INTERNET-DRAFT~~—There are 7 issues highlighted like this~~
<draft-ietf-ipp-not-spec-04.txt>

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

July ~~7~~13, 2000

Internet Printing Protocol (IPP):

IPP Event Notification Specification

Status of this Memo

Abstract

This document describes an extension to the IPP/1.0, IPP/1.1, and future versions. This extension allows a client to subscribe to printing related Events. Subscriptions are modeled as *Subscription Objects*. The Subscription Object specifies that when one of the specified *Event* occurs, the Printer sends an asynchronous *Event Notification* to the specified *Notification Recipient* via the specified *Delivery Method* (i.e., protocol). A client associates Subscription Objects with a particular Job by performing the Create-Job-Subscriptions operation or by submitting a Job with subscription information. A client associates Subscription Objects with the Printer by performing a Create-Printer-Subscriptions operation. Four other operations are defined for Subscription Objects: Get-Subscriptions-Attributes, Get-Subscriptions, Renew-Subscription, and Cancel-Subscription.

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 object 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 (IPP): "Job Progress Attributes" [ipp-prog]
One or more Delivery Method Documents registered with IANA (see section 13).

Note: this document does not define any Delivery Methods, but it does define the rules for conformance for Delivery Method Documents.

Refer to the Table of Contents for the layout of this document.

1.1 Notification Overview

This document defines operations that a client can perform in order to create *Subscription Objects* in a Printer and carry out other operations on them. A Subscription Object represents a Subscription abstraction. The Subscription Object specifies that when one of the specified *Events* occurs, the Printer sends an asynchronous *Event Notification* to the specified *Notification Recipient* via the specified *Delivery Method* (i.e., protocol).

When a client (called a *Subscribing Client*) performs an operation that creates a Subscription Object, the operation contains one or more *Subscription Template Attributes Group*s. Each such group holds information used by the Printer to initialize a newly created Subscription Object. The Printer creates one Subscription Object for each Subscription Template Attributes Group in the operation. This group is like the Job Template Attributes group defined in [ipp-mod]. The following is an example of the information included in a Subscription Template Attributes Group (see section 5 for details on the Subscription Object attributes):

1. The names of Subscribed Events that are of interest to the Notification Recipient.
2. The address (URL) of one Notification Recipient.
3. The Delivery Method (i.e., the protocol) which the Printer uses to send the Event Notification.
4. Some opaque data that the Printer sends to the Notification Recipient in the Event Notification. The Notification Recipient might use this opaque data as a forwarding address for the Event Notification.
5. The charset to use in text fields within an Event Notification
6. The natural language to use in the text fields of the Event Notification
7. The requested lease time in seconds for the Subscription Object

An operation that creates a Subscription Object is called a *Subscription Creation Operation*. These operations include the following operations (see section 11.1 for further details):

- **Job Creation operation**: When a client performs such an operation (Print-Job, Print-URI, and Create-Job), a client can include zero or more Subscription Template Attributes Groups in the request. The Printer creates one Subscription Object for each Subscription Template Attributes Group in the request, and the Printer associates each such Subscription Object with the newly created Job. This document extends these operations' definitions in [ipp-mod] by adding Subscription Template Attributes Groups in the request and Subscription Attributes Groups in the response.

- **Create-Job-Subscriptions operation:** A client can include one or more Subscription Template Attributes Groups in the request. The Printer creates one Subscription Object for each Subscription Template Attributes Group and associates each with the job that is the target of this operation.

- **Create-Printer-Subscriptions operation:** A client can include one or more Subscription Template Attributes Groups in the request. The Printer creates one Subscription Object for each Subscription Template Attributes Group and associates each with the Printer that is the target of this operation.

For each of the above operations:

- the Printer associates a Subscription Object with the Printer or a specific Job. When a Subscription Object is associated with a Job Object, it is called a *Per-Job Subscription Object*. When a Subscription Object is associated with a Printer Object, it is called a *Per-Printer Subscription Object*.

- the response contains one Subscription Attributes Group for each Subscription Template Attributes Group in the request and in the same order. When the Printer successfully creates a Subscription Object, its corresponding Subscription Attributes Group contains the "notify-subscription-id" attribute. This attribute uniquely identifies the Subscription Object and is analogous to a "job-id" for a Job object. Some operations described below use the "notify-subscription-id" to identify the target Subscription Object.

This document adds the following additional operations (see section 11.2 for further details)::

- **Validate-Job operation:** When a client performs this operation, a client can include zero or more Subscription Template Attributes Groups in the request. The Printer determines if it could create one Subscription Object for each Subscription Template Attributes Group in the request. This document extends this operation's definition in [ipp-mod] by adding Subscription Template Attributes Groups in the request and Subscription Attributes Groups in the response.

- **Get-Subscription-Attributes operation:** This operation allows a client to obtain the specified attributes of a target Subscription Object.

- **Get-Subscriptions operation:** This operation allows a client to obtain the specified attributes of all Subscription Objects associated with the Printer or a specified Job.

- **Renew-Subscription operation:** This operation renews the lease on the target Per-Printer Subscription Object before it expires. A newly created Per-Printer Subscription Object receives an initial lease. It is the duty of the client to use this operation frequently enough to preserve a Per-Printer Subscription Object. The Printer deletes a Per-Printer Subscription Object when its lease expires. A Per-Job Subscription Object last exactly as long as its associated Job Object and thus doesn’t have a lease.

- **Cancel-Subscription operation:** This operation cancels the lease on the specified Per-Printer Subscription Object and thereby deletes the Subscription Object.

When an Event occurs, the Printer finds all Subscription Objects listening for the Event (see section 9 for details on finding such Subscription Objects). For each such Subscription Object, the Printer:

a) generates an Event Notification with information specified in section 9, AND

b) either:

   i) delivers the Event Notification using the Delivery Method and target address identified in the Subscription Object’s “notify-recipient-uri” attribute if the Delivery Method is a “push”, OR

   ii) saves Event Notification for a time period defined by the Delivery Method if the Delivery Method is a “pull”, i.e., the Notification Recipient is expected to fetch the Event Notifications.

2 Models for Notification

2.1 Model for Notification (Simple Case)

As part of a Subscription Creation Operation, an IPP Printer (i.e., an output device or a server) creates one or more Subscription Objects. In a Subscription Creation Operation, the client specifies the Notification Recipient to which the Printer is to deliver Event Notifications. A Notification Recipient can be the Subscribing Client or a third party.

Figure 1 shows the Notification model for a simple Client-Printer relationship.

```
embedded printer:
                                        output device or server
   PDA, desktop, or server                 +---------------+
        +--------+                         |  ###########  |
        | client |-----Subscription ---------># Printer #  |
        +--------+  Creation Operation     |  # Object  #  |
     +------------+                        |  #####|#####  |
     |Notification|                        +-------|-------+
     |Recipient   |<----IPP Event Notifications----+
     +------------+    (Job and/or Printer Events)
```

Figure 1 – Model for Notification

2.2 Model for Notification with Cascading Printers

With this model, there is an intervening Print server between the human user and the Printer in the output device. If the Printer in the output device generates an Event, the system can be configured to send Event Notification either

- directly to the Notification Recipient specified by the Subscribing Client or
- via the Print Server to the Notification Recipient specified by the Subscribing Client.

See Appendix A for more details.

2.3 Distributed Model for Notification

The preceding sections (2.1 and 2.2) assume that the Notification software resides in the same device or Server box as the rest of the Printer software. In many implementations, the assumption is correct. However, the Notification model also permits a distributed implementation.

