of the printer object (see [IPP-MOD] sections 1 and 8.5). Otherwise the IPP object must reject the operation and return: client-error-forbidden, client-error-not-authenticated, and client-error-not-authorized as appropriate.~~

~~ISSUE 12 - Can't we have authentication for subscriptions as we have for jobs? Then the owner of the subscription, i.e., the user that performed the Explicit Subscription can Renew or Unsubscribe. Or does the lease and sequence number concepts mean that only operators need to be able to Unsubscribe?~~

~~11.4.1Unsubscribe Request~~

~~The following groups of attributes are part of the Unsubscribe Request:~~

~~Group 1: Operation Attributes~~

~~Natural Language and Character Set:~~

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

~~Target:~~

~~The "printer-uri" (uri) operation attribute which is the target for this operation as described in [ipp-mod] section 3.1.5.~~

~~ISSUE 13 - A "printer-uri" plus "subscription-id" isn't enough to target an Explicit Job Subscription? Do we really need both Unsubscribe-Printer and Unsubscribe-Job, both which require a "subscription-id", but the former has a Printer object target and the latter has a Job object target?~~

~~Requesting User Name:~~

~~The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as described in [ipp-mod] section 8.3.~~

~~"subscription-id" (integer(1:MAX)):~~

~~The client MUST supply this attribute. The printer object MUST support this attribute. It identifies the subscription to be removed, i.e., it is the integer subscription identifier returned by the Printer object in a previous Subscribe or Renew-Subscription operation.~~

~~The Printer returns a 'client-error-not-found' status code, if the subscription id is not found (or has expired).~~

~~8.4.2Unsubscribe Response~~

~~The following groups of attributes are part of the Unsubscribe Response:~~

~~Group 1: Operation Attributes~~

~~Status Message:~~

~~In addition to the REQUIRED status code returned in every response, the response OPTIONALLY includes a "status-message" (text) operation attribute as described in [ipp-mod] section 3.1.6.~~

~~Natural Language and Character Set:~~

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

~~Group 2: Unsupported Attributes~~

~~See section 3.1.7 for details on returning Unsupported Attributes.~~

~~ISSUE 14 - Do we need a Get-Printer-Subscriptions operation that returns a list of explicit subscriptions to the Printer object? Jini has such an operation. This operation could be used by a client that has forgotten the subscription id of a long standing subscription. On the other hand, the lease concept was introduced to eliminate the need for a Get-Printer-Subscriptions operation. Also the sequence number mechanism also helps with filtering out duplicate subscriptions that send the same event more than once, since the same sequence number is send in each. If we do add a Get-Printer-Subscriptions operation, then probably need to include a "which-subscriptions" operation attribute with values: 'my-subscriptions' and 'all-subscriptions', just like Get-Jobs?~~

~~10Job Description Attributes~~

~~The following Job Description attributes are defined for use with notification:~~

~~10.1notify-recipients (1setOf uri)~~

~~This REQUIRED READ-ONLY attribute describes both where (the address) and how (the delivery method) notifications are to be delivered when any of the events specified in the "notify-events" attribute occur. The Printer object MUST populate this Job Description attribute from the corresponding Operation attribute supplied by the client in t for more description of this attribute.~~

~~Note: There is no way to query the recipients that have been subscribed using the Subscribe-Job operation.~~

~~ISSUE 18 - How many individuals, recipients can be established to be notified of a particular event? The number of recipients should have a minimum number required for support. We did not agree on a such a number. This needs more discussion. If the maximum number required is 1, then clients might not bother supporting more than one recipient in a subscription.~~

~~10.2notify-events (1setOf type2 keyword)~~

~~for the behavior if the client-supplied value is outside this range.~~

~~11.12notify-lease-time-default (integer(0:MAX))~~

~~This Printer attribute specifies the value of the lease time that the Printer object has been configured to assume if the client does not supply a "notify-lease-time" operation attribute in the Subscribe-Printer or Renew-Subscription operations. A Printer MUST support this attribute if it supports the Subscribe-Printer operation.~~

~~11.13number-of-subscriptions-supported (integer(1:MAX))~~

~~This OPTIONAL attribute identifies the number of explicit subscriptions using the Subscribe-Printer operation that this Printer object can handle at once. The Printer object MUST support this Printer Description attribute, if is supports the Subscribe-Printer operation.~~

~~NOTE: An event for subscription expiration is not defined. It is the client's responsibility to set up a renew timer. Such a timer will not depend on the reliability of notification delivery, and so will be more reliable.~~

10 Conformance Requirements

This section further enhances the Conformance Requirements detailed in [IPP-MOD] section 5. Extensions made to the events herein must be made such that new events or event attributes are backward compatible to clients who implemented early versions of this notification specification.

It is OPTIONAL to implement this Event Notification specification. If implemented, IPP objects MUST support all of the REQUIRED object attributes as defined in this document in the indicated sections.

~~If an object supports an attribute, it MUST support only those values specified in this document or through the extension mechanism described in [IPP-MOD] section 5.2.4.~~

If IPP Notification is implemented, ~~it is OPTIONAL to support the "Explicit Subscriptions" operations. If "Explicit Subscriptions" is implemented,~~ the operations described in this document must be supported as described in Table 10:

Table 10 - Conformance Requirements for Operations

Attribute	Conformance requirements
"job-notify" in Job Creation operations (section 1)	REQUIRED
Set-Job-Attributes (see [ipp-set2])	OPTIONAL
~~Subscribe-Job (section 6.1)~~	~~OPTIONAL~~
~~Subscribe~~Create-Printer-Subscription (section 8.1)	REQUIRED
Get-Printer-Subscription-Attributes (section 8.2)	REQUIRED
Get-Printer-Subscriptions (section 8.3)	REQUIRED
Renew-Printer-Subscription (section 8.4)	REQUIRED
~~Unsubscribe~~Cancel-Printer-Subscription (section 8.5)	REQUIRED

~~ISSUE 26 - Can there be registered event extensions as well as private event extensions?~~

11 IANA Considerations

IANA will be called on to register URL schemes for notification delivery for use in the "notification-recipient" attribute, using the same procedures outlined in [ipp-mod].

12 Internationalization Considerations

This IPP notification specification continues the internationalization of [ipp-mod] for attributes containing text strings and names. A subscribing client can specify a different natural language and charset for each Notification content delivered to a Notification Recipient.

The Human Consumable Notification content is a 'text/plain; charset=utf-8' by default where the Notification Sender has localized the text message as requested by the subscriber for the intended Notification Recipient.

13 Security Considerations

By far the biggest security concern is the abuse of notification: sending unwanted notifications to third parties (i.e., spam). The problem is made worse by notification addresses that may be redistributed to multiple parties (e.g. mailing lists). There exist scenarios where third party notification is required (see Scenario #2 and #3 in [ipp-not-req]). The fully secure solution would require active agreement of all recipients before sending out anything. However, requirement #9 in [ipp-req] ("There is no requirement for IPP Printer receiving the print request to validate the identity of an event recipient") argues against this. Certain systems may decide to disallow third party notifications (a traditional fax model).

Clients submitting notification requests to the IPP Printer has the same security issues as submitting an IPP/1.1 print job request. The same mechanisms used by IPP/1.1 can therefore be used by the client notification submission. Operations that require authentication can use the HTTP authentication. Operations that require privacy can use the HTTP/TLS privacy.

The notification access control model should be similar to the IPP access control model for Jobs. Creating a Per-Printer Notification Subscription object is associated with a user. Only the creator or an operator can cancel the subscription. The system may limit the listing of items to only those items owned by the user. Some subscriptions (e.g. those that have a lifetime longer than a job) can be done only by privileged users (users having operator and/or administrator access rights), if that is the authorization policy.

The standard security concerns (delivery to the right user, privacy of content, tamper proof content) apply to the notification delivery. IPP should use the security mechanism of the delivery method used. Some delivery mechanisms are more secure than others. Therefore, sensitive notifications should use the delivery method that has the strongest security.

~~ISSUE 27 - Waiting for security requirements in the IPP/1.1 Notification Requirements, such as authorization and authentication for Renew-Subscription and Unsubscribe.~~

14 Status Codes

~~Add~~ The following status codes are defined as extensions for notification:

14.1 client-error-uri-notification-scheme-not-supported (0x04??)

The scheme of the client-supplied URI in a "notify-recipient~~s~~" operation attribute in a Create-Printer-Subscription ~~create job, Subscribe-Job, or Subscribe-Printer~~ operation is not supported. See [ipp-mod] section 3.1.7.

Note: There is no corresponding Per-Job subscription error, since the Printer object MUST ignore any errors in the "job-notify" operation attribute and perform the Job Creation operation (see section 5.1), since the job can still be printed.

ISSUE 4 - The client-error-uri-scheme-not-supported is already an IPP/1.1 status code and section 3.1.6.1 indicates that the "document-uri" attribute NEED NOT be returned in the Unsupported Attributes group, otherwise, we could use that error code and REQUIRE that offending "xxx-uri" attribute(s) to be returned.

14.2 server-error-too-many-subscriptions (0x04??)

The bounded set of concurrent Per-Printer subscriptions supported by~~on~~ the Printer object would be exceeded if this request were accepted.

Note: There is no corresponding Per-Job subscription error, since the Printer object MUST ignore any errors in the "job-notify" operation attribute and perform the Job Creation operation (see section 5.1), since the job can still be printed.

14.3 server-error-too-many-events (0x04??)

The client supplied more events in the "notify-events" operation attribute in a Create-Printer-Subscription that the Printer supports, as indicated in its "max-events-supported" attribute (see section 5.5.4).

Note: There is no corresponding Per-Job subscription error, since the Printer object MUST ignore any errors in the "job-notify" operation attribute and perform the Job Creation operation (see section 5.1), since the job can still be printed.

ISSUE 5 - Should the entire "job-notify" operation attribute be ignored, if one of its member attributes has a problem, or should just that member attribute be ignored? If the latter, then is just the member attribute returned in the Unsupported Attributes group?

15 References

~~[draft-prtmib]~~
~~Turner, R., "Printer MIB", <draft-ietf-printmib-mib-info-04.txt>, work in progress, January 22, 1999.~~

[ipp-coll]
deBry, R., , Hastings, T., Herriot, R., "Internet Printing Protocol/1.0 & 1.1: collection attribute syntax", <draft-ietf-ipp-collection.doc>, work in progress, August 22, 1999.

[ipp-mod]
deBry, R., , Hastings, T., Herriot, R., Isaacson, S., Powell, P., "Internet Printing Protocol/1.1: Model and Semantics", < draft-ietf-ipp-model-v11-04.txt>, work in progress, June 23, 1999.

[ipp-not-hist]
deBry, R., Lewis, H., Hastings, T., "Internet Printing Protocol/1.1: Requirements for IPP Notifications", <draft-ietf-ipp-not-change-history-00.doc>, work in progress, August 22, 1999.

[ipp-not-req]
deBry, R., Lewis, H., Hastings, T., "Internet Printing Protocol/1.1: Requirements for IPP Notifications", <draft-ietf-ipp-not-03.txt>, work in progress, August 11, 1999.

[ipp-set2~~adm~~]
Kugler, C., , Hastings, T., "Internet Printing Protocol/1.0 & 1.1: Additional ~~Administrative~~ Operations, Set 2", <draft-ietf-ipp-~~ops-~~set2~~admin-990719~~.txt>, work in progress, ~~July 19~~August 22, 1999.

[ipp-prog]
Hastings, T., Bergman, R., Lewis, H., "Proposed Job Progress Attributes for IPP", <draft-ietf-ipp-~~job-~~prog~~-attr-990518~~.txt> work in progress, May 18, 1999.

~~[RFC1759]~~
~~Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759, March 1995.~~

~~[RFC1892]~~
~~Vaudreuil, G., "The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages, RFC 1892, January 1996.~~

[RFC2046]
Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed & N. Borenstein. November 1996. (Obsoletes RFC1521, RFC1522, RFC1590), RFC 2046.

[RFC2119]
S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March 1997

[RFC2566]
deBry, R., , Hastings, T., Herriot, R., Isaacson, S., Powell, P., "Internet Printing Protocol/1.0: Model and Semantics", RFC 2566, April 1999.

~~[sense]~~
~~Martin, J. et all., "System Event Notification System Environment (SENSE)", ftp://ftp.pwg.org/pub/pwg/sense/, work in progress, Spring 1996.~~

16 Author's Addresses

Scott A. Isaacson (Editor)
Novell, Inc.
122 E 1700 S
Provo, UT 84606

Phone: 801-861-7366
Fax: 801-861-2517
e-mail: sisaacson@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

Roger deBry
Utah Valley State College
Orem, UT 84058

Phone: (801) 222-8000
EMail: debryro@uvsc.edu

Jay Martin
e-mail: jkm@underscore.com

Michael Shepherd
Xerox Corporation
800 Phillips Road MS 128-51E
Webster, NY 14450

Phone: 716-422-2338
Fax: 716-265-8871
e-mail: mshepherd@crt.xerox.com

Ron Bergman (Editor)
Dataproducts Corp.
1757 Tapo Canyon Road
Simi Valley, CA 93063-3394

Phone: 805-578-4421
Fax: 805-578-4001
Email: rbergman@dpc.com

~~17Appendix A: Registration Forms to be filled out and submitted to IANA~~

~~16.2Registration of ipp-udp-notify scheme for use with IPP~~

~~This appendix contains the information that IANA requires for registering a URL scheme for use with the "application/ipp" MIME media type. The information following this paragraph will be forwarded to IANA to register 'ipp-udp-notify' whose contents are defined in Section 5.1.1 "notify-recipients (1setOf uri)" in this document:~~

~~TBD~~

~~Required parameters:~~ ~~none~~

~~Optional parameters:~~ ~~none~~

~~Encoding considerations:~~

~~Security considerations:~~

~~IPP/1.0 protocol requests/responses do not introduce any security risks not already inherent in the underlying transport protocols. Protocol mixed-version interworking rules in [ipp-mod] as well as protocol encoding rules in [ipp-pro] are complete and unambiguous.~~

~~Interoperability considerations:~~

~~TBD~~

~~Published specification:~~

~~[ipp-not] Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., "Internet Printing Protocol/1.0 & 1.1: IPP Event Notification" draft-ietf-ipp-not-02.txt, May, 1999.~~

~~Applications which use this URL scheme:~~

~~TBD~~

~~Person & email address to contact for further information:~~

~~Thomas N. Hastings~~
~~Xerox Corporation~~
~~737 Hawaii St.~~
~~El Segundo, CA 90245~~

~~Phone: (310) 333-6413~~

~~Fax: (310) 333-5514~~

~~Email: hastings@cp10.es.xerox.com~~

~~16.3Registration of multipart/report type, 'print-notification'~~

~~We may have to register a particular report sub-type for use with 'multipart/report' [RFC 1892]. See RFC 2298 for a similar kind of registration. The use of 'multipart/report' needs more understanding and work.~~

A. Appendix - Summary of Notification Attribute Usage

This appendix summarizes the usage of Notification attributes in the Job operation attribute collection, Job object, Subscription object, Notification content, Job operations, and Subscriptions operations.

ISSUE 6 - Should this appendix be kept in the spec or moved to the Implementer's Guide?

	Objects		Notification content		Job operations		Subscription operations		
Attributes	Job object	Subscription object	Job Notification Content	Printer Notification Content	Job Creation Validate-Job	Set-Job-Subscription	Create-Printer-Subscription	Renew-Subscription	Cancel-Subscription
1. notify-recipient~~s~~ (~~1setOf ~~uri)	R	R			[R]	[R]	[R]		
2. notify-events (1setOf type2 keyword)	R	R		100	[R]	[R]	[R]		
3. notify-content-type (mimeMediaType)	R	R			[R]	[R]	[R]		
4. subscriber~~notify~~-user-data (octetString(63))	R	R	R	R	[R]	[R]	[R]		
5. attributes-charset (charset)	R*		R (6)	R (6)	R	R	R	R	R
6. notify-attributes-charset (charset)	R	O~~R~~	~~R~~	~~R~~	[O]	[O]	[O]		
7. attributes-natural-language (naturalLanguage)	R*	~~R~~	R (8)	R (8)	R	R	R	R	R
8. notify-attributes-natural-language (naturalLanguage)	R	O~~R~~	~~R~~	~~R~~	[O]	[O]	[O]		
9. notify-lease-time-requested (integer(0:MAX))							[R]	[R]	
10. notify-lease-time-granted (integer(0:MAX))							Return	Return	
11. notify-lease-expiration-time (integer(0:MAX))		R					Return	Return	
12. notify-persistence-requested (boolean)							[O]		
13. notify-persistence-granted (boolean)							Return	~~Return~~	
14. trigger-event (type2 keyword)			R (2)	R (2)					
15. trigger-time (integer(MIN:MAX))			R	R					
16. trigger-date-time (dateTime)			[ ]	[ ]					
17. request-id (integer (0~~1~~:MAX))	R	R	R	R					
18. subscription-id (integer(1:MAX))	**	R	R	R			Return	R	R

Legend: "R" - REQUIRED for Printer to support; "O" - OPTIONAL for Printer to support; "[ ]" - OPTIONAL for sender to send; Return - Printer MUST return; "Per-Pr" - REQUIRED if Notification from a Per-Printer subscription; * - Job object, not collection;

"Per-Job" - REQUIRED if Notification from a Per-Job subscription; (nn) indicates source of value in a Notification;
** - 1setOf collection index is the implicit subscription-id for a Per-Job subscription and is sent in the Notification.

	Objects		Notification		Job operations		Subscription operations		
Attributes	Job object	Subscription object	Job Notification	Printer Notification	Job Creation Validate-Job	Set-Job-Subscription	Create-Printer-Subscription	Renew-Subscription	Cancel-Subscription
19. version-number (integer (0:32767))			R	R					
20. status-code (integer (0:32767))			(600)	(601)					
21. job-printer-uri (uri)	R		~~R (22)~~						
22. printer-uri (uri)		R	R (21)	R ~~(21)~~	R	R	R	R	R
23. printer-name (name(127))			R	R					
24. job-id (integer(1:MAX))	R		R	Per-Jb					
25. job-name (name(MAX)) [SNMP can truncate]	R		R	Per-Jb					
26. requesting-user-name (name(MAX))	R								
27. subscriber-user-name (name(MAX))		R	R (26)	R (26)					
28. job-state (type1 enum)	R		R						
29. job-state-reasons (1setOf type2 keyword)	R		R						
30. notify-printer-up-time (integer(1:MAX))		R							
31. job-k-octets-processed (integer(0:MAX))	O		[pg/cp]						
32. job-impressions-completed (integer(0:MAX))	O		[pg/cp]						
33. job-media-sheets-completed (integer(0:MAX))	O		[pg/cp]						
34. job-collation-type (type2 enum)	O		[pg/cp]						
35. sheet-completed-copy-number (integer(-2:MAX))	O		[pg/cp]						
36. sheet-completed-document-number(integer(-2:MAX))	O		[pg/cp]						
37. impressions-interpreted (integer(-2:MAX))	O		[pg/cp]						
38. impressions-completed-current-copy (integer(-2:MAX))	O		[pg/cp]						

Legend: "R" - REQUIRED for Printer to support; "O" - OPTIONAL for Printer to support; [ ]" - OPTIONAL for sender to send; "Return - Printer MUST return; "Per-Pr" - REQUIRED if Notification from a Per-Printer subscription; "Per-Jb" - REQUIRED if Notification from a Per-Job subscription; "[pg/cp]" - OPTIONAL in a 'job-progress' or 'job-completed' notification; (nn) - an alternate source of the data in a Notification.

	Objects		Notification		Job operations		Subscription operations		
Attributes	Job object	Printer object	Job Notification	Printer Notification	Job Creation Validate-Job	Set-Job-Subscription	Create-Printer-Subscription	Renew-Subscription	Cancel-Subscription
39. printer-state (type1 enum)		R		R					
40. printer-state-reasons (1setOf type2 keyword)		R		R					
41. printer-is-accepting-jobs (boolean)		R		R					
42. notify-schemes-supported (1setOf uriScheme)		R							
~~notify-content-types-supported (1setOf collection)~~		~~R~~							
43. notify-events-default (1setOf type2 keyword)		R							
44. notify-events-supported (1setOf type2 keyword)		R							
45. max-events-supported (integer(5:MAX))		R							
46. max-job-subscriptions-supported (integer(1:MAX))		R							
47. max-printer-subscriptions-supported (integer(0:MAX))		R							
48. notify-lease-time-supported (rangeOfInteger(0:MAX))		R							
49. notify-lease-time-default (integer(0:MAX))		R							
50. persistent-jobs-supported (boolean)		O~~R~~							
51. persistent-subscriptions-supported (boolean)		O							
52. printer-state-change-time (integer(1~~MIN~~:MAX))		O							
53. printer-state-change-date-time (dateTime)		O							

Legend: "R" - REQUIRED for Printer to support; "O" - OPTIONAL for Printer to support; "[ ]" - OPTIONAL for sender to send;
Return - Printer MUST return; "Per-Pr" - REQUIRED if Notification from a Per-Printer subscription;
"Per-Jb" - REQUIRED if Notification from a Per-Job subscription;
"[pg/cp]" - OPTIONAL in a 'job-progress' or 'job-completed' notification; (nn) - an alternate source of the data in a Notification.

17 Appendix C: Full Copyright Statement