INTERNET-DRAFT
<draft-ietf-ipp-not-spec-02.txt>

S. Isaacson
Novell, Inc.
J. Martin
Underscore
R. deBry
Utah Valley State College
T. Hastings
Xerox Corporation
M. Shepherd
Xerox Corporation
R. Bergman
Hitachi Koki Imaging Solutions

March 8, 2000

Internet Printing Protocol (IPP):
IPP Event Notification Specification

Abstract

This document describes an extension to the IPP/1.0, IPP/1.1, and future versions that allows a client to subscribe to printing related events. Subscriptions include "Per-Job subscriptions" and "Per-Printer subscriptions". One or more Per-Job Submission subscriptions are specified by the client when submitting a job. Additional Per-Job and Per-Printer subscriptions are created by performing separate explicit Create-Job-Subscription Create-Printer-Subscription operations, respectively. Subscriptions are modeled as Subscription objects. Four other operations are defined for Subscription objects: Get-Attributes, Get-Subscriptions, Renew-Subscription, and Cancel-Subscription.

A subscription request and the Subscription object includes: the names of Job and/or Printer events, the Notification Recipient URL, the

Notification Content format, if several are supported, possibly some opaque data, and the charset and natural language. In addition, the subscription request includes: the requested lease time and persistency for the subscription. When the event occurs, a notification is generated and delivered using the information specified in the subscription.

1 ISSUE is included in the text.

The full set of IPP documents includes:

  Design Goals for an Internet Printing Protocol [RFC2567]
  Rationale for the Structure and Model and Protocol for the Internet
     Printing Protocol [RFC2568]
  Internet Printing Protocol/1.1: Model and Semantics [IPP-MOD]
  Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]
  Internet Printing Protocol/1.1: Implementer's Guide [IPP-IIG]
  Mapping between LPD and IPP Protocols [RFC2569]

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/1.1: Implementer's Guide" document gives insight and advice to implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.0 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

Tables

Figures

1 Introduction

This IPP notification specification is an extension to IPP/1.0 [RFC2568, RFC2569] and IPP/1.1 [ipp-mod, ipp-pro]. 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]

In addition, each notification delivery method, whether REQUIRED or OPTIONAL, is described in separate documents:

Internet Printing Protocol/1.0 & 1.1: "Notification Delivery Method xxx [TBD]
Internet Printing Protocol/1.0 & 1.1: "Notification Delivery Method yyy [TBD]

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 Subscription object and its attributes.

- Section 6 defines the Printer Description attributes

- Section 7 defines the Notification content of Human Consumable and Machine Consumable Event formats.

- Sections 8 and 9 define the Per-Job and Per-Printer Subscription operations.

- Section 10 defines the out-of-band values.

- Section 11 and 12 define the conformance requirements and IANA requirements, respectively.

- Section 13 - 15 cover Internationalization, Security, and Status codes.

- Appendix A is a summary of the Notification Attribute usage

- Appendix B is a Change History

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-Job or Per-Printer subscriptions are created by performing separate explicit Create-Job-Subscription or Create-Printer-Subscriptions operations, respectively.

A Per-Job or Per-Printer subscription request 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. if Human Consumable notification content is to be sent, which text format

4. some opaque data that the subscriber wants to be sent 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

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 contains one or more collection values, each consisting of a number of member attributes that specify a subscription, so that a Job can have more than one Per-Job subscription. The 'collection' is a new attribute syntax (see [ipp-coll]). The member attributes of each collection value are copied to separate Subscription objects to populate the corresponding Subscription Description attributes.

For Per-Printer subscriptions and Per-Job subscriptions created after the Job has been created, 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. The Create-Job-Subscription and Create-Printer-Subscription operations create 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-Subscription operation.

Table 1 summarizes the Per-Job and Per-Printer Subscription operations, assigns their operation-id (see [ipp-mod] section 4.4.15) and their salient input operation attributes.

Table 1 - Summary of Per-Job and Per-Printer Subscription operations

Operation:	Oper. ID	Per-Job	Per-Printer	Brief Description salient inputs beside printer-uri:
Print-Job, Print-URI, Create-Job	see [ipp-mod]	yes	no	Create one or more Per-Job Subscriptions as part of the Job Creation operations 1setOf {recipient, [events,] [format,] [user-data,] [charset,] [natural-language]}
Validate-Job	see [ipp-mod]	yes	no	Check that the Job Creation operation would be accepted including the Subscription 1setOf {recipient, [events,] [format,] [user-data,] [charset,] [natural-language]}
Create-Printer-Subscription	0x0016	no	yes	Create a Per-Printer Subscription recipient, [events,] [format,] [user-data,] [charset,] [natural-language,] [lease-time-requested,] [persistence-requested]
Create-Job-Subscription	0x0017	yes	no	Create a Per-Job Subscription recipient, job-id, [events,] [format,] [user-data,] [charset,] [natural-language,]
Get-Subscription-Attributes	0x0018	yes	yes	Get specified Subscription attributes of the specified Subscription object subscription-id, [requested-attributes]
Get-Subscriptions	0x0019	yes	yes	Get the specified Subscription attributes from all my or all of the Subscription objects [job-id], [my-subscriptions,] [requested-attributes]

Renew-Subscription	0x001A	yes	yes	Renew the Subscription lease subscription-id, [lease time-requested]
Cancel-Subscription	0x001B	yes	yes	Cancel the 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, 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 Printer increments the Subscription object's "subscription-request-id" (integer (0:MAX)) attribute and a notification is generated and delivered using the methods and target addresses identified in the subscription, passing the incremented "request-id". The "subscription-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 subscriptions are created by a client (desktop or server acting as a client) as part of creation of the job in an IPP Printer (printing device or server). More than one subscription may be submitted with a job. Additional subscriptions may be associated with the job using the Create-Job-Subscription operation. The IPP Printer object delivers a Notifications to the Notification Recipient 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                     +---------------+
        +--------+                         |  ###########  |
        | client |---IPP job submission------># Printer #  |
        +---------+  Per-Job subscription   |  # Object  #  |
     +------------+                         |  #####|#####  |
     |Notification|                         +-------|-------+
     |Recipient   |<-----IPP Notifications---------+
     +------------+    (Job and/or Printer events)
```

Figure 1 - Client-Printer Per-Job 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 in turn each implement an IPP Printer object (3 and 4, respectively).

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 |---Create-Printer-Subscription------># Printer #  |
       +--------+    (Per-Printer subscription)    |  # Object  #  |
                                                   |  #####|#####  |
    +------------+                                 +-------|-------+
    |Notification|<----------IPP Notifications-------------+
    |Recipient   |    (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

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. This section describes two methods of using third party notification services. The first is transparent to the client and the second is transparent to the IPP Printer.

2.3.1 Use of a Notification Service transparently to clients

An implementation may be configured to use some other notification service to either (1) delivery the Notifications to the Notification

Recipient(s) specified in the IPP Subscription or (2) keep the Subscriptions, accept events, possibly format the notification in the natural language of the Notification Recipient when a Human Consumable text format is used, and deliver the Notifications to the Notification Recipient(s) indicated in the IPP Subscription. Figure 4 shows this partitioning.

```
                                        output device or server
     desktop or server                     +---------------+
        +--------+                         |  ###########  |
        | client |---IPP subscription-------->#  Printer #  |
        +--------+                         |  # Object  #  |
                                           |  #####|#####  |
                                           +-------|-------+
                                                   | Subscriptions
                                            *******| OR Events
     +------------+                        *       |
     |Notification|                       * +------v--------+
     |Recipient   |<--IPP Notifications-----| Notification  |
     +------------+                      *  | Service       |
                                         *  +---------------+
                                         *
                                         *

     *** = Implementation configuration opaque boundary
```

Figure 4 - Opaque Use of a Notification Service Transparent to the Client

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 the client and this specification.

2.3.2 Use of Notification Service transparently to the IPP Printer

Another way that a Notification Service can be used is if the Notification Recipient indicated in the IPP Subscription is a notification service (transparent to the IPP Printer), which in turn forwards the Notification to the Ultimate Notification Recipient using additional parameters in the IPP Subscription (URI parameters or subscriber user data). In such cases, the Ultimate Notification Recipient has also subscribed directly with the other notification service (by means outside this document). As far as the IPP Printer is concerned, the IPP Subscription indicated that the IPP Printer is to delivery Notifications to the Notification Recipient (Notification Service) using the specified notification delivery method. The method that the Notification Recipient uses for delivering the notification to the Ultimate Notification Recipient is beyond the scope of this document and is transparent to the IPP Printer. However, the client does have to know how to pass additional information to the Notification Recipient in the IPP Subscription using either extra parameters in the URI or subscriber user data. Examples of this latter approach are paging, immediate messaging services, and NOS vendors infrastructure. Figure 5 shows this approach.