For example, the software that supports both Subscription Creation Operations and sending of Event Notifications could be on hardware that is separate from the output device. To make this work, there must be a symbiotic relationship between the output device software and the remote Notification software. Without the remote Notification software, the output device software is not a complete Printer.

The term “Printer” in this document includes the software on the output device or server box as well as Notification software that is local to or remote from the output device.

Appendix B describes this example in detail.

2.4 Extended Notification Recipient

The model allows for an extended Notification Recipient that is itself a Notification service that forwards each Event Notification to another recipient. The client contacts this Notification Recipient to arrange for forwarding by means outside the scope of this document. The Printer need not be aware that the Notification Recipient forwards Event Notifications.

Appendix C describes this example in detail.

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]. See Appendix D for complete details.

Note: a feature that is OPTIONAL in this document becomes REQUIRED if the Printer implements a Delivery Method that REQUIRES the feature

READ-ONLY - an adjective used in an attribute definition to indicate that an IPP Printer MUST NOT allow the attribute's value to be modified with the Set-Job-Attributes or Set-Printer-Attributes operations (see [ipp-set]). Note: there is no Set-Subscription operation so this term is not used for Subscription object attributes.

3.2 Other Terminology

Administrator - A human user who establishes 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.

IPP Client (or client) - The software component (PDA, desktop, or server) that performs an IPP operation directed at an IPP Printer (server or output device).

Job Creation operation - One of the operations that creates a Job object: Print-Job, Print-URI and Create-Job. The Validate-Job operation is not a Job Creation operation because no Job object is created. Therefore, when a statement also applies to the Validate-Job operation, it is mentioned explicitly.

Event - some occurrence (either expected or unexpected) within the printing system of a change of state, condition, or configuration of a Job or Printer object. An Event occurs only at one instant in time and does not span the time the physical Event takes place. For example, jam-occurred and jam-cleared are two distinct, instantaneous Events, even though the jam may last for a while.

Job Event – an Event caused by some change in a particular job on the Printer, e.g., job-completed.

Printer Event – an Event caused by some change in the Printer that is not specific to a job, e.g., printer-state-changed.

Subscribed Event – an Event that the Subscribing Client expresses interest in by making it a value of the "notify-events" attribute on a Subscription Object.

Subscribed Job Event – a Subscribed Event that is a Job Event.

Subscribed Printer Event – a Subscribed Event that is a Printer Event.

Event Notification - the information about an Event that the Printer sends when an Event occurs.

Notification Recipient - the entity to which the Printer sends an Event Notification.

Delivery Method - the mechanism by which the Printer delivers the Event Notification, e.g., via email or via SNMP.

Delivery Method Document - a document, separate from this document, that defines a Delivery Method.

Compound Event Notification - two or more Event Notifications that a Printer sends together as a single entity. The Delivery Method Document specifies whether the Delivery Method supports Compound Event Notifications.

Subscription Object - An object containing a set of attributes that indicate: the Notification Recipient, the Delivery Method, the Subscribed Events that cause the Printer to send an Event Notification, and the information to send in an Event Notification.

Per-Job Subscription Object - A Subscription Object that is associated with a single Job. The Create-Job-Subscriptions operation and Job Creation operations create such an object.

Per-Printer Subscription Object - A Subscription Object that is associated with the Printer as a whole. The Create-Printer-Subscriptions operation creates such an object.

Subscribing Client - The client that creates the Subscription Object.

Subscription Creation Operation - An operation that creates a Subscription Object: Job Creation operations, Create-Job-Subscriptions operation, and Create-Printer-Subscriptions operation. In the context of a Job Creation operation, a Subscription Creation Operation is the part of the Job Creation operation that creates a Subscription object.

Subscription Creation Request – The request portion of a Subscription Creation Operation.

Subscription Template Attributes – Subscription Object attributes that a client can supply in a Subscription Creation Operation and associated Printer Object attributes that specify supported and default values for the Subscription Object attributes.

Subscription Description Attributes – Subscription Object attributes that a Printer supplies during a Subscription Creation Operation.

Subscription Template Attributes Group – The attributes group in a request that contains Subscription Object attributes that are Subscription Template Attributes.

Subscription Attributes Group – The attributes group in a response that contains Subscription Object attributes.

Human Consumable Event Notification – localized text for human consumption only. There is no standardized format and thus programs should not try to parse this text.

Machine Consumable Event Notification - bytes for program consumption. The bytes are formatted according to the Delivery Method document.

Printer – the software that supports an output device or print server (see IPP/1.1 [ipp-mod] which uses the terms Printer and Printer object interchangeably). This document extends the IPP/1.1 Printer definition to include the software that implements Subscription Creation Operations and the sending of Event Notifications, even if the software for such a Printer would be distributed across a network (see section 2.3).

Notification – when not in the phrases ‘Event Notification’ and ‘Notification Recipient’ — the concepts of this specification, i.e., Events, Subscription Objects, and Event Notifications.

4 Object Relationships

This section defines the object relationships between the Printer, Job, and Subscription Objects. It does not define the implementation. For an illustration of these relationships, see Appendix E.

4.1 Printer and Per-Printer Subscription Objects

1. A Printer object can be associated with zero or more Per-Printer Subscription Objects.

2. Each Per-Printer Subscription Object is associated with exactly one Printer object.

4.2 Printer, Job and Per-Job Subscription Objects

1. A Printer object is associated with zero or more Job objects.

2. Each Job object is associated with exactly one Printer object.

3. A Job object is associated with zero or more Per-Job Subscription Objects.

4. Each Per-Job Subscription Object is associated with exactly one Job object.

5 Subscription Object

A Subscribing Client creates a Subscription Object with a Subscription Creation Operation in order to indicate its interest in certain Events. See section 11 for a description of these operations. When an Event occurs, the Subscription Object specifies to the Printer where to send Event Notifications, how to send them and what to put in them. See section 9 for details on the contents of an Event Notification.

Using the IPP Job Template attributes as a model (see [ipp-mod] section 4.2), the attributes of a Subscription Object are divided into two categories: Subscription Template Attributes and Subscription Description Attributes.

Subscription Template attributes are, in turn, like the Job Template attributes, divided into

1. Subscription Object attributes that a client can supply in a Subscription Creation Request and

2. their associated Printer Object attributes that specify supported and default values for the Subscription Object attributes

The remainder of this section specifies general rules for Subscription Template Attributes and describes each attribute in a Subscription Object.

5.1 Rules for Support of Subscription Template Attributes

Subscription Template Attributes are fundamental to the Notification model described in this specification. The client supplies these attributes in Subscription Creation Operations and the Printer uses these attributes to populate a newly created Subscription Object.

Subscription Objects attributes that are Subscription Template Attributes conform to the following rules:

1. Each attribute’s name starts with the prefix string “notify-” and this document calls such attributes “notify-xxx”.

2. For each “notify-xxx” Subscription Object attribute defined in column 1 of Table 1, Table 1 specifies corresponding Printer attributes: “notify-xxx-default”, “notify-xxx-supported”, “yyy-supported” and “notify-max-xxx-supported” defined in column 2 of Table 1.

3. If a Printer supports “notify-xxx” in column 1 of Table 1, then the Printer MUST support all associated attributes specified in column 2 of Table 1. For example, Table 1 shows that if the Printer supports “notify-events”, it MUST support “notify-events-default”, “notify-events-supported” and “notify-max-events-supported”.

4. If a Printer does not support “notify-xxx” in column 1 of Table 1, then the Printer MUST NOT support any associated “notify-yyy” attributes specified in column 2 of Table 1. For example, Table 1 shows that if the Printer doesn’t support “notify-events”, it MUST NOT support “notify-events-default”, “notify-events-supported” and “notify-max-events-supported”. Note this rule does not apply to attributes whose names do not start with the string “notify-” and are thus defined in another object and used by other attributes.

5. Most “notify-xxx” attributes have a corresponding “yyy-supported” attribute that specifies the supported values for “notify-xxx”. Column 2 of Table 1 specifies the name of each “yyy-supported” attribute. The naming rules of IPP/1.1 (see [ipp-mod]) are used when “yyy-supported” is “notify-xxx-supported”.

6. Some “notify-xxx” attributes have a corresponding “notify-xxx-default” attribute that specifies the value for “notify-xxx” if the client does not supply it. Column 2 of Table 1 specifies the name of each “notify-xxx-default” attribute. The naming rules of IPP/1.1 (see [ipp-mod]) are used.

If a client wishes to present an end user with a list of supported values from which to choose, the client SHOULD query the Printer for its supported value attributes. The client SHOULD also query the default value attributes. If the client then limits selectable values to only those values that are supported, the client can guarantee that the values supplied by the client in the create request all fall within the set of supported values at the Printer. When querying the Printer, the client MAY enumerate each attribute by name in the Get-Printer-Attributes Request, or the client MAY just supply the ‘subscription-template’ group name in order to get the complete set of supported attributes (both supported and default attributes).

5.2 Rules for Processing Subscription Template Attributes

This section defines a detailed set of rules that a Printer follows when it processes Subscription Template Attributes in a Subscription Creation Request. These rules for are similar to the rules for processing Operation attributes in [ipp-mod]. That is, the Printer may or may not support an attribute and a client may or may not supply the attribute. Some combinations of these cases are OK. Others return warnings or errors, and perhaps a list of unsupported attributes.

A Printer MUST implement the following behavior for processing Subscription Template Attributes in a Subscription Creation Request:

1. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer supports it and its value, the Printer MUST populate the attribute on the created Subscription Object.

2. If a client supplies a “notify-xxx” attribute from column 1 of Table 1 and the Printer doesn’t support it or its value, the Printer MUST NOT populate the attribute on the created Subscription Object with it. The Printer MUST do one of the following:

    a) If the value of the “notify-xxx” attribute is unsupported, the Printer MUST return the attribute with its value in the Subscription Attributes Group of the response.

    b) If “notify-xxx” is an unsupported attribute, the Printer MUST return the attribute in the Subscription Attributes Group of the response with the ‘unsupported’ out-of-band value.

    Note: The rules of this step are the same as for Unsupported Attributes [ipp-mod] section 3.1.7. except that the unsupported attributes are returned in the Subscription Attributes Group rather than the Unsupported Attributes Group because Subscription Creation Operations can create more than one Subscription Object).

3. If a client is REQUIRED to supply a “notify-xxx” attribute from column 1 of Table 1 and the Printer doesn’t support the supplied value, the Printer MUST NOT create a Subscription Object. The rules for Unsupported Attributes in step #2 still apply.

4. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 and the attribute is REQUIRED for the client to supply, the Printer MUST reject the Subscription Creation Operation (including Job Creation operations) without creating a Subscription Object, and MUST return in the response:

    c) the status code ‘client-error-bad-request’ AND

    d) no Subscription Attribute Groups.

5. If a client does not supply a “notify-xxx” attribute from column 1 of Table 1 that is OPTIONAL for the client to supply, and column 2 of Table 1 either:

    a) specifies a “notify-xxx-default” attribute, the Printer MUST behave as if the client had supplied the “notify-xxx-default” attribute (see step #1) and populate the Subscription object with the value of the “notify-xxx-default” attribute as part of the Subscription Creation operation (unlike Job Template attributes where the Printer does not populate the Job object with defaults - see [ipp-mod]) OR

    b) does not specify a “notify-xxx-default” attribute, the Printer MUST populate the “notify-xxx” attribute on the Subscription Object according to the definition of the “notify-xxx” attribute in a section 5.3. For some attributes, the “notify-xxx” is populated with the value of some other attribute, and for others, the “notify-xxx” is NOT populated on the Subscription object at all.

6. A Printer MUST create a Subscription Object for each Subscription Template Attributes group in a request unless the Printer:

    a) encounters some attributes in a Subscription Template Attributes Group that require the Printer not to create the Subscription Object OR

    b) would ~~be~~ create a Per-Job Subscription Object ~~and the number of Per-Job Subscription Objects already equals the value of the “notify-max-job-subscriptions-supported” Printer attribute~~when it doesn’t have space for another Per-Job Subscription Object OR

    c) would create ~~be~~ a Per-Printer Subscription Object when it doesn’t ~~doesn’t~~ have space for another Per-Printer Subscription Object~~and the number of Per-Printer Subscription Objects already equals the value of the “notify-max-printer-subscriptions-supported” Printer attribute~~.

7. A response MUST contain one Subscription Attributes Group for each Subscription Template Attributes Group in the request (and in the same order) whether the Printer creates a Subscription Object from the Subscription Template Attributes Group or not. However, the attributes in each Subscription Attributes Group can be in any order.

8. The Printer MUST populate each Subscription Attributes Group of the response such that each contains:

    a) the “notify-subscription-id” attribute (see section 5.4.1), if and only if the Printer creates a Subscription Object.

b) the "notify-lease-duration" attribute (see section 5.3.7), if and only if the Printer creates a Per-Printer Subscription Object. The value of this attribute is the value of the Subscription Object's "notify-lease-duration" attribute. This value MAY be different from the client-supplied value (see section 5.3.7). If a client supplies this attribute in the creation of a Per-Job Subscription Object, it MUST appear in this group with the out-of-band value 'unsupported' to indicate that the Printer doesn't support it in this context.

c) all of the unsupported Subscription Template Attributes from step #2.

d) the "notify-status-code" attribute if the Printer does not create the Subscription Object or if there are unsupported attributes from step #2. The possible values of the "notify-status-code" attribute are shown below (see section 17 for more details). The Printer returns the first value in the list below that describes the status.

'client-error-uri-scheme-not-supported': the Subscription Object was not created because the scheme of the "notify-recipient-uri" attribute is not supported. See section 17.1 for more details about this status code. See step #3 in this section for the case that causes this error, and the resulting step #6a) that causes the Printer not to create the Subscription Object.

'client-error-too-many-subscriptions': the Subscription Object was not created because the Printer has no space for additional Subscription Objects~~the number of Subscription Objects would exceed the value of the Printer's "notify-max-job-subscriptions-supported" or "notify-max-printer-subscriptions-supported" attributes~~. The client SHOULD try again later. See section 17.2 for more details about this status code. See steps #6b) and #6c) in this section for the cases that causes this error.

'successful-ok-too-many-events': the Subscription Object was created without the "notify-events" values included in this Subscription Attributes Group because the "notify-events" attribute contains too many values. See section 17.3 for more details about this status code. See step #2 in this section and section 5.3.2 for the cases that cause this status code.

'successful-ok-ignored-or-substituted-attributes' : the Subscription Object was created but some supplied Subscription Template Attributes are unsupported. These unsupported attributes are also in the Subscription Attributes Group. See section 17.4 for more details about this status code. See step #2 in this section for the cases that cause this status code.

9. The Printer MUST validate all Subscription Template Attributes and MUST return all unsupported attributes and values in the corresponding Subscription Attributes Group of the response (see step #2) unless it determines that it could not create additional Subscription Objects because of condition #6b) or condition #6c). Then, the Printer NEED NOT validate these additional Subscription Template Attributes and the client MUST NOT expect to find unsupported attributes from step #2 in such additional Subscription Attribute Groups.

5.3 Subscription Template Attributes

This section contains the Subscription Template Attributes defined for the Subscription and Printer objects.

Table 1 below shows the Subscription Template Attributes and has two columns:

- **Attribute in Subscription Object:** the name and attribute syntax of each Subscription Object Attribute that is a Subscription Template Attribute

- **Default and Supported Printer Attributes:** the default attribute and supported Printer attributes that are associated with the attribute in column 1.

A Printer MUST support all attributes in Table 1 below except for “notify-attributes” (and “notify-attributes-supported”). A client MUST supply “notify-recipient-uri” and MAY omit any of the rest of the attributes in column 1 of Table 1 in a Subscription Creation Request.

Table 1 – Subscription Template Attributes

Attribute in Subscription Object	Default and Supported Printer Attributes
notify-recipient-uri (uri)	notify-schemes-supported (1setOf uriScheme)
notify-events (1setOf type2 keyword)	notify-events-default (1setOf type2 keyword) notify-events-supported (1setOf type2 keyword) notify-max-events-supported (integer(2:MAX))
notify-attributes (1setOf type2 keyword)	notify-attributes-supported (1setOf type2 keyword)
notify-user-data (octetString(63))	
notify-charset (charset)	charset-supported (1setOf charset)
notify-natural-languages (naturalLanguage)	generated-natural-language-supported (1setOf naturalLanguage)
notify-lease-duration (integer(0:MAX))	notify-lease-duration-default (integer(0:67108863)) notify-lease-duration-supported (1setOf (integer(0: 67108863) | rangeOfInteger(0:67108863)))
notify-time-interval (integer(0:MAX))	

5.3.1 notify-recipient-uri (uri)

This attribute’s value is a URL, which is a special case of a URI. Its value consists of a scheme and an address. The address specifies the Notification Recipient and the scheme specifies the Delivery Method for each Event Notification associated with this Subscription Object.

A Printer MUST support this attribute.

A client MUST supply this attribute in Subscription Creation Operation. Thus there is no need for a default attribute.

The “notify-schemes-supported (1setOf uriScheme)” attribute MUST specify the schemes supported for this attribute.

If the client supplies an unsupported scheme in the value of this attribute, then the Printer MUST not create the Subscription Object and MUST return the “notify-status-code” attribute with the ‘client-error-uri-scheme-not-supported’ value in the Subscription Attributes Group in the response.

The Printer MUST treat the address part of this attribute as opaque.

5.3.2 notify-events (1setOf type2 keyword)

This attribute contains a set of Subscribed Events. When an Event occurs and it “matches” a value of this attribute, the Printer sends an Event Notification using information in the Subscription Object. The details of “matching” are described subsection 5.3.2.2.

A Printer MUST support this attribute.

A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this attribute in Subscription Creation Operation, the Printer MUST populate this attribute on the Subscription Object with its “notify-events-default” attribute value.

Each value of this attribute on a Subscription Object MUST be one of the values of the “notify-events-supported (1setOf type2 keyword)” attribute.

The number of values of this attribute MUST NOT exceed the value of the “notify-max-events-supported” attribute. A Printer MUST support at least 2 values per Subscription Object. If the number of values supplied by a client in a Subscription Creation Operation exceeds the value of this attribute, the Printer MUST treat extra values as unsupported values and MUST use the value of ‘successful-ok-too-many-events’ for the “notify-status-code” attribute in the Subscription Attributes Group of the response.

~~ISSUE 01: OK that we changed the number from 5 to 2 because we have rearranged the categories of Events to have group events?~~

5.3.2.1 Standard Values for Subscribed Events

Each value of this attribute is a keyword and it specifies a Subscribed Event that represents certain changes. Some keywords represent a subset of changes of another keyword, e.g., ‘job-completed’ is an Event value which is a sub-value of ‘job-state-change’. See section 5.3.2.2 for the case where this attribute contains both a value and a sub-value.

The values in this section are divided into three categories: No Events, Job Events and Printer Events.

A Printer MUST support the Events indicated as “REQUIRED” and MAY support the Events indicated as “OPTIONAL”.

5.3.2.1.1 No Events

The standard and only keyword value for No Events is:

'none': REQUIRED - no Event Notifications for any Events. As the sole value of "notify-events-supported", this value means that the Printer does not support the sending of Event Notifications. As the sole value of "notify-events-default", this value means that a client MUST specify the "notify-events" attribute in order for a Subscription Creation Operation to succeed. If the Printer receives this value as the sole value of a Subscription Creation Operation, it does not create a Subscription Object. If a Printer receives this value with other values of a Subscription Creation Operation, the Printer MUST treat this value as an unsupported value.

5.3.2.1.2 Subscribed Printer Events

~~For a Printer, the first Printer Event MUST be 'printer-restarted' and the last Printer Event MUST be 'printer-shutdown'.~~

The standard keyword values for Subscribed Printer Events are:

'printer-state-changed': REQUIRED - the Printer changed state from any state to any other state. Specifically, the value of the Printer's "printer-state", "printer-state-reasons" or "printer-is-accepting-jobs" attributes changed.

This Subscribed Event value has the following sub-values: 'printer-restarted' and 'printer-shutdown'. A client can listen for any of these sub-values if it doesn't want to listen to all printer-state changes:

'printer-restarted': OPTIONAL - when the printer is powered up ~~or the Restart-Printer operation is performed (see [ipp-set2]). This event is the first Printer Event that can be received from a Printer~~.

'printer-shutdown': OPTIONAL - when the device is being powered down ~~or the Shutdown-Printer operation has been performed (see [ipp-set2]). This event is the last Printer Event that can be received from a Printer~~.

'printer-stopped: REQUIRED - when the printer stops printing, i.e. the value of the "printer-state" Printer attribute becomes 'stopped'.

'printer-config-changed': OPTIONAL - when the configuration of a Printer has changed, i.e., the value of the "printer-message-from-operator" or any "configuration" Printer attribute has changed. A "configuration" Printer attribute is an attribute which can change value because of some human interaction either direct or indirect, and which is not covered by one of the other Events in this section. Examples of "configuration" Printer attributes are any of the Job Template attributes, such as "xxx-supported", "xxx-ready" and "xxx-default". Often, such a change is the result of a client performing a Set-Printer-Attributes operation (see [ipp-set]) on the Printer. 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.

This Event value has the following sub-values: 'printer-media-changed' and 'printer-finishings-changed'. A client can listen for any of these sub-values if it doesn't want to listen to all printer-configuration changes:

'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 two cases: an input tray that goes empty and an input tray that receives additional media of the same type or of a different type. The client must check the "media-ready" Printer attribute (see [ipp-mod] section 4.2.11) separately to find out what changed.

'printer-finishings-changed': OPTIONAL - when the finisher on a printer has been changed, i.e., the "finishings-ready" attribute has changed. This Event includes two cases: a finisher that goes empty and a finisher that is refilled (even if it is not full). The client must check the "finishings-ready" Printer attribute separately to find out what changed.

'printer-queue-order-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 has just become able to accept a Job Creation operation, Send-Document operation, or Send-URI operation. A Printer sends this Event when it has acquired more buffer space to accept jobs after it previously did not have room to accept any more jobs and would have rejected a Job Creation Operation, a Send-Document operation, or Send-URI operation. A Notification Recipient listens for this Event when there is more than one client feeding a printer/server (fan-in).~~

~~**'printer-full'**: OPTIONAL - when the Printer has just become unable to accept a Job Creation operation, Send-Document operation, or Send-URI operation due to lack of buffer space. It is intended that a Notification Recipient use this Event to stop whatever the 'printer-no-longer-full' Event starts.~~

~~**ISSUE 02**: OK to add 'printer-full' Event?~~

~~**'printer-almost-idle'**: OPTIONAL - when the Printer needs another Job in order to stay busy. A Printer that is an output device MAY use this Event to request a new job sufficiently ahead of time so as not to run out of work between jobs. A Printer that is a fan-out spooler MAY listen for this Event and hold pending Jobs until a downstream Printer sends this Event to indicate that it needs another Job in order to stay busy.~~

~~**'printer-not-almost-idle'**: OPTIONAL - when the Printer no longer needs another Job in order to stay busy. It is intended that a Notification Recipient use this Event to stop whatever the 'printer-almost-idle' Event starts.~~

~~**ISSUE 03**: OK to add 'printer-not-almost-idle' Event?~~

5.3.2.1.3 Subscribed Job Events

~~For each Job object, the first Job Event MUST be 'job-created' and the last Job Event MUST be 'job-completed'.~~

The standard keyword values for Subscribed Job Events are:

‘job-state-changed’: REQUIRED - the job has changed from any state to any other state. Specifically, the Printer sends this Event whenever the value of the “job-state” attribute or “job-state-reasons” attribute changes. When a Job is removed from the Job History (see [ipp-mod] 4.3.7.1), no Event is generated.

This Event value has the following sub-values: ‘job-created’, ‘job-completed’ and ‘job-purged’. A client can listen for any of these sub-values if it doesn’t want to listen to all ‘job-state changes’.

‘job-created’: REQUIRED - the Printer has accepted a Job Creation operation 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.~~ This event is the first Job Event that can be received from a Job~~.

‘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).~~ This event is the last Job Event that can be received from a Job~~. The Printer also sends this Event when a Job is removed with the Purge-Job operation. In this case, the Event Notification MUST report the ‘job-state’ as ‘canceled’.

~~**‘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. The Printer MUST immediately send a ‘job-completed’ event after this event to meet the requirement that ‘job-completed’ is the last event for the Job.~~

‘job-stopped: OPTIONAL - when the job stops printing, i.e. the value of the “job-state” Job attribute becomes ‘processing-stopped’.

‘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 “configuration” Job attributes have changed. A “configuration” Job attribute is an attribute that can change value because of some human interaction either direct or indirect. Examples of “configuration” Job attributes are any of the job template attributes and 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-progress’:** OPTIONAL – when the Printer has completed Printing a sheet. ~~an impression, sheet, or copy has completed. ~~See the separate [ipp-prog] specification for additional attributes that a Printer MAY send in an Event Notification caused by this Event. The “notify-time-interval” attribute affects this Event by causing the Printer NOT to send an Event Notification every time a ‘job-progress’ Events occurs. See section 5.3.8 for full details.

5.3.2.2 Rules for Matching of Subscribed Events

When an Event occurs, the Printer MUST find each Subscription object whose “notify-events” attribute “matches” the Event. The rules for “matching” of Subscribed Events are described separately for Printer Events and for Job Events. This section also describes some special cases.

5.3.2.2.1 Rules for Matching of Printer Events

Suppose that the Printer causes Printer Event E to occur. For each Per-Job or Per-Printer Subscription S in the Printer, if E equals a value of this attribute in S or E is a sub-value of a value of this attribute in S, the Printer MUST generate an Event Notification.

> Consider the example. There are three Subscription Objects each with the Subscribed Printer Event ‘printer-state-changed’. Subscription Object A is a Per-Printer Subscription Object. Subscription Object B is a Per-Job Subscription Object for Job 1, and Subscription Object C is a Per-Job Subscription Object for Job 2. When the Printer enters the ‘stopped’ state, the Printer sends an Event Notification to the Notification Recipients of Subscription Objects A, B, and C because this is a Printer Event. Note if Job 1 has already completed, the Printer would not send an Event Notification for its Subscription Object.

5.3.2.2.2 Rules for Matching of Job Events

Suppose that Job J causes Job Event E to occur.

3. For each Per-Printer Subscription S in the Printer, if E equals a value of this attribute in S or E is a sub-value of a value of this attribute in S, the Printer MUST generate an Event Notification.

4. For each Per-Job Subscription S associated with Job J, if E equals a value of this attribute in S or E is a sub-value of a value of this attribute in S, the Printer MUST generate an Event Notification.

5. For each Per-Job Subscription S that is NOT associated Job J, if E equals a value of this attribute in S or E is a sub-value of a value of this attribute in, the Printer MUST NOT generate an Event Notification from S.

> Consider the example: There are three Subscription Objects listening for the Job Event ‘job-completed’. Subscription Object A is a Per-Printer Subscription Object. Subscription Object B is a Per-Job Subscription Object for Job 1, and Subscription Object C is a Per-Job Subscription Object for Job 2. In addition, Per-Printer Subscription Object D is listening for the Job Event ‘job-state-changed’. When Job 1 completes, the Printer sends an Event Notification to the Notification Recipient of Subscription Object A (because it is Per-Printer) and Subscription Object B because it is a Per-Job Subscription Object associated with the Job generating the Event. The Printer also sends an Event Notification to the Notification Recipient of Subscription Object D because ‘job-completed’ is a sub-value of ‘job-state-changed’ – the value that Subscription Object D is listening for. The Printer does not send an Event Notification to the Notification Recipients of Subscription Object C because it is a Per-Job Subscription Object associated with some Job other than the Job generating the Event.

5.3.2.2.3 Special Cases for Matching Rules

This section contains rule for special cases.

If an Event matches Subscribed Events in two different Subscription Objects and the Printer would send two identical Event Notifications (except for the “notify-subscription-id” attribute) to the same Notification Recipient using the same Delivery Method, the Printer MUST send both Event Notifications. That is, the Printer MUST NOT try to consolidate seemingly identical Event Notifications that occur in separate Subscription objects. Incidentally, the Printer MUST NOT reject Subscription Creation Operations that would create this scenario.

If an Event matches two values of this “notify-events” attribute in a single Subscription object (e.g., a value and its sub-value), a Printer MAY send one Event Notification for each matched value in the Subscription Object or it MAY send only one Event Notification per Subscription Object. The rules in sections 5.3.2.2.1 and 5.3.2.2.2 are purposefully ambiguous about the number of Event Notification sent when Event E matches two or more values in a Subscription Object.

> Consider the example: There are two Per-Printer Subscription Objects when a Job completes. Subscription Object A has the Subscribed Job Event ‘job-state-changed’. Subscription Object B has the Subscribed Job Events ‘job-state-changed’ and ‘job-completed’. The Printer sends an Event Notification to the Notification Recipient of Subscription Object A with the value of ‘job-state-changed’ for the “notify-subscribing-event” attribute. The Printer sends either one or two Event Notifications to the Notification Recipient of Subscription Object B, depending on implementation. If it sends two Event Notifications, one has the value of ‘job-state-changed’ for the “notify-subscribing-event” attribute, and the other has the value of ‘job-completed’ for the “notify-subscribing-event” attribute. If it sends one Event Notification, it has the value of either ‘job-state-changed’ or ‘job-completed’ for the “notify-subscribing-event” attribute, depending on implementation. The algorithm for choosing such a value is implementation dependent.

~~In addition, Delivery Methods MAY allow the Printer to moderate certain high frequency events (see section 5.3.8).~~

5.3.3 notify-attributes (1setOf type2 keyword)

This attribute contains a set of attribute names. When a Printer sends a Machine Consumable Event Notification, it includes a fixed set of attributes (see section 9.1). If this attribute is present and the Event Notification is Machine Consumable, the Printer also includes the attributes specified by this attribute.

A Printer MAY support this attribute.

A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this attribute in Subscription Creation Operation or the Printer does not support this attribute, the Subscription Object MUST NOT contain the “notify-attributes” attribute. There is no “notify-attributes-default” attribute.

Each keyword value of this attribute on a Subscription Object MUST be a value of the “notify-attributes-supported (1setOf type2 keyword)” attribute. The “notify-attributes-supported” MAY contain any Printer attribute, Job attribute or Subscription Object attribute that the Printer supports in an Event Notification. It MUST NOT contain any of the attributes in Section 9.1 that a Printer automatically puts in an Event

Notification; it would be redundant. If a client supplies an attribute in Section 9.1, the Printer MUST treat it as an unsupported attribute value of the “notify-attributes” attribute.

The following rules apply to each keyword value N of the “notify-attributes” attribute: If the value N names:

a) a Subscription attribute, the Printer MUST use the attribute N in the Subscription Object that is being used to generate the Event Notification.

b) a Job attribute and the Printer is generating an Event Notification from a Per-Job Subscription Object S, the Printer MUST use the attribute N in the Job object associated with S.

c) a Job attribute and the Printer is generating an Event Notification from a Per-Printer Subscription Object and the Event is:

   - a Job Event, the Printer MUST use the attribute N in the Job object that caused the Event.

   - a Printer Event, the Printer MUST use the attribute N in the active Job.

If a Printer supports this attribute and a Subscription Object contains this attribute and the Delivery Method generates a Machine Consumable Event Notification, the Printer MUST include in each Event Notification:

a) the attributes specified in section 9.1 and

b) each attribute named by this attribute.

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

This attribute contains opaque data that some Delivery Methods include in each Machine Consumable Event Notification. The opaque data might contain, for example:

- the identity of the Subscriber

- a path or index to some Subscriber information

- a key that identifies to the Notification Recipient the ultimate recipient of the Event Notification

- the id for a Notification Recipient that had previously registered with an Instant Messaging Service

A Printer MUST support this attribute.

A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this attribute in Subscription Creation Operation, the Subscription Object MUST NOT contain the “notify-user-data” attribute. There is no “notify-user-data-default” attribute.

There is no “user-data-supported” attribute. Rather, any octetString whose length does not exceed 63 octets is a supported value. If the length exceeds 63 octets, the Printer MUST treat it as an unsupported value.

5.3.5 notify-charset (charset)

This attribute specifies the charset to be used in the Event Notification content sent to the Notification Recipient, whether the Event Notification content is Machine Consumable or Human Consumable.

A Printer MUST support this attribute.

A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this attribute in Subscription Creation Operation or supplies an unsupported value, the Printer MUST populate this attribute in the Subscription Object with the value of the “attributes-charset” operation attribute, which is a REQUIRED attribute in all IPP requests (see [ipp-mod]). If the value of the “attributes-charset” attribute is unsupported, the Printer MUST populate this attribute in the Subscription Object with the value of the Printer’s “charset-configured” attribute. There is no “notify-charset-default” attribute.

The value of this attribute on a Subscription Object MUST be a value of the “charset-supported (1setOf charset)” attribute.

5.3.6 notify-natural-language (naturalLanguage)

This attribute specifies the natural language to be used in any human consumable text in the Event Notification content sent to the Notification Recipient, whether the Event Notification content is Machine Consumable or Human Consumable.

A Printer MUST support this attribute.

A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this attribute in Subscription Creation Operation or supplies an unsupported value, the Printer MUST populate this attribute in the Subscription Object with the value of the “attributes-natural-language” operation attribute, which is a REQUIRED attribute in all IPP requests (see [ipp-mod]). If the value of the “attributes-natural-language” attribute is unsupported, the Printer MUST populate this attribute in the Subscription Object with the value of the Printer’s “natural-language-configured” attribute. There is no “notify-natural-language-default” attribute.

The value of this attribute on a Subscription Object MUST be a value of the “generated-natural-language-supported (1setOf type2 naturalLanguage)” attribute.

5.3.7 notify-lease-duration (integer(0:67108863))

This attribute specifies the duration of the lease associated with the Per-Printer Subscription Object at the time the Subscription Object was created or the lease was renewed. The duration of the lease is infinite if the value is 0, i.e., the lease never expires.

This attribute is not present on a Per-Job Subscription Object because the Subscription Object lasts exactly as long as the associated Job object. See section 5.4.3 on “notify-lease-expiration-time (integer(0:MAX))” for more details.

A Printer MUST support this attribute.

For a Subscription Object Creation operation of a Per-Job Subscription Object, the client MUST NOT supply this attribute. If the client does supply this attribute, the Printer MUST treat it as an unsupported attribute.

For a Subscription Creation Operation of a Per-Printer Subscription Object or a Renew-Subscription operation, a client MAY supply this attribute. If the client does not supply this attribute, the Printer MUST populate this attribute with its “notify-lease-duration-default” (0:67108863) attribute value. If the client supplies this attribute with an unsupported value, the Printer MUST populate this attribute with a supported value, and this value SHOULD be as close as possible to the value requested by the client. Note: this rule implies that a Printer doesn’t assign the value of 0 (infinite) unless the client requests it.

After the Printer has populated this attribute with a supported value, the value represents the “granted duration” of the lease and the Printer sets the value of the Subscription Object’s “notify-lease-expiration-time” attribute as specified in section 5.4.3.

The value of this attribute on a Subscription Object MUST be a value of the “notify-lease-duration-supported” (1setOf (integer(0:67108863) | rangeOfInteger(0:67108863))) attribute.

A Printer MAY require authentication in order to return the value of 0 (the lease never expires) as one of the values of “notify-lease-duration-supported”, and to allow 0 as a value of the “notify-lease-duration” attribute.

Note: The maximum value 67,108,863 is 2 raised to the 26 power minus 1 and is about 2 years in seconds. The value is considerably less than MAX so that there is virtually no chance of an overflow when it is added to “printer-up-time” to produce “notify-lease-expiration-time”.

5.3.8 notify-time-interval (integer(0:MAX))

The ‘job-progress’ Event occurs each time that a Printer completes a sheet. Some Notification Recipients do not want to receive an Event Notification every time this Event occurs. This attribute allows a Subscribing Client to request how often it want to receive Event Notifications for ‘job-progress’ Events.

The Printer MUST support this attribute if and only if the Printer supports the ‘job-progress’ Event.

A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this attribute, the Printer MUST not populate this attribute on the Subscription Object. There is no default “notify-time-interval-default” attribute.

There is no “notify-time-interval-supported”. The value of this attribute MAY be any nonnegative integer (0,MAX).

If the ‘job-progress’ Event occurs and a Subscription Object contains the ‘job-progress’ Event as a value of the ‘notify-events’ attribute, there are two cases to consider:

1. This attribute is not present on the Subscription Object or has the value of 0. The Printer MUST generate and send an Event Notification (as is the case with other Events).

2. This attribute is present with a nonzero value of N:

    a) If the Printer has not sent an Event Notification for the ‘job-progress’ Event for the associated Subscription Object within the past N seconds, the Printer MUST send an Event Notification for the Event that just occurred. Note when the Printer completes the first page of a Job, this rule implies that the Printer sends an Event Notification for a Per-Job Subscription Objects.

    b) Otherwise, the Printer MUST NOT generate or send an Event Notification for the associated Subscription Object. The Printer MUST NOT increase the value of the “notify-sequence-number” Subscription Object attribute (i.e., the sequence of values of the “notify-sequence-number” attribute counts the Event Notifications that the Printer sent and not the Events that do not cause an Event Notification to be sent).

It is RECOMMENDED that a Subscribing Client use this attribute when it subscribes to the ‘job-progress’ Event, and that the value be sufficiently large to limit the frequency with which the Printer sends Event Notifications.

This attribute MUST not effect any Events other than ‘job-progress’.

~~notify-persistence (boolean)~~

~~This attribute specifies whether the Printer preserves the Subscription Object across power cycles.~~

~~A Printer MUST support this attribute.~~

~~A client MAY supply this attribute in a Subscription Creation Operation. If the client does not supply this attribute in Subscription Creation Operation, the Printer MUST populate this attribute with its “notify-persistence-default” (boolean) attribute value. If the client supplies this attribute with an unsupported value, the Printer MUST populate this attribute with a supported value. The Printer MAY populate this attribute with a value other than the one the client requests. For example, if the client specifies ‘true’ and the Printer doesn’t have space for another Subscription Object, it sets the value of this attribute to ‘false’. If the client specifies ‘false’ and the Printer has a policy of setting this attribute to ‘true’ if there is space, the Printer sets this attribute to ‘true’.~~

~~The value of this attribute on a Subscription Object MUST be a value of the “notify-persistence-supported (1setOf boolean)” attribute. The “notify-persistence-supported” (1setOf boolean) attribute can have one of the following three values:~~

- ~~true: all Subscription Objects are persistent (if there is space).~~

- ~~false: no Subscription Objects are persistent~~

- ~~true, false: some Subscription Objects are persistent and others are not. For example, the Printer may have room for only 2 Subscription Objects.~~

~~It is RECOMMENDED that all Subscription Objects be persistent. If Jobs are persistent, the Per-Job Subscription Objects MUST be persistent too.~~

~~**ISSUE 04**: it would be better for this attribute to be a Subscription Description attribute that the Printer sets to show whether the Object is persistent or not. Agree?~~

5.4 Subscription Description Attributes

Subscription Description Attributes are those attributes that a Printer adds to a Subscription Object at the time of its creation.

A Printer MUST support all attributes in this Table 2.

A client MUST NOT supply the attributes in Table 2 in a Subscription Template Attributes Group of a Subscription Creation Operation. If the client supplies them, the Printer MUST NOT set them and MUST treat them as unsupported attributes. There are no corresponding default or supported attributes.

Table 2 – Subscription Description Attributes

Subscription Object attributes:
notify-subscription-id (integer(1:MAX))
notify-sequence-number (integer(0:MAX))
notify-lease-expiration-time (integer(0:MAX))
notify-printer-up-time (integer(1:MAX))
notify-printer-uri (uri)
notify-job-id (integer(1:MAX))
notify-subscriber-user-name (name(MAX))

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

This attribute identifies a Subscription Object instance with a number that is unique within the context of the Printer. The Printer generates this value at the time it creates the Subscription Object.

A Printer MUST support this attribute.

The Printer SHOULD NOT assign the value of this attribute sequentially as it creates Subscription Objects. Sequential assignment makes it easy for rogue clients to guess the value of this attribute on other Subscription Objects.

The Printer SHOULD avoid re-using recent values of this attribute during continuous operation of the Printer as well as across power cycles. Then a Subscribing Client is unlikely to find that a stale reference accesses a new Subscription Object.

The 0 value is not permitted in order to allow for compatibility with “job-id” and with SNMP index values, which also cannot be 0.

5.4.2 notify-sequence-number (integer (0:MAX))

The value of this attribute indicates the number of times that the Printer has generated and attempted to send an Event Notification. When an Event Notification contains this attribute, the Notification Recipient can determine whether it missed some Event Notifications (i.e., numbers skipped) or received duplicates (i.e., same number twice).

A Printer MUST support this attribute.

When the Printer creates a Subscription Object, it MUST set the value of this attribute to 0. This value indicates that the Printer has not sent any Event Notifications for this Subscription Object.

Each time the Printer sends a newly generated Event Notification, it MUST increase the value of this attribute by 1. For some Delivery Methods, the Printer MUST include this attribute in each Event Notification, and the value MUST be the value after it is increased by 1. That is, the value of this attribute in the first Event Notification after Subscription object creation MUST be 1, the second MUST be 2, etc. If a Delivery Method is defined such that the Notification Recipient returns a response, the Printer can re-try sending an Event Notification a certain number of times with the same sequence number when the Notification Recipient fails to return a response.

If a Subscription Object lasts long enough to reach the value of MAX, its next value MUST be 0, i.e., it wraps.

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

This attribute specifies the time in the future when the lease on the Per-Printer Subscription Object will expire, i.e. the “printer-up-time” value at which the lease will expire. If the value is 0, the lease never expires.

A Printer MUST support this attribute.

When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present – the Subscription Object lasts exactly as long as the associated Job object.

When the Printer creates a Per-Printer Subscription Object, it populates this attribute with a value that is the sum of the values of the Printer’s “printer-up-time” attribute and the Subscription Object’s “notify-lease-duration” attribute with the following exception. If the value of the Subscription Object’s “notify-lease-duration” attribute is 0 (i.e., no expiration time), then the value of this attribute MUST be set to 0 (i.e., no expiration time).

When the Printer powers up, it MUST set the value of this attribute in each persistent Subscription Object using the algorithm in the previous paragraph.

When the “printer-up-time” equals the value of this attribute, the Printer MUST delete the Subscription Object. A client can extend a lease of a Per-Printer Subscription Object with the Renew-Subscription operation (see section 11.2.5).

Note: In order to compute the number of seconds remaining in a lease for a Per-Printer Subscription Object, a client can subtract the Subscription's "notify-printer-up-time" attribute (see section 5.4.4) from the Subscription's "notify-lease-expiration-time" attribute.

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

This attribute is an alias for the Printer's "printer-up-time" attribute " (see [ipp-mod] section 4.4.29).

A Printer MUST support this attribute.

When the Printer creates a Per-Job Subscription Object, this attribute MUST NOT be present. When the Printer creates a Per-Printer Subscription Object, this attribute MUST be present.

Note: this attribute exists in a Per-Printer Subscription Object so that a client using the Get-Subscription-Attributes or Get-Subscription operations can convert the Per-Printer Subscription's "notify-lease-expiration-time" attribute to wall clock time with one request. If the value of the "notify-lease-expiration-time" attribute is not 0 (i.e., no expiration time), then the difference between the "notify-lease-expiration-time" attribute and the "notify-printer-up-time" is the remaining number of seconds on the lease from the current time.

5.4.5 notify-printer-uri (uri)

This attribute identifies the Printer object that created this Subscription Object.

A Printer MUST support this attribute.

During a Subscription Creation Operation, the Printer MUST populate this attribute with the value of the "printer-uri" operation attribute in the request. From the Printer URI, the client can, for example, determine what security scheme was used.

5.4.6 notify-job-id (integer(1:MAX))

This attribute specifies whether the containing Subscription Object is a Per-Job or Per-Printer Subscription Object, and for Per-Job Subscription Objects, it specifies the associated Job.

A Printer MUST support this attribute.

If this attribute is not present, the Subscription Object MUST be a Per-Printer Subscription. If this attribute is present, the Subscription Object MUST be a Per-Job Subscription Object and this attribute MUST identify the Job with which the Subscription Object is associated.

Note: This attribute could be useful to a Notification Recipient that receives an Event Notification generated from a Per-Job Subscription Object and caused by a Printer Event. The Event Notification gives access to the Printer and the Subscription Object. The Event Notification gives access to the associated Job only via this attribute.~~**ISSUE 05**: OK that we added the REQUIRED "notify-job-id" attribute because it is needed for a client to determine from a random subscription-id whether a Subscription is Per-Printer or Per-Job and if the latter which Job~~.

5.4.7 notify-subscriber-user-name (name(MAX))

This attribute contains the name of the user who performed the Subscription Creation Operation.

A Printer MUST support this attribute.

The Printer sets this attribute to the most authenticated printable name that it can obtain from the authentication service over which the Subscription Creation Operation was received. The Printer uses the same mechanism for determining the value of this attribute as it does for a Job's "job-originating-user-name" (see [ipp-mod] section 4.3.6).

Note: To help with authentication, a Subscription Object may have additional private attributes about the user, e.g., a credential of a principal. Such private attributes are implementation-dependent and not defined in this document.

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, indicates the Printer support required for conformance, and whether or not the attribute is READ-ONLY (see section 3.1):

Table 3 – Printer Description Attributes Associated with Notification

Printer object attributes:	REQUIRED	READ-ONLY
printer-state-change-time (integer(1:MAX))	No	Yes
printer-state-change-date-time (dateTime)	No	Yes

~~6.1notify-max-printer-subscriptions-supported (integer(0:MAX))~~

~~This attribute specifies the maximum number of un-expired Per-Printer Subscription Objects that the Printer supports at one time. A value of MAX indicates no effective maximum.~~

~~A Printer MUST support this attribute.~~

~~A Printer MUST support at least 1 Per-Printer Subscription Object. An implementation MAY allow an Administrator to set the value of this attribute to 0 in order to disable creation of Per-Printer Subscription Objects.~~

~~If the number of Per-Printer Subscription Objects equals the value of this attribute during a Subscription Creation Operation, the Printer MUST NOT create any additional Per-Printer Subscription Objects. See section 11.1.2 for details on the creation of Subscription Objects and how the Printer indicates such failure in a Subscription Creation Operation.~~

~~**ISSUE 06**: OK to use MAX to mean no limit and 0 to mean that an admin has turned off subscriptions?~~

~~6.2notify-max-job-subscriptions-supported (integer(0:MAX))~~

~~This attribute specifies the maximum number of Per-Job Subscription Objects that the Printer supports for each job. For example, if a Printer can hold 2 Jobs and this attribute has the value of 3, it can hold a total of 6 Per-Job Subscription Objects. A value of MAX indicates no effective maximum.~~

~~A Printer MUST support this attribute.~~

~~A Printer MUST support at least 1 Per-Job Subscription Object per Job. An implementation MAY allow an Administrator to set the value of this attribute to 0 in order to disable creation of Per-Job Subscription Objects.~~

~~If the number of Per-Job Subscription Objects associated with the specified Job equals the value of this attribute during a Subscription Creation Operation, the Printer MUST NOT create any additional Per-Job Subscription Objects. See section 11.1 for details on the creation of Subscription Objects and how the Printer indicates such failure in a Subscription Creation Operation.~~

~~**ISSUE 07**: OK to use MAX to mean no limit and 0 to mean that an admin has turned off subscriptions?~~

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

This attribute records the most recent time at which the ‘printer-state-changed’ Printer Event occurred whether or not any Subscription objects were listening for this event. This attribute helps a client or operator to determine how long the Printer has been in its current state.

A Printer MAY support this attribute and if so, the attribute MUST be READ-ONLY.

On power-up, the Printer MUST set the value of this attribute to be the value of its “printer-up-time” attribute, so that it always has a value. Whenever the ‘printer-state-changed’ Printer Event occurs, the Printer MUST set this attribute to the value of the Printer’s “printer-up-time” attribute.

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

This attribute records the most recent time at which the ‘printer-state-changed’ Printer Event occurred whether or not there were any Subscription Objects listening for this event. This attribute helps a client or operator to determine how long the Printer has been in its current state.

A Printer MAY support this attribute and if so, the attribute MUST be READ-ONLY.

On power-up, the Printer MUST set the value of this attribute to be the value of its “printer-current-time” attribute, so that it always has a value (see [ipp-mod] section 4.4.30 on “printer-current-time”). Whenever the ‘printer-state-changed’ Printer Event occurs, the Printer MUST set this attribute to the value of the Printer’s “printer-current-time” attribute.

7 New Values for Existing Printer Description Attributes

7.1 operations-supported (1setOf type2 enum)

The following “operation-id” values are added in order to support the new operations defined in this document:

Table 4 – Operation-id assignments

Value	**Operation Name**
0x0016	Create-Printer-Subscriptions
0x0017	Create-Job-Subscriptions
0x0018	Get-Subscription-Attributes
0x0019	Get-Subscriptions
0x001A	Renew-Subscription
0x001B	Cancel-Subscription

8 Attributes Only in Event Notifications

This section contains those attributes that exist only in Event Notifications.

8.1 notify-subscribed-event (type2 keyword)

This attribute indicates the Subscribed Event that caused the Printer to send this Event Notification. This attribute exists only in Event Notifications.

~~The Printer MUST send this attribute. This attribute exists only in Event Notifications.~~

This attribute MUST contain one of the values of the “notify-events” attribute in the Subscription Object, i.e., one of the Subscribed Event values. Its value is the Subscribed Event that “matches” the Event that caused the Printer to send this Event Notification. This Subscribed Event value may be identical to the Event or the Event may be a sub-value of the Subscribed Event. For example, the ‘job-completed’ Event (which is a sub-event of the ‘job-state-changed’ event) would cause the Printer to send an Event Notification for either the ‘job-completed’ or ‘job-state-changed’ Subscribed Events and to send the ‘job-completed’ or ‘job-state-changed’ value for this attribute, respectively~~,~~. See section 5.3.2.2 for the “matching” rules of Subscribed Events and for additional examples.

The Delivery Method Document specifies whether the Printer includes the value of this attribute in an Event Notification.

8.2 notify-text (text(MAX))

This attribute contains a Human Consumable text message (see section 9.2). This message describes the Event and is encoded as plain text, i.e., ‘text/plain’ with the charset specified by Subscription Object’s “notify-charset” attribute.

The Delivery Method Document specifies whether the Printer includes this attribute in an Event Notification.

~~The Printer MAY support this attribute. If a Printer supports a Delivery Method that requires this attribute, then the Printer MUST support this attribute~~

9 Event Notification Content