```
       desktop or server                     server or printing device
                                                   +---------------+
       +--------+                                  |  ###########  |
       | client |----------IPP Subscriptions--------->#  Printer #  |
       +--------+                                  |  # Object  #  |
                                                   |  #####|#####  |
+------------+     +------------+                  +-------|-------+
|Ultimate    |     |Notification|<---IPP Notifications-----+
|Notification|<----|Recipient   |
|Recipient   |     +------------+
+------------+     (Notification Service)
```

Figure 5 - Use of a Notification Service transparent to the IPP Printer

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].

REQUIRED - an adjective used to indicate that a conforming IPP Printer implementation MUST support the indicated operation, object, attribute, attribute value, status code, or out-of-band value in requests and responses. See [ipp-mod] "Appendix A - Terminology for a definition of "support". 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 OPTIONAL notification specification is implemented".

RECOMMENDED - an adjective used to indicate that a conforming IPP Printer implementation is recommended to support the indicated operation, object, attribute, attribute value, status code, or out-of-band value in requests and responses. 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 "RECOMMENDED if this OPTIONAL notification specification is implemented".

OPTIONAL - an adjective used to indicate that a conforming IPP Printer implementation MAY, but is NOT REQUIRED to, support the indicated operation, object, attribute, attribute value, status code, or out-of-band value in requests and responses.

settable - an adjective used to indicate that a Printer implementation supports setting the value(s) of an attribute using the Set-Job-Attributes or Set-Printer-Attributes operations (see [ipp-set]).

READ-ONLY - an adjective used in an attribute definition to indicate that an IPP Printer MUST NOT support the attribute as being settable using the Set-Job-Attributes or Set-Printer-Attributes operations (see [ipp-set]).

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.

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 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 Subscription - A Subscription that a client specifies as part of a create job operation (Print-Job, Print-URI, Create-Job), a Validate-Job operation, or an explicit Create-Job-Subscription operation with a Job object as the target.

Per-Printer Subscription - A Subscription that a client specifies using an explicit Create-Printer-Subscription operation with a Printer object as the target.

Notification Source - The entity that sends Event Notifications. It MAY be the IPP Printer itself or the IPP Printer MAY be configured to use a Notification Service to delivery Notifications transparently to the subscribing clients (see Figure 4).

Notification Recipient - The entity identified as a recipient within a subscription that receives IPP Notifications about Job and/or Printer events (see Figure 4 and Figure 5). A Notification Recipient may be a: Job Submitting End User, Job Submitting Application (desktop or server), Job Recipient, Job Recipient Proxy, a Notification Service, an Operator, or Administrator, etc., and their representative or log file or usage statistics gathering application or other active or passive entities. A Java Listener is an example of a Notification Recipient.

Ultimate Notification Recipient - The entity to which the Notification Recipient (stores and) forwards an IPP Notification when the Notification Recipient is a Notification Service (see Figure 5).

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 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 has subscribed to be notified for that event or not. 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 Notification Recipient that has a subscription 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 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 and Instant Messaging services are examples of a store and forward notification delivery method.

Human Consumable Notification - Notifications that are intended to be consumed by human End Users only. 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. They use the encoding of an IPP response. The Notification Recipient must localize the contents, if displaying it to a human.

Subscription Creation operation - One of the operations that creates a Subscription object: Job Creation operations (Create-Job, Print-Job, and Print-URI), Create-Job-Subscription, and Create-Printer-Subscription.

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 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 object
                     ++---------++
                      |         |
       +----+         |   j2    |
       | s5 |<------->|         |
       +----++        |         |
        | s6 |<------>|         |    s5 and s6 are Per-Job subscription
        +----+        ++---------++                 objects
                       |         |
                       |   j3    |
                       |         |
                       |         |         <----> indicates association
                       +---------+
```

Figure 6 - Object Model for Notification

s1, s2, and s3 are Per-Printer Subscription objects and can identify Printer and/or Job events.
s4, s5, and s6 are Per-Job subscription objects and can identify Printer and/or Job events.

4.1 Object relationships

This sub-section defines the object relationships between the Printer, Job, and Subscription objects. Whether Per-Printer Subscription objects are actually contained in a Printer object or are just bi-directionally associated with them in some way is IMPLEMENTATION DEPENDENT and is transparent to the client. Similarly, whether Per-Job Subscription objects are actually contained in a Job object or are just bi-directionally associated with them in some way is IMPLEMENTATION DEPENDENT and is transparent to the client. The object relationships are defined as follows:

4.1.1 Printer object and Per-Printer Subscription objects

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

2. A Per-Printer Subscription object (s1-s3) cannot be contained in (or associated with) more than one Printer object (p1).

3. Each Per-Printer Subscription object (s1, s2, and s3) is contained in (or is associated with) one Printer object (p1) and each represents one Per-Printer subscription.

4. 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.

4.1.2 Job object and Per-Job Subscription objects

1. A Job object (j1, j2, j3) contains (is associated with) zero or more Per-Job subscription objects (s4-s6). Job j1 is associated with Per-Job subscription object s4, Job j2 is associated with Per-Job subscription objects s5 and s6, and Job j3 is not associated with any Per-Job subscription object.

2. A Per-Job Subscription object cannot be contained in (or associated with) more than one Job object.

3. Each Per-Job Subscription object is contained in (or associated with) one Job object and each represents one Per-Job Subscription.

4. Each Per-Job Subscription 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 Subscription Object attributes

Table 2 lists the Subscription object attributes defined in this section and the related Printer Description attributes defined in section 6, if any. The definitions of the object attributes are specified in this section so that they can be referred to from the subsequent definitions of the operations that set them. All of the Subscription object attributes are READ-ONLY. They can be set only by specific operations that create or perform operations on Subscription objects (see section 8).

Table 2 - Subscription object attributes

Subscription object attributes:	Printer support	Related Printer Description attribute(s)
Set by client input to a Subscription Creation operation		
notify-recipient (uri)	REQUIRED	notify-schemes-supported (1setOf uriScheme)
notify-events (1setOf type2 keyword)	REQUIRED	notify-events-default (1setOf type2 keyword) notify-events-supported (1setOf type2 keyword) max-events-supported (integer(5:MAX))
notify-format (mimeMediaType)	REQUIRED	notify-format-supported (1setOf mimeMediaType)
subscriber-user-data (octetString(63))	REQUIRED	n/a
notify-charset (charset)	OPTIONAL	charset-supported (1setOf charset)
notify-natural-languages (1setOf naturalLanguage)	OPTIONAL	generated-natural-language-supported (1setOf naturalLanguage)
Printer initializes to 0, increments at beginning of each event		
subscription-request-id (integer(0:MAX))	REQUIRED	n/a

Subscription object attributes:	Printer support	Related Printer Description attribute(s)
Printer sets as part of Subscription Creation operation		
subscription-id (integer(1:MAX))	REQUIRED	max-job-subscriptions-supported (integer(0:MAX)) max-printer-subscriptions-supported (integer(0:MAX))
notify-lease-expiration-time (integer(0:MAX))	REQUIRED	notify-lease-time-default (integer(0:MAX)) notify-lease-time-supported (rangeOfInteger(0:MAX))
printer-uri (uri)	REQUIRED	printer-uri-supported
subscriber-user-name (name(MAX))	REQUIRED	n/a

Subscription object attributes:	Printer support	Related Printer Description attribute(s)
Returned by Printer on Subscription Creation operation		
notify-server-up-time (integer(1:MAX))	REQUIRED	printer-up-time (integer(1:MAX))
notify-persistence-granted (boolean)	REQUIRED	persistent-subscriptions-supported (boolean)

Note: The Subscription object does not contain the "job-id" Subscription Description attribute. The Get-Subscriptions operation has the "job-id" as an input operation attribute, so the "job-id" isn't returned in the response. If an implementation needs such a link between Subscription objects and Job objects, then it keeps such a link as in internal attribute. The intent is that whether Per-Job

Subscription objects are actually contained in a Job object or are just associated with them in some way is IMPLEMENTATION DEPENDENT and is transparent to the client.

5.1 notify-recipient (uri)

This REQUIRED READ-ONLY Subscription object attribute describes both where (the address of the Notification Recipient) 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.

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 supplementary 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, 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).

The following notification delivery schemes are defined in other documents for use as part of the URI value of this attribute:

'ipp:' - The Notification Recipient uses server directed polling to pull Notifications [ipp-method-poll]

'indp:' - The Notification Source sends Send-Notifications operations to the Notification Recipient [ipp-method-indp]

'mailto:' - The Notification Source sends a email message using SMTP with possible attachments containing Machine Consumable Notification Content [ipp-method-mailto]

'snmpnotify:' - The Notification Source sends SNMP traps and/or informs to the Notification Recipient [ipp-method-snmp]

This list of notification delivery schemes will be added to as needed using the registration procedures defined in [ipp-mod].

ISSUE 01 - 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?

5.2 notify-events (1setOf type2 keyword)

This REQUIRED READ-ONLY Subscription object attribute identifies the job and/or printer events that are to be delivered to the Notification Recipient as Notifications as defined in section 7. If the client did not supply this attribute when supplying the subscription, the Printer object populates this attribute with its "notify-events-default" attribute value (see section 6.2).

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. Job and printer events have the following semantics depending on whether the subscription is a Per-Job or a Per-Printer subscription:

Job events: The semantics for job events depends on whether the subscription is a Per-Job Subscription or a Per-Printer Subscription. For Per-Job Subscriptions, the Printer MUST generate Notifications only for the job events of the job with which the Subscription is associated. For Per-Printer Subscriptions, the Printer MUST generate Notifications for the job events for any job submitted to the Printer.

For example, consider the event when there are 10 'pending' jobs, one 'processing' job, and 30 'completed' jobs and the 'processing' job completes. The Printer MUST generate a 'job-completed' job event Notification if the job has a Per-Job Subscription that contained the 'job-completed' event in this attribute. The Printer MUST NOT generate Notifications for any other jobs whose Per-Job Subscriptions contain the 'job-completed' event in this attribute.

Printer events: The semantics for printer events does not depend on whether the subscription is a Per-Job or a Per-Printer Subscription. For both Per-Job Subscriptions and Per-Printer Subscriptions, the Printer MUST generate Notifications for printer events, no matter what job is processing, including when no jobs are processing.

For example, consider the event when there are 10 'pending' jobs, one 'processing' job, and 30 'completed' jobs and the Printer enters the 'stopped' state. The Printer MUST generate a 'printer-state-changed' printer event Notification for each of the 11 'pending' and 'processing' jobs whose Per-Job

Subscriptions contain the 'printer-state-changed' event in this attribute. The Printer MUST NOT generate Notifications for any of the 'completed' jobs' whose Per-Job Subscriptions contain the 'printer-state-changed' event in this attribute.

The events are defined to be disjoint. For example, the 'job-state-changed' event does not include the 'job-created' and 'job-completed' events. In order to get all three events, the client supplies all three keywords in this attribute.

A Printer MUST support the events indicated as "REQUIRED" and MAY support of the events indicated as "OPTIONAL".

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 a Job Creation operation (Print-Job, Print-URI, or Create-Job) and the job's "time-at-creation" attribute value is set (see [ipp-mod] section 4.3.14.1). The Printer puts the job in the 'pending', 'pending-held' or 'processing' states.
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 "date-time-at-completed" (if supported) attributes are set (see [ipp-mod] section 4.3.14).
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.

'job-state-changed': REQUIRED - the job has changed from any state 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 ('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.

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.

'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. Often, such a change is the result of the user or the operator performing a Set-Job-Attributes operation (see [ipp-set]) on the Job object. The client performs a Get-Job-Attributes to find out the new values of the changed attributes. This event 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.

The standard Printer event keywords values are:

'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-mod]). 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 (see [ipp-set2]). 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 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.

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 has to perform a Get-Printer-Attributes to find out the new values of these changed attributes. This event 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 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 notify-format (mimeMediaType)

This REQUIRED READ-ONLY Subscription object attribute indicates the MIME Media type of Human Consumable and/or Machine Consumable format content that is to be sent in the Notifications for the Notification Content that is under control of the Subscriber. Depending on the delivery method definition document, this attribute MAY specify the MIME Media type for the entire Notification Content, or just a portion that is controllable by the subscriber, if there is a fixed part as well. For example, if the entire Notification Content is always a fixed Machine Consumable MIME Media type, such as the 'ipp:' delivery method which uses the 'application/ipp' MIME Media type (see [indp-method]), then this attribute controls the format of the "human-readable-report" (text(MAX)) Notification Content attribute (see section 7.2) in the Machine Consumable form. As another example, if the Notification Content is always a Human Consumable format content with a MIME Media attachment, such as with the 'mailto:' delivery method [ipp-method-mailto], this attribute controls the format of the attachment.

If the 'text' MIME media type registration permits a charset parameter, than such a specification MUST be used (instead of the "notify-charset" attribute - see section 5.5) 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 Subscription object attribute with either the 'none' out-of-band value (see section 10.1) or one of the values of the Printer's "notify-format-supported" attribute (see section 6.5), depending on the delivery method and implementation, since there is no "notify-format-default" defined in this document.

Standard mimeMediaType values are:

'none': This out-of-band value (see section 10.1) indicates that the Subscriber-controllable part of the Notification content is not to be sent.
'text/plain; charset=utf-8': Human Consumable plain text containing characters from ISO 10646 represented as UTF-8 [RFC2279] as defined in section 7.1.1.
'text/html': Human Consumable HTML data [RFC1866] as defined in section 7.1.1.
'text/xml': Machine Consumable XML data [RFC2376] as defined in section 7.1.2.
'application/ipp': Machine Consumable IPP request as defined in section 7.1.2 data encoded as in [ipp-pro].
'application/postscript': Human Consumable PostScript [RFC2046]
'image/tiff': Human Consumable image data [RFC2302]

5.4 subscriber-user-data (octetString(63))

This REQUIRED READ-ONLY Subscription object attribute holds opaque information being sent from the Subscriber to the Notification Recipient, 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 and the ultimate recipient isn't included in the value of the "notify-recipient" attribute. An Instant Messaging Service is an example of such a general application where the "subscriber-user-data" might be the user's id for that messaging service and the "notification-recipient" is the URL of the messaging service.

5.5 notify-charset (charset)

This OPTIONAL READ-ONLY Subscription object attribute specifies the charset to be used in the Notification content sent to the Notification Recipient, whether the notification content is Machine Consumable or Human Consumable. This attribute MUST NOT be used when the "notify-format" attribute value specifies the charset parameter in its MIME media type value, e.g., 'text/plain; charset=utf-8'.

5.6 notify-natural-language (naturalLanguage)

This OPTIONAL READ-ONLY Subscription object attribute specifies the natural language for the IPP object to use in the Notification content that is sent to the Notification Recipient, whether the notification content is Machine Consumable or Human Consumable.

5.7 subscription-request-id

This REQUIRED READ-ONLY Subscription object attribute holds the most recent request-id sequence number delivered in a Notification content to the Notification Recipient. A value of 0 indicates that no Notifications have been sent for this subscription. The first request-id sent for a subscription MUST be 1. Each Notification Recipient receives its own monotonically increasing series of request-ids operation parameters (see [ipp-mod] section 3.1.2), i.e., no gaps, in order to be able to detect a missing notification.

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

This REQUIRED READ-ONLY Subscription object attribute uniquely identifies this Subscription object instance on this Printer object or this Job object. The Printer object, on acceptance of a Create-Job-Subscription or Create-Printer-Subscription request, generates an ID which identifies the new Subscription object on that Printer or Job. The Printer returns the value of the "subscription-id" attribute as part of the response to a Create-Job-Subscription or 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. Note: it is assumed that Per-Job subscriptions are persistent if Jobs are persistent, in order to be consistent with the persistency of Job objects. The [ipp-mod] RECOMMENDS that Job objects be persistent.

5.9 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 Per-Printer Subscription object, it populates this attribute with the appropriate value. When the indicated time arrives, the Printer MUST delete the Per-Printer Subscription object. Per-Job Subscription objects always return a value of 0 since Per-Job Subscriptions don't have a lease, but exist for the life-time of the Job instead.

A client is able to extend a lease of a Per-Printer subscription using the Renew-Subscription operation (see section 8.3.3). 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 6.8) and the subscriber is authorized to request an infinite lease. A Per-Job subscription cannot be renewed.

Note: In order to compute the number of seconds remaining in a Per-Printer Subscription lease, a client can subtract the "notify-server-up-time" Subscription object attribute (see section 5.12) from the "notify-lease-expiration-time" Subscription object attribute.

5.10printer-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.

5.11subscriber-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). For Per-Job subscriptions created as part of the Job creation operation, the value of the "subscriber-user-name" is the same as the "job-originating-user-name" Job attribute (see [ipp-mod] section 4.3.6).

The value of the "subscriber-user-name" is implementation dependent when a server accepts a request and forwards it to a downstream IPP Printer (see Figure 2 and the [ipp-iig]).

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.

5.12notify-server-up-time (integer(1:MAX))

This REQUIRED READ-ONLY Subscription object attribute indicates the amount of time (in seconds) that the Printer or Notification Delivery Service 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) if the Printer is keeping the Subscription objects or is the up time for the Notification Delivery Service if the Printer has delegated the responsibility for keeping Subscription objects to a notification delivery service, in an analogous way that the Job's "job-printer-up-time" is an alias for "printer-up-time" Printer attribute (see [ipp-mod] section 4.3.13.4).

Note: The purpose of this attribute is so that a client can request this attribute in a Get-Subscription-Attributes or Get-Subscriptions request and use the value returned in combination with the "notify-lease-expiration-time" (see section 5.9) in order to display the wall clock time equivalent to the user. The difference between this attribute and the 'integer' value of the "notify-lease-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.

5.13notify-persistence-granted (boolean)

This REQUIRED READ-ONLY Subscription object attribute indicates whether or not the Per-Job or Per-Printer Subscription is persistent, i.e., saved across power cycles in an implementation-define manner.

6 Printer Description Attributes related to Notification

This section defines the Printer Description attributes that are related to Notification. Table 3 lists the Printer Description attributes and indicates the Printer support required for conformance and whether or not the attribute is READ-ONLY: .

Table 3 - Printer Description attributes associated with Notification

Printer object attributes:	Printer support	READ-ONLY?
notify-schemes-supported (1setOf uriScheme)	REQUIRED	no
notify-events-default (1setOf type2 keyword)	REQUIRED	no
notify-events-supported (1setOf type2 keyword)	REQUIRED	no
max-events-supported (integer(5:MAX))	REQUIRED	no
notify-format-supported (1setOf mimeMediaType)	REQUIRED	no
max-job-subscriptions-supported (integer(0:MAX))	REQUIRED	no
max-printer-subscriptions-supported (integer(0:MAX))	REQUIRED	no
notify-lease-time-supported (rangeOfInteger(0:MAX))	REQUIRED	no
notify-lease-time-default (integer(0:MAX))	REQUIRED	no
persistent-jobs-supported (boolean)	OPTIONAL	no
persistent-subscriptions-supported (boolean)	OPTIONAL	no
printer-state-change-time (integer(1:MAX))	PTIONAL	yes
printer-state-change-date-time (dateTime)	OPTIONAL	yes

6.1 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 5.1.

6.2 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 a Subscription Creation operation. Any value in this attribute MUST also appear in the "notify-events-supported" attribute, i.e., be a supported event.

6.3 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. Standard values are defined in Section 5.2.

6.4 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 'client-error-too-many-events (see section 15.4). If notification is not supported, this attribute MUST NOT be supported.

6.5 notify-format-supported (1setOf mimeMediaType)

This REQUIRED Printer attributes identifies the MIME media types supported for Human Consumable and/or Machine Consumable notification content that the subscriber can control, if any. If the Notification content is fixed by the implementation, then the value of this attribute is that MIME media type, if is has been registered, or the 'none' out-of-band value (see section 10.1) otherwise. See the definition of the "notify-format" attribute in section 5.3 for details and some example values.

Not all formats listed in the "notify-format-supported" attribute need be available with all delivery methods specified in the "notify-schemes-supported". However, all Human Consumable formats, if any, SHOULD be available with all delivery methods. It is much harder to support a Human Consumable format because of localization issues. Once the code is written to support a particular Human Consumable format, it is easy to transmit it on any of the supported notify-schemes. Thus, if a vendor decides to support a notify-scheme, it has already committed to implement the Machine Consumable format. This may be simple if existing code can be reused, e.g. application/ipp and or more difficult if new code must be written, e.g., it is SNMP.

6.6 max-job-subscriptions-supported (integer(0:MAX))

This REQUIRED Printer attribute specifies the maximum number of Per-Job subscriptions that are supported for a job, i.e., the maximum number of collection values for the "job-notify" operation attribute, and/or the maximum number of subsequent Create-Job-Subscription operation requests in combination for a job. 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 exceeds the value of this attribute or would exceed some implementation-defined total number of Per-Job Subscriptions for the Printer, the Printer MUST accept the Job Creation and ignore the excess subscriptions. If a subsequent Create-Job-Subscription request would exceed this number, the Printer rejects the request and returns the 'client-error-too-many-subscriptions' (see section 15.3).

If a Printer does not support Per-Job Subscriptions, it MUST NOT support this attribute. The usual way for a client to determine whether an IPP Printer supports a feature is to query the Printer's "operations-supported" attribute. However, there is no new REQUIRED operation for Per-Job Subscriptions. Therefore, the RECOMMENDED way for a client to determine whether or not a Printer supports Per-Job Subscriptions is to query this attribute to see if it is supported (since the Create-Job-Subscriptions is an OPTIONAL operation).

6.7 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 Per-Printer 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 exceeds the value of this attribute or would exceed some implementation-defined total number of Per-Printer Subscriptions for the Printer (if any), the Printer rejects the Create-Printer-Subscription request and returns the 'client-error-too-many-subscriptions' (see section 15.3).

If a Printer does not support Per-Printer Subscriptions, then it MUST NOT support this attribute. None the less, the RECOMMENDED way for a client to determine whether or not a Printer supports Per-Printer Subscriptions is to query the "operations-supported" to see if the Create-Printer-Subscriptions operation is supported, rather than querying this attribute.

6.8 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-Subscription request for a Per-Printer subscription. 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 5.9). A value of 0 indicates an infinite lease, i.e., one that does not expire.

6.9 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 Create-Printer-Subscription or Renew-Subscription requests.

6.10persistent-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.

As with any settable attribute, if the Printer supports setting this attribute to more than one value, then the value being set MUST control whether or not the Printer keeps Jobs persistently as per the rules in [ipp-set]). As with any settable attribute, if the Printer only supports one value, it MAY support this attribute either (1) as settable to that one value, or as not-settable, depending on implementation (see [ipp-set]).

6.11persistent-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.

As with any settable attribute, if the Printer supports setting this attribute to more than one value, then the value being set MUST control whether or not the Printer keeps Jobs persistently as per the rules in [ipp-set]). As with any settable attribute, if the Printer only supports one value, it MAY support this attribute either (1) as settable to that one value, or as not-settable, depending on implementation (see [ipp-set]).

6.12printer-state-change-time (integer(1:MAX))

This OPTIONAL READ-ONLY 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.

6.13printer-state-change-date-time (dateTime)

This OPTIONAL READ-ONLY Printer attribute records the date and time, i.e., copy of the Printer's "printer-current-time" (dateTime) attribute (see [ipp-mod] section 4.4.30), 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.

7 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, i.e., 'text' MIME media type. For most notification delivery methods both content types are defined. Each Notification Content type will either define one specific Machine Consumable form (usual) or indicate that no Machine Consumable form is defined. In addition, each Notification Content type will indicates whether (usual) or not Human Consumable forms are permitted. But the definition will not define the Human Consumable forms. For those Human Consumable forms that a Printer implementation supports as indicated in the Printer's "notify-format-supported" attribute, it MUST support for all Notification Content formats supported that permit Human Consumable form.

7.1 Notification content MIME media type formats

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. The Notification is either in a Human Consumable or Machine Consumable form.

7.1.1 Human Consumable form

If the notification delivery method is defined to permit Human Consumable forms such as the entire Notification Content or a portion of the Notification Content, such as either an attachment or the "human-readable-report" attribute (see section 7.2) then the following RECOMMENDATIONS apply:

The text message SHOULD include information about the attributes in sections 7.2 and 7.3 for job events or in sections 7.2 and 7.4 for printer events. This information is localized according to the information about natural language and charset in the subscription.

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

7.1.2 Machine Consumable form

If the notification delivery method is defined to have a Machine Consumable form and that form is defined to be the 'application/ipp'

MIME media type [ipp-mod], then the following rules apply:

1. The notification content MUST use the 'application/ipp' MIME media type [ipp-mod] using one of: (1) the Get-Job-Attributes response encoding for job events, (2) Get-Printer-Attributes response for printer events, or (3) a new Sent-Notifications operation for both job and printer events.

2. The attributes listed in sections 7.2 and 7.3 are sent in a Notification for Job events.

3. The attributes listed in sections 7.2 and 7.4 are sent in a Notification for Printer events.

4. For any 'text' or 'name' attribute value in any notification, the charset and natural language rules that apply to all IPP operations apply to these attributes as well, since they are represented as operation responses.

5. The "human-readable-report" attribute can be defined to be REQUIRED, OPTIONAL, or MUST NOT be sent.

6. 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. Notification Recipients MUST be able to accept Notifications containing attributes they do not recognize. What a Notification Recipient does with an unrecognized attribute is implementation-dependent. Notification Recipients MAY attempt to display unrecognized attributes anyway or MAY ignore them.

7.2 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.

The notification delivery method either models the Notification as a (1) request if the IPP Printer sends the Notification to the Notification Recipient when the event occurs or as a (2) response if the IPP Printer is queried for the Notification by a Notification Recipient:

> Request: If the delivery method defines that the IPP Printer sends the Notification, then the delivery method also defines whether or not the Notification Recipient returns a response. If the IPP Printer sends a Notification request, it uses the "operation-id alternative for item #0 in Table 4:
>
> Response: If the delivery method defines that the IPP Printer supports an operation that returns a Notification as a response to the Notification Recipient, it uses the "status-code" alternative for item #0 in Table 4.

Some events do not include all of these attributes as shown in Table 4. 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 the Printer will send or return them as separate Job Notification content and Printer Notification contents to the same Notification Recipient.

Table 4 lists the attributes that are defined for use in Notifications and indicates the Printer support required for conformance: References of the form "mod m.n.o" refer to [ipp-mod] sections.

Table 4 - Common Job and Printer Notification content attributes

Attributes	Reference	Events	
		'job-progress'	all others
The following attributes MUST occur in this order:			
version-number (integer (0:32767))	mod 3.1.1	REQUIRED	REQUIRED
operation-id (integer (0:32767)) or status-code (integer (0:32767))	mod 3.1.1	REQUIRED	REQUIRED
request-id (integer (0:MAX))	5.7 & mod 3.1.1	REQUIRED	REQUIRED
attributes-charset (charset)	5.5 & mod 3.1.4	REQUIRED	REQUIRED
attributes-natural-language (naturalLanguage)	5.6 & mod 3.1.4	REQUIRED	REQUIRED
printer-uri (uri)	5.10	REQUIRED	REQUIRED
The following attribute MAY occur in any order:			
printer-name (name(127))	mod 4.4.4	n/a	REQUIRED
job-id (integer(1:MAX))	mod 4.3.2	REQUIRED	REQUIRED**
job-name (name(MAX))	mod 4.3.5	n/a	REQUIRED**
trigger-event (type2 keyword)	7.2.1	REQUIRED	REQUIRED
trigger-time (integer(MIN:MAX))	7.2.2	REQUIRED	REQUIRED
trigger-date-time (dateTime)	7.2.3	n/a	OPTIONAL

subscription-id (integer(1:MAX))	5.8	REQUIRED	REQUIRED
subscriber-user-name (name(MAX))	5.11	REQUIRED	REQUIRED
subscriber-user-data (octetString(63))	5.4	REQUIRED	REQUIRED
notify-format (mimeMediaType)	5.3	OPTIONAL	OPTIONAL

Attributes	Reference	Events	
		'job-progress'	all others
human-readable-report (text(MAX))	7.2.4	OPTIONAL	OPTIONAL

Attribute Notes:

"version-number" - the major and minor version number.

"operation-id" - the operation id if the delivery method document models the Notification as the Printer sending the Notification to the Notification Recipient (whether the Notification Recipient returns a response or not).

"status-code" - a value of 0x600 (hexadecimal 600) for a Job event and 0x601 (hexadecimal 601) for a Printer event if the delivery method document models the Notification as the Printer returning the Notification to the Notification Recipient as a response to a request.

"request-id" - the sequence number for this subscription, starting at 1, for each subscription, i.e., the Printer copies the current value of the "subscription-request-id' Subscription object attribute to the Notification request content.

"attributes-charset" - the value comes from the "notify-charset" attribute in the Subscription object.

"attributes-natural-language" - the value comes from the "notify-natural-language" attribute in the 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.

"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. See [ipp-method-snmp]

"subscription-id" - the unique identifier for the Subscription object on this Printer.

"subscriber-user-name" - the subscriber user name that created the Subscription object. 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. See [ipp-method-snmp] "subscriber-user-data" - opaque user data that may identify either the Subscriber and/or the ultimate Notification Recipient. The client MUST supply this attribute, if the definition of the delivery method specified in the "notify-recipient" attribute REQUIRES the client to supply it.

"notify-format (mimeMediaType) - indicates the text MIME media type (see section 5.3) for the "human-readable-report" attribute. This attribute MUST be present if the "human-readable-report" (text(MAX)) attribute is present in order to unambiguously identify the format of its text value to the Notification Recipient.

7.2.1 "trigger-event" (type2 keyword)

This REQUIRED Notification Content attribute indicates the event that caused this Notification to be delivered.

7.2.2 "trigger-time" (integer(MIN:MAX))

This REQUIRED Notification Content attribute indicates the time at which the event occurred, i.e., the "printer-up-time" value (see [ipp-mod] section 4.4.29) when the event occurred.

7.2.3 "trigger-date-time" (dateTime)

This OPTIONAL Notification Content attribute indicates the date and time at which the event occurred, i.e., the "printer-current-time" value (see [ipp-mod] section 4.4.30) when the event occurred.

7.2.4 "human-readable-report" (text(MAX))

This OPTIONAL Notification Content attribute contains the human consumable text message (see section 7.1.1) that describes the event and is represented in the text format specified by the "notify-format" attribute. Whether or not this attribute is defined depends on the delivery method definition document and MUST be defined only for use in the Machine Consumable format. If this attribute is defined for use in the Machine Consumable Notification, then whether this attribute is REQUIRED or OPTIONAL also depends on the delivery method definition document.

7.3 Additional Notification content attributes for Job events only

Table 5 lists the additional attributes that are included only in Job event Notifications and indicates the Printer support required for conformance: "R" indicates REQUIRED, "O" indicates OPTIONAL, and "CR" indicates CONDITIONALLY REQUIRED, i.e., REQUIRED in a Notification if the corresponding Job attributes are supported. Some events do not include all of these attributes as shown in Table 5.

Table 5 - Additional Notification content attributes for Job events only

Attributes	Reference	Events

		'job-progress'	'job-completed'	all others
job-state (type1 enum)	mod 4.3.7	R	R	R
job-state-reasons (1setOf type2 keyword)	mod 4.3.8	R	R	R
job-k-octets (integer(0:MAX))	mod 4.3.17.1	O	O	
job-k-octets-processed (integer(0:MAX))	mod 4.3.18.1	O	O	
job-impressions (integer(0:MAX))	mod 4.2.17.2	O	O	
job-impressions-completed (integer(0:MAX))	mod 4.3.18.2	CR	CR	
impressions-completed-current-copy (integer(0:MAX))	[ipp-prog]	R		
job-media-sheets (integer(0:MAX))	mod 4.3.17.3	O	O	
job-media-sheets-completed (integer(0:MAX))	mod 4.3.18.3	CR	CR	
job-collation-type (type2 enum)	[ipp-prog]	R		
sheet-completed-copy-number (integer(0:MAX))	[ipp-prog]	R		
sheet-completed-document-number(integer(0:MAX))	[ipp-prog]	R		

7.4 Additional Notification content attributes for Printer events only

Table 6 lists the additional attributes that are included only in Printer event Notifications and indicates the Printer support required for conformance:

Table 6 - Additional Notification content attributes for Printer events only

Attributes	Reference	Events
		all printer events

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

8 Operations for notification

This section defines all of the operations for notification which are summarized in Table 1 in section 1.1, including the assignment of the operation-id.

8.1 Operations for Per-Job Subscriptions only

This section defines the operation requests and responses that are related to Per-Job subscriptions and its Subscription object. Section **8.3 defines the REQUIRED operation requests and responses associated** with the REQUIRED Per-Printer subscription and its Subscription object.

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

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 member attributes listed in Table 7 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 "job-notify" attribute in the create operation, there is no subscription made (either implicitly or explicitly).

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 7 - Member attributes of the "job-notify" collection operation attribute

Member attribute of "job-notify" collection	Reference	client MUST supply	Printer support
notify-recipient (uri)	5.1	yes	REQUIRED
notify-events (1setOf type2 keyword)	5.2	no	REQUIRED
notify-format (mimeMediaType)	5.3	no	REQUIRED
subscriber-user-data (octetString(63))	5.4	depends on delivery method **	REQUIRED
notify-charset (charset)	5.5	no	OPTIONAL
notify-natural-language (naturalLanguage)	5.6	no	OPTIONAL

Table 8 shows the "job-notify" (collection) member attributes with their corresponding "xxx-default" and "xxx-supported" attributes. The Printer uses the "xxx-default" values if the client omits the member attribute in the request. The "xxx-supported" attributes are used by the Printer to validate the request. The client can query to determine supported values.

Table 8 - "job-notify" supported and default attributes

"job-notify" member attribute	"xxx-default", if any	"xxx-supported"
notify-recipient (uri)	none	notify-schemes-supported (1setOf uriScheme)
notify-events (1setOf type2 keyword)	notify-events-default (1setOf	notify-events-supported (1setOf type2 keyword) max-events-supported

	type2 keyword)	(integer(5:MAX))
notify-format (mimeMediaType)	none	notify-format-supported (1setOf mimeMediaType)
subscriber-user-data (octetString(63))	none	none
notify-charset (charset)	charset-configured (charset)	charset-supported (1setOf charset)
notify-natural-	natural-language-	generated-natural-language-

"job-notify" member attribute	"xxx-default", if any	"xxx-supported"
language (naturalLanguage)	configured (naturalLanguage)	supported (1setOf naturalLanguage)

See the referenced sections for a definition of these operation attributes, since they are copied to the Subscription object as the Subscription Description attributes described in section 5.

The following rules apply to Per-Job subscriptions created as part of the Job Creation operations:

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 Create-Printer-Subscription operation (section 8.2.1) for an explicit operation to subscribe for job and/or printer events independently of any particular job submission, i.e., Per-Printer subscriptions.

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. **** The client MUST supply the "subscriber-user-data"** (octetString(63)) attribute, if the definition of the delivery method specified in the "notify-recipient" attribute REQUIRES the client to supply it.

8. **If the OPTIONAL "notify-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].

9. **If the OPTIONAL "notify-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].

10. **If a collection contains other unrecognized, unsupported member attributes and/or conflicting values, the attribute returned in the** Unsupported Group is a collection containing the unrecognized, unsupported member attributes, and/or conflicting values. The Printer MUST return the unrecognized member attributes with the out-of-band value of 'unsupported'. The Printer MUST return the unsupported member attributes and conflicting values with their unsupported values. See [ipp-coll].

11. **If the number of events supplied in the "notify-events" attribute exceeds the Printer's "max-events-supported" attribute, the Printer** MUST accept the request with the status code 'successful-ok-ignored-or-substituted-attributes' and return the "job-notify" collection in the Unsupported Attributes Group with only the "job-events" member attribute containing the events that exceed the maximum.

12. **If the Per-Job subscriptions would exceed the limit of Per-Job subscriptions supported per job as specified by the Printer's "max-**job-subscriptions-supported" attribute or would exceed some implementation-defined limit on the total number of Per-Job subscriptions for the Printer (if any), the Printer MUST accept the request with the status code 'successful-ok-ignored-subscriptions', MUST return the "job-notify" attribute in the Unsupported Attributes Group with only the collection value(s) that represent the excess subscriptions that are being ignored, and MUST perform the Job Creation operation (see section 8.1.1), since the job can still be printed.

13. **If the job is accepted and one or more subscriptions are ignored, the status code returned is 'successful-ok-ignored-subscriptions.** This status code is returned even if other job attributes are unsupported or in conflict. That is, if an IPP Printer finds a warning that would allow it to return 'successful-ok-ignored-subscriptions' and either 'successful-ok-ignored-or-substituted-attributes' and/or 'successful-ok-conflicting-attributes', it must return 'successful-ok-ignored-subscriptions'. In other words, the precedence for returning success codes is: 'successful-ok-ignored-subscriptions', 'successful-ok-conflicting-attributes', and 'successful-ok-ignored-or-substituted-attributes'.

8.1.2Create-Job-Subscription operation

The OPTIONNAL Create-Job-Subscription operation creates a Per-Job Subscription object . The client can specify one or more job and/or printer events to be delivered as notifications to one Notification Recipient. For the Per-Job subscription objects, the Job events are for this job only. The printer events are any events generated by that Printer for any job or when no job is involved at all (same as for Per-Job Subscriptions).

The Printer returns a subscription id and the length of time for which it has granted a lease for the subscription.

A client can unsubscribe using the Cancel-Subscription operation (section 8.3.4) and the subscription id.

Two Create-Job-Subscription operations with the same events and same Notification Recipient MUST be kept as distinct subscriptions and be assigned distinct subscription ids. A Printer MUST allow such duplicate subscriptions such that Cancel-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 object.

If the Printer has a bounded set of concurrent Per-Job subscriptions and the request would exceed that bound, the Printer rejects the operation and returns the 'client-error-too-many-subscriptions' status code. The client SHOULD try again later.

Access Rights: To create Per-Job subscription objects, the authenticated user (see [IPP-MOD] section 8.3) performing this operation MUST either be the job owner or have operator or administrator access rights for the Printer object (see [IPP-MOD] sections 1 and 8.5). Otherwise the IPP object MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' status code as appropriate.

Request:

Group 1: Operation Attributes	Reference [ipp-mod]	client MUST supply	Printer support
"attributes-charset" (charset)	3.1.4.1	yes	REQUIRED
"attributes-natural-language"	3.1.4.1	yes	REQUIRED

(naturalLanguage)			
"printer-uri" (uri)	3.1.5	yes	REQUIRED
"job-id" (integer(1:MAX))	3.1.5	yes	REQUIRED
"job-uri" (uri)	3.1.5	if not "printer-uri" and "job-id"	REQUIRED
"requesting-user-name" (name(MAX))	8.3	RECOMMENDED	REQUIRED

Group 2: Subscription Attributes	Reference	client MUST supply	Printer support
"notify-recipient" (uri)	5.1	yes	REQUIRED
"notify-events" (1setOf type2 keyword)	5.2	no	REQUIRED
"notify-format" (mimeMediaType)	5.3	no	REQUIRED
"subscriber-user-data" (octetString(63))	5.4	depends on delivery method	REQUIRED
"notify-charset" (charset)	5.5	no	OPTIONAL
"notify-natural-language" (naturalLanguage)	5.6	no	OPTIONAL
"notify-persistence-requested" (boolean)	see below	no	OPTIONAL

Response:

Group 1: Operation Attributes	Reference [ipp-mod]	REQUIRED in response
"status-code" (type2 enum)	3.1.6.1	REQUIRED
"attributes-charset" (charset)	3.1.4.2	REQUIRED
"attributes-natural-language" (naturalLanguage)	3.1.4.2	REQUIRED
"status-message" (text(255))	3.1.6.2	OPTIONAL
"detailed-status-message" (text(MAX))	3.1.6.3	OPTIONAL

Group 2: Unsupported Attributes (see [ipp-mod] section 3.1.7)	REQUIRED if unsupported attributes

Group 3: Subscriptions Attributes	Reference	REQUIRED in response
"subscription-id" (integer(1:MAX))	5.8	REQUIRED
"notify-persistence-granted" (boolean)	5.13	REQUIRED

Attribute Notes:

"job-id" (integer(1:MAX)) - the client MUST either (1) supply this attribute, in combination with the "printer-uri" attribute, in order to create a Per-Job subscription for the Job identified by the "job-id" value or supply the "job-uri" instead, as with any Job operation:

"job-uri" (uri) - the client MUST supply this attribute, if the "printer-uri" and "job-id" are not supplied, in order to identify the Job to which the Per-Job Subscription is being created.

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

"notify-event" (1setOf type2 keyword) - if the client does not supply this attribute, the Printer populates the "notify-events" Subscription Description attribute from its "notify-events-default" Printer Description attribute.

"notify-format" (mimeMediaType) - if the client supplies this attribute, the value indicates which Human Consumable text format is requested for use in the Notification using the delivery method that the client supplies in the "notify-recipient" attribute. If the client does not supply this attribute, the Machine-Consumable form of the delivery method that the client supplies in the "notify-recipient" attribute is used.

"subscriber-user-data" (octetString(63)) - the client MUST supply this attribute, if the definition of the delivery method specified in the "notify-recipient" attribute REQUIRES the client to supply it.

"notify-persistence-requested" (boolean) - whether or not the Per-Job Subscription is to be persistent, i.e., saved across power cycles. Note: Persistent trap registrations is a client option in SNMPv3 [RFC2573].

"notify-persistence-granted" (boolean) - whether or not this 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 Operations for Per-Printer Subscriptions only

This section defines the operation requests and responses associated with the Per-Printer subscription and its Subscription object.

8.2.1 Create-Printer-Subscription operation

The REQUIRED Create-Printer-Subscription operation creates a Per-Printer Subscription object . The client can specify one or more job and/or printer events to be delivered as notifications to one Notification Recipient. For the Per-Printer subscription objects, the job events are for any job submitted to the Printer. The printer events are any events generated by that Printer for any job or when no job is involved at all.

The Printer returns a subscription id and the time at which the subscription lease expires (which may be earlier or later than the client requested).

The client MUST renew the Per-Printer subscription using the Renew-Subscription operation (see section 8.3.3) before the lease runs out in order to maintain the subscription. A client can unsubscribe using the Cancel-Subscription operation (section 8.3.4) and the subscription id.

Two Create-Printer-Subscription operations with the same events and same Notification Recipient MUST be kept as distinct subscriptions and be assigned distinct subscription ids. A Printer MUST allow such duplicate subscriptions such that Cancel-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 Per-Printer subscriptions and the request would exceed that bound, the Printer rejects the operation and returns the 'client-error-too-many-subscriptions' status code. The client SHOULD try again later.

Access Rights: To create Per-Printer subscription objects, the authenticated user performing this operation MUST have Per-Printer subscription rights for this Printer. Otherwise the IPP object MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' status code as appropriate.

Request:

Group 1: Operation Attributes	Reference [ipp-mod]	client MUST supply	Printer support
"attributes-charset" (charset)	3.1.4.1	yes	REQUIRED
"attributes-natural-language" (naturalLanguage)	3.1.4.1	yes	REQUIRED
"printer-uri" (uri)	3.1.5	yes	REQUIRED
"requesting-user-name" (name(MAX))	8.3	RECOMMENDED	REQUIRED

Group 2: Subscription Attributes	Reference	client MUST supply	Printer support
"notify-recipient" (uri)	5.1	yes	REQUIRED
"notify-events" (1setOf type2 keyword)	5.2	no	REQUIRED
"notify-format" (mimeMediaType)	5.3	no	REQUIRED
"subscriber-user-data" (octetString(63))	5.4	depends on the delivery method	REQUIRED
"notify-charset" (charset)	5.5	no	OPTIONAL
"notify-natural-language" (naturalLanguage)	5.6	no	OPTIONAL
"notify-lease-time-requested" (integer(0:MAX))	see below	no	REQUIRED
"notify-persistence-requested" (boolean)	see below	no	OPTIONAL

Response:

Group 1: Operation Attributes	Reference	REQUIRED in response
"status-code" (type2 enum)	3.1.6.1	REQUIRED
"attributes-charset" (charset)	3.1.4.2	REQUIRED
"attributes-natural-language" (naturalLanguage)	3.1.4.2	REQUIRED
"status-message" (text(255))	3.1.6.2	OPTIONAL
"detailed-status-message" (text(MAX))	3.1.6.3	OPTIONAL

Group 2: Unsupported Attributes (see [ipp-mod] section 3.1.7)	REQUIRED if unsupported attributes

Group 3: Subscription Attributes	Reference	REQUIRED in response
"subscription-id" (integer(1:MAX))	5.8	REQUIRED
"notify-lease-expiration-time" (integer(0:MAX))	5.9	REQUIRED
"notify-server-up-time" (integer(1:MAX))	5.12	REQUIRED

```
     "notify-persistence-granted" (boolean)    5.13      REQUIRED
```

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 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" Printer Description attribute value (see section 6.9).

"notify-event" (1setOf type2 keyword) - if the client does not supply this attribute, the Printer populates the "notify-events" Subscription Description attribute from its "notify-events-default" Printer Description attribute.

"notify-format" (mimeMediaType) - if the client supplies this attribute, the value indicates which Human Consumable text format is requested for use in the Notification using the delivery method that the client supplies in the "notify-recipient" attribute. If the client does not supply this attribute, the Machine-Consumable form of the delivery method that the client supplies in the "notify-recipient" attribute is used.

"subscriber-user-data" (octetString(63)) - the client MUST supply this attribute, if the definition of the delivery method specified in the "notify-recipient" attribute REQUIRES the client to supply it.

"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].

"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 Printer will delete the Subscription. A value of 0 indicates that the lease subscription will never expire.

"notify-server-up-time" (integer(1:MAX)) - The Printer object MUST return this attribute which is an alias for the Printer's "printer-up-time" Printer Description attribute. The client subtracts this value from the "notify-lease-expiration-time" value returned in order to determine the number of second in the future that the subscription will expire. This computed value 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 (see section 6.8).

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

8.3 Common Operations for Per-Job and Per-Printer Subscriptions

This section defines the operations that are common to both Per-Job and Per-Printer subscriptions.

8.3.1 Get-Subscription-Attributes operation

The REQUIRED Get-Subscription-Attributes returns the requested attributes of the identified Subscription object. See section 5.

Request:

Group 1: Operation Attributes	Reference [ipp-mod]	client MUST supply	Printer support
"attributes-charset" (charset)	3.1.4.1	yes	REQUIRED
"attributes-natural-language" (naturalLanguage)	3.1.4.1	yes	REQUIRED
"printer-uri" (uri)	3.1.5	yes	REQUIRED
"requesting-user-name" (name(MAX))	8.3	RECOMMENDED	RECOMMENDED

Group 2: Subscription Attributes	Reference	client MUST supply	Printer support
"subscription-id" (integer(1:MAX))	5.8	yes	REQUIRED
"requested-attributes" (1setOf type2 keyword)	see below	no	REQUIRED

Response:

Group 1: Operation Attributes	Reference [ipp-mod]	REQUIRED in response
"status-code" (type2 enum)	3.1.6.1	REQUIRED
"attributes-charset" (charset)	3.1.4.2	REQUIRED
"attributes-natural-language" (naturalLanguage)	3.1.4.2	REQUIRED
"status-message" (text(255))	3.1.6.2	OPTIONAL
"detailed-status-message (text(MAX))	3.1.6.3	OPTIONAL

Group 2: Unsupported Attributes (see [ipp-mod] section 3.1.7)	REQUIRED if unsupported attributes
Group 3: <the requested Subscription object attributes>	REQUIRED

This operation is similar to the Get-Printer-Attributes operation.

"requested-attributes" (1setOf keyword): The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute.

It is a set of Subscription attribute names and/or the 'all' attribute group name in whose values the requester is interested. This set of attributes is returned for each Job object that is returned. If the client does not supply this attribute, the Printer MUST respond as if the client had supplied this attribute with the value: 'all'.

8.3.2 Get-Subscriptions operation

The REQUIRED Get-Subscriptions operation returns the requested attributes of the Subscription objects (see section 5).

Request:

Group 1: Operation Attributes	Reference [ipp-mod]	client MUST supply	Printer support
"attributes-charset" (charset)	3.1.4.1	yes	REQUIRED
"attributes-natural-language" (naturalLanguage)	3.1.4.1	yes	REQUIRED
"printer-uri" (uri)	3.1.5	yes	REQUIRED
"job-id" (integer(1:MAX))	3.1.5	yes, if Per-Job Subscription	REQUIRED
"requesting-user-name" (name(MAX))	8.3	RECOMMENTED	RECOMMENDED
"limit" (integer(1:MAX))	see below	OPTIONAL	REQUIRED
"requested-attributes" (1setOf type2 keyword)	see below	OPTIONAL	REQUIRED
"my-subscriptions" (boolean)	see below	OPTIONAL	REQUIRED

Response:

Group 1: Operation Attributes	Reference [ipp-mod]	REQUIRED in response
"status-code" (type2 enum)	3.1.6.1	REQUIRED
"attributes-charset" (charset)	3.1.4.2	REQUIRED
"attributes-natural-language" (naturalLanguage)	3.1.4.2	REQUIRED
"status-message" (text)	3.1.6.2	OPTIONAL
"detailed-status-message" (text(MAX))	3.1.6.3	OPTIONAL

Group 2: Unsupported Attributes (see [ipp-mod] section 3.1.7)	REQUIRED if unsupported attributes
Group 3 to N:<the requested Subscription Attributes for each Subscription object in a separate group>	see below REQUIRED

Attribute 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'.

"job-id" (integer(1:MAX)) - If the client supplies this attribute, all of the Per-Job Subscription objects for the identified job are candidates for return. It this attribute is omitted, all of the Per-Printer Subscription objects are candidates for return.

"limit" (integer(1:MAX)): The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It is an integer value that determines the maximum number of Subscription objects that a client will receive from the Printer even if "my-subscriptions" constrain which subscriptions are returned. The limit is a "stateless limit" in that if the value supplied by the client is 'N', then only the first 'N' Subscriptions are returned in the Get-Subscriptions Response. There is no mechanism to allow for the next 'M' Subscriptions after the first 'N' Subscriptions. If the client does not supply this attribute, the Printer object responds with all applicable Subscriptions.

"requested-attributes" (1setOf keyword): The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It is a set of Subscription attribute names and/or the 'all' attribute group name in whose values the requester is interested. This set of attributes is returned for each Job object that is returned. If the client does not supply this attribute, the Printer MUST respond as if the client had supplied this attribute with the value: 'all'.

"my-subscriptions" (boolean): The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It indicates whether Subscriptions from all users or just the Subscriptions submitted by the requesting user of this request MUST be returned by the Printer object. If the client does not supply this attribute, the Printer object MUST respond as if the client had supplied the attribute with a value of 'false', i.e., Subscriptions from all users. The means for authenticating the requesting user and matching the Subscription objects is the same as for jobs as described in [ipp-mod] section 8.

Groups 3 to N: Subscription Object Attributes: The Printer object responds with one set of Subscription Object Attributes for each returned Subscription object. The Printer object ignores (does not respond with) any requested attribute or value which is not supported or which is restricted by the security policy in force, including whether the requesting user is the user that created the Subscription object (subscribing user) or not (see [ipp-mod] section 8).

8.3.3 Renew-Subscription operation

The REQUIRED Renew-Subscription operation permits a client to request the IPP Printer to extend the lease on a Subscription object instance. There is no way to renew a Per-Job subscription, since they are automatically canceled after the job completes and no longer has any documents, i.e., the job is no longer retained (see [ipp-mod] section 4.3.7.2). If the requested subscription object is a Per-Job subscription, the Printer MUST grant an infinite lease by returning a 0 value for the "notify-lease-expiration-time".

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

Request:

Group 1: Operation Attributes	Reference [ipp-mod]	client MUST supply	Printer support
"attributes-charset" (charset)	3.1.4.1	yes	REQUIRED
"attributes-natural-language" (naturalLanguage)	3.1.4.1	yes	REQUIRED
"printer-uri" (uri)	3.1.5	yes	REQUIRED
"requesting-user-name" (name(MAX))	8.3	RECOMMENDED	RECOMMENDED

Group 2: Subscription Attributes	Reference	client MUST supply	Printer support
"subscription-id" (integer(1:MAX))	5.8	yes	REQUIRED
"notify-lease-time-requested" (integer(0:MAX))	see below	no	REQUIRED

Response:

Group 1: Operation Attributes	Reference	REQUIRED

	[ipp-mod]	in response
"status-code" (type2 enum)	3.1.6.1	REQUIRED
"attributes-charset" (charset)	3.1.4.2	REQUIRED
"attributes-natural-language" (naturalLanguage)	3.1.4.2	REQUIRED
"status-message" (text(255))	3.1.6.2	OPTIONAL
"detailed-status-message" (text(MAX))	3.1.6.3	OPTIONAL

Group 2: Unsupported Attributes (see [ipp-mod] section 3.1.7)		REQUIRED if unsupported attributes
Group 3: Subscriptions Attributes	Reference	REQUIRED in response
"notify-lease-expiration-time" (integer(0:MAX))	5.9	REQUIRED
"notify-server-up-time" (integer(1:MAX))	5.12	REQUIRED

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 8.2.1).

"notify-lease-expiration-time" - the time in the future when the subscription will expire. Same as for the Create-Printer-Subscription operation (see section 8.2.1).

"notify-server-up-time" - the Printer object MUST return this attribute which is an alias for the Printer's "printer-up-time" Printer Description attribute. The client subtracts this value from the "notify-lease-expiration-time" value returned in order to determine the number of second in the future that the subscription will expire (see further explanation in section 8.2.1).

Note: There is no way to change any of the Subscription object attributes, except the "notify-lease-expiration-time" attribute (using the Renew-Subscription operation). In order to change other attributes, a client can create a new Subscription object and then use the Cancel-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).

Note: There is no need to renew a Per-Job Subscription, since it is effectively the time that the Job is active (see section 8.1).

8.3.4 Cancel-Subscription operation

The REQUIRED Cancel- Subscription operation allows a client to remove a Subscription object. 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 have operator or administrator access rights for the Printer object (see [IPP-MOD] sections 1 and 8.5). Otherwise the IPP object

MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' status code as appropriate.

Request:

Group 1: Operation Attributes	Reference [ipp-mod]	client MUST supply	Printer support
"attributes-charset" (charset)	3.1.4.1	yes	REQUIRED
"attributes-natural-language" (naturalLanguage)	3.1.4.1	yes	REQUIRED
"printer-uri" (uri)	3.1.5	yes	REQUIRED
"requesting-user-name" (name(MAX))	8.3	RECOMMENDED	RECOMMENDED

Group 2: Subscription Attributes	Reference	client MUST supply	Printer support
"subscription-id" (integer(1:MAX))	5.8	yes	REQUIRED

Response:

Group 1: Operation Attributes	Reference [ipp-mod]	REQUIRED in response
"status-code" (type2 enum)	3.1.6.1	REQUIRED
"attributes-charset" (charset)	3.1.4.2	REQUIRED
"attributes-natural-language" (naturalLanguage)	3.1.4.2	REQUIRED
"status-message" (text(255))	3.1.6.2	OPTIONAL
"detailed-status-message" (text(MAX))	3.1.6.3	OPTIONAL

Group 2: Unsupported Attributes (see [ipp-mod] section 3.1.7)	REQUIRED if unsupported attributes

9 Comparison of Per-Job versus Per-Printer Subscriptions

Per-Job and Per-Printer subscriptions are quite similar. Either type of subscription can subscribe to Job events, Printer events, or both. Both types of subscriptions can be queried using the Get-Subscriptions and Get-Subscription-Attributes operations and canceled using the Cancel-Subscription operation. Both types of subscriptions create Subscription objects which have the same attributes defined. However, there are some semantic differences between Per-Job subscriptions and Per-Printer subscriptions. A Per-Job Submission Subscription is established by the client when submitting a job and after creating the job using the Create-Job-Subscription operation by specifying the "job-id" of the job. A Per-Printer Subscription is established between a client and a

Printer using the Create-Printer-Subscription operation. Some specific differences are:

1. A client usually creates a Per-Job subscription as part of the Job Creation operations (Create-Job, Print-Job, and Print-URI), rather than using the OPTIONAL Create-Job-Subscription operation, especially since Printer implementations NEED NOT support the Create-Job-Subscription operation, since it is OPTIONAL.

2. For Per-Job subscriptions, the subscription is only valid while the job is "not-complete" (see sections 7.2 and 7.3) while for the Per-Printer subscriptions, the subscription is valid until the time (in seconds) that the Printer returned in the "notify-lease-expiration-time" operation attribute expires.

3. Job Events in a Per-Job subscription apply only to "one job" (the Job created by the job creation operation or references by the Create-Job-Subscription operation) while Job Events in a Per-Printer subscription apply to ALL jobs contained in the IPP Printer object.

10 Out of Band Values

This section defines out-of-band values (see [ipp-mod] section 4.1) for use with attributes defined in this and other documents.

10.1'none'

This out-of-band value permits a client to indicate in a request that a specified attribute in the request MUST NOT be applied to the job. Specifically, this value overrides the Printer's "xxx-default" attribute value for the attribute, if one exists. This out-of-band value is needed since attributes that are of the 'collection', 'uri', and 'mimeMediaType' type need a way for a client to specify that the Printer MUST NOT apply the value of an "xxx-default" attribute to the job. See [ipp-coll] for a complete specification, including encoding.

11 Conformance Requirements

It is OPTIONAL to implement this Event Notification specification. An implementation conforming to this specification MUST also meet the Conformance Requirements detailed in [IPP-MOD] section 5. If this Event Notification specification is implemented, IPP objects MUST support all of the REQUIRED object attributes as defined in this document in the indicated sections:

1. REQUIRED Subscription object attributes in section 5.

2. REQUIRED Printer Description object attributes in section 6.

3. REQUIRED attributes in Notification content in section 7.

In addition, the Printer MUST support the 'collection' attribute syntax (see [ipp-coll]).

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.

If this Event Notification specification is implemented, the operations described in section 8 MUST be supported as described in Table 9:

Table 9 - Conformance Requirements for Operations

Attribute	Conformance requirements
"job-notify" (1setOf collection) in Job Creation operations (section 8.1.1)	REQUIRED
Create-Printer-Subscription (section 8.2.1)	REQUIRED
Create-Job-Subscription (section 8.1.2)	OPTIONAL
Get-Subscription-Attributes (section 8.3.1)	REQUIRED
Get-Subscriptions (section 8.3.2)	REQUIRED
Renew-Subscription (section 8.3.3)	REQUIRED
Cancel-Subscription (section 8.3.4)	REQUIRED

12 IANA Considerations

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

13 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.

14 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.

15 Status Codes

The following status codes are defined as extensions for notification:

15.1'successful-ok-ignored-subscriptions' (0x0003)

The number of subscriptions supplied in a Job Creation operation (Create-Job, Print-Job, Print-URI) exceeds either the limit of Per-Job subscriptions supported per job as specified by the Printer's "max-job-subscriptions-supported" attribute or some implementation-defined limit on the total number of Per-Job subscriptions for the Printer (if any). The Printer MUST accept the request with this status code, MUST return the "job-notify" attribute in the Unsupported Attributes Group with only the collection value(s) that represent the excess subscriptions that are being ignored, and MUST perform the Job Creation operation (see section 8.1.1), since the job can still be printed.

This status code is returned even if other job attributes are unsupported or in conflict. That is, if an IPP Printer finds a warning that would allow it to return 'successful-ok-ignored-subscriptions' and either 'successful-ok-ignored-or-substituted-attributes' and/or 'successful-ok-conflicting-attributes', it must return 'successful-ok-ignored-subscriptions'. In other words, the precedence for returning success codes is: 'successful-ok-ignored-subscriptions', 'successful-ok-conflicting-attributes', and 'successful-ok-ignored-or-substituted-attributes'.

15.2client-error-uri-notification-scheme-not-supported (0x0414)

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

There is no corresponding Per-Job subscription error for a Job Creation operation, since the Printer object MUST ignore any errors in the "job-notify" operation attribute, MUST return the "notify-recipient" attribute in the Unsupported Attributes Group, and perform the Job Creation operation (see section 8.1.1), since the job can still be printed.

15.3client-error-too-many-subscriptions (0x0415)

The bounded set of concurrent Per-Printer subscriptions supported by the Printer object would be exceeded if this Create-Printer-Subscription request were accepted or the bounded set of concurrent Per-Job subscriptions supported by the Printer object would be exceeded if this Create-Job-Subscriptions request were accepted.

Note: There is no corresponding Per-Job subscription error on a Job Creation operation, since the Printer object MUST ignore any errors in the "job-notify" operation attribute and perform the Job Creation operation (see section 8.1.1), since the job can still be printed.

15.4client-error-too-many-events (0x0416)

The client supplied more events in the "notify-events" operation attribute in a Create-Job-Subscription or Create-Printer-Subscription operation than the Printer supports, as indicated in its "max-events-supported" attribute (see section 6.4).

There is no corresponding Per-Job subscription error for a Job Creation operation, since the Printer object MUST ignore any errors in the "job-notify" operation attribute, MUST return the "notify-events" attribute in the Unsupported Attributes Group with only the excess events that are being ignored, and perform the Job Creation operation (see section

8.1.1), since the job can still be printed.

16 Addition attribute tag encodings

The Subscription attributes tag and the Notification attributes tag are assigned the following delimiter tag values as extensions to the encoding defined in [ipp-pro]) for use in requests and responses to delimit Subscription object attribute Groups and Notification Content attribute Groups:

Tag Value (Hex)	Delimiter
0x06	subscription-attributes-tag
0x07	notification-attributes-tag

17 References

[ipp-coll]
deBry, R., , Hastings, T., Herriot, R., "Internet Printing Protocol/1.0 & 1.1: collection attribute syntax", <draft-ietf-ipp-collection-01.doc>, work in progress, February 22, 2000.

[ipp-mod]
deBry, R., , Hastings, T., Herriot, R., Isaacson, S., Powell, P., "Internet Printing Protocol/1.1: Model and Semantics", < draft-ietf-ipp-model-v11-06.txt>, work in progress, March 1, 2000.

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

[ipp-method-indp]
Parra, H., Hastings, T., "Internet Printing Protocol (IPP): The INDP Event Notification Delivery Method", <draft-ietf-indp-method-00.txt>, work in progress, March 8, 2000.

[ipp-method-ipp]
Manros, C., Hastings, T., Herriot, R., Lewis, H., "Internet Printing Protocol (IPP): The 'ipp' Notification Delivery Polling Method", <draft-ietf-ipp-notify-poll-00.txt>, work in progress, March 8, 2000.

[ipp-method-mailto]
Holst, H. and Hastings, T., "Internet Printing Protocol (IPP): The 'mailto' Notification Delivery Method", <draft-ietf-ipp-notify-mailto-00.txt>, work in progress, March 9, 2000.

[ipp-method-snmp]
McDonald, I., Hastings, T., "Internet Printing Protocol (IPP): Notifications over SNMP", <draft-ietf-ipp-not-over-snmp-01.txt>, work in progress, December 1, 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-pro]
Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and Transport", <draft-ietf-ipp-protocol-v11-05.txt>, work in progress, March 1, 2000.

[ipp-prog]
Hastings, T., Bergman, R., Lewis, H., "Proposed Job Progress Attributes for IPP", <draft-ietf-ipp-job-prog.txt> work in progress, February 2, 2000.

[ipp-set]
Kugler, C., , Hastings, T., Herriot, R., Lewis, H, "Internet Printing Protocol (IPP): Job and Printer Set Operations", <draft-ietf-ipp-job-printer-set-ops-01.txt>, work in progress, March 8, 2000.

[ipp-set2]
Kugler, C., , Hastings, T., Lewis, H, "Internet Printing Protocol (IPP): Additional Operations, Set 2", <draft-ietf-ipp-ops-set2.txt>, work in progress, February 3, 2000.

[RFC1866]
Berners-Lee, T., Connolly, D., "Hypertext Markup Language - 2.0", RFC 1866, November 1995.

[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

[RFC2279]
F. Yergeau , "UTF-8, a transformation format of ISO 10646", RFC 2279. January 1998.

[RFC2302]
Parsons, G., et. al., "Tag Image File Format (TIFF) - image/tiff", RFC 2302, March 1998.

[RFC2376]
Whitehead, E., Murata, M., "XML Media Types", RFC 2376, July 1998.

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

[RFC2567]
Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.

[RFC2568]
Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol", RFC 2568, April 1999.

[RFC2569]
Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC 2569, April 1999.

18 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)
Hitachi Koki Imaging Solutions
1757 Tapo Canyon Road
Simi Valley, CA 93063-3394

Phone: 805-578-4421
Fax: 805-578-4001
Email: rbergma@hitachi-hkis.com

A. Appendix: Change History

This section summarizes the changes to the document. Each sub-section is in reverse chronological order. Adding or removing ISSUES that don't change the document are not listed here.

18.1Changes to the March 6, 2000 version to create the March 8, 2000 version

The following changes were made to the March 6, 2000 version to create the March 8, 2000 version based on the agreements reached on the mailing list:

1. Changed the name of the SNMP delivery method from 'snmp' to 'snmpnotify', since the Notification Recipient isn't an SNMP agent.

2. Clarified that an implementation with only a single value for persistent-jobs-supported (boolean) or persistent-subscriptions-supported (boolean) MAY make it settable to the single value or make it not-settable.

18.2Changes to the February 2, 2000 version to create the March 6, 2000 version

The following changes were made to the February 2, 2000 version to create the March 6, 2000 version based on the agreements reached on the mailing list, at the February IPP WG meetings, and reflected in the minutes:

1. Clarified that this extension is intended as an extension to IPP/1.0, IPP/1.1, and future versions.

2. Allocated the operation-id 0x0016 to 0x001B values for the Notification operations defined in the document.

3. Pre-pended the word "subscription-" on the front of the "request-id" Subscription object attribute to distinguish it from the "request-id" parameter that is sent in every request and response.

4. Added the term "settable" for describing attributes that are not READ-ONLY.

5. Added the term "Subscription Creation operation" to stand for any operation that can create a Subscription object: Job Creation operations (Create-Job, Print-Job, and Print-URI), Create-Job-Subscription, and Create-Printer-Subscription.

6. Changed the "subscriber-user-name" (name(MAX)) Subscription object attribute from OPTIONAL to REQUIRED.

7. Changed the name and semantics of "notify-printer-up-time(integer(1:MAX)) to notify-server-up-time so that it can be

either the Printer's uptime or a Notification Delivery Service uptime.

8. Added the 'ipp:', 'indp:', 'mailto:, and 'snmp:' notification delivery schemes to the definition of the "notify-recipients" to indicate possible schemes.

9. Changed the name and semantics of "notify-text-format" (mimeMediaType) to "notify-format" so that it can be used to specify either Human Consumable or Machine Consumable formats where the implementation supports both. Clarified that this attribute controls whatever variable Notification Content that the implementation supports, which may be an attachment to the fixed content format or the contents of the "human-readable-report" (text(MAX)) attribute. Clarified that an implementation NEED NOT support all of its supported Notification Content formats with all of its supported delivery methods.

10. **Added 'text/xml', 'application/ipp', 'application/postscript', and** 'image/tiff' and additional example MIME media types for "notify-format" (mimeMediaType).

11. **Clarified that the recommend way for a client to determine whether or not a Printer supports Per-Job Subscriptions is to query the** Printer's "max-job-subscriptions-supported" attribute, since Create-Job-Subscriptions is an OPTIONAL operation.

12. **Clarified that the recommend way for a client to determine whether or not a Printer supports Per-Printer Subscriptions is to query the** Printer's "operations-supported" attribute to see if the Create-Printer-Subscriptions operations is supported, since this is the usual way to determine a Printer's capabilities.

13. **Clarified that if "persistent-jobs-supported" (boolean) and** "persistent-subscriptions-supported" (boolean) are settable, then setting them must affect whether or not jobs and subscriptions are persistent.

14. **Allowed delivery methods to send operations with or without a response, depending on the definition of the delivery method.**

15. **Indicated that a deliver method definition is free to REQUIRE that the client supply the "subscriber-user-data" attribute.**

16. **Required that the Printer support the "job-uri" operation attribute as a target, in addition to "printer-uri"&"job-id", i.e., keep** consistent with all Job operations.

17. **Changed the 'none' out-of-band value to be a reference to the collection document [ipp-coll], since the use for it in this document** is with the 'collection' attribute syntax.

18. **Clarified that a conforming implementation MUST support the** 'collection' attribute syntax, since that is required in Job Creation operations.

19. **Allocated the values to the new status codes defined in this document.**

20. **Allocated the [ipp-pro] subscription-attributes-tag and notification-attributes-tag delimiter tags to delimit Subscription** attributes and Notification Content attributes in requests and responses.

21. **Changed the 'server-error-too-many-subscriptions' and 'server-error-too-many-events' to be client errors, i.e., 'client-error-too-**many-subscriptions' and 'client-error-too-many-events', since other errors of this type are client errors.

18.3Changes to the October 14, 1999 version to create the February 2, 2000 version

The following changes were made to the October 14, 1999 version to create the February 2, 2000 version based on the agreements reached at the October and December IPP WG meetings and reflected in the minutes:

1. Added a Java Listener as an example of a Notification Recipient.

2. Clarified the object relationships in section 4.1.

3. Clarified how job events differ for Per-Job versus Per-Printer Subscriptions.

4. Added the ability for the Machine Consumable form to contain a Human Readable "human-readable-report" (text) attribute so that both forms could be sent in the same Notification.

5. Clarified that the 'none' value for notify-text-format (mimeMediaType) has to be out-of-band, not the text string 'none' as a mimeMediaType.

6. Clarified that 'none' means send the Machine Consumable form without the "human-readable-report" (text) attribute, if it is defined.

7. Clarified that Notification Recipients MUST be able to accept unrecognized attributes.

8. Allowed the notification delivery method definition to be modeled as (1) a request with an operation code without a response, (2) a request with a operation code with a response or (3) a response with a status code.

9. Added "notify-text-format" (mimeMediaType) and "human-readable-report" (text(MAX)) to be able to be sent in a Notification content, if the notification delivery method document permits it.

10. **Added "job-k-octets" (integer(0:MAX)), "job-impressions"** (integer(0:MAX)), and "job-media-sheets" (integer(0:MAX)) as OPTIONAL for Notification content for use in job-progress events to show the target values so that the Notification Recipient can show a thermometer.

11. **Added a Subscription Attributes group (and subscription-attributes tag) the Create-Job-Subscription and Create-Printer-Subscription** requests and responses.

12. **Added the 'none' out-of-band value for use with "notify-text-format" (mimeMediaType) attribute.**

13. **Changed the job progress attributes from using -2 to mean 'unknown' as in the PWG Job Monitoring MIB, to use the 'unknown' out-of-band** value.

B. Appendix: Full Copyright Statement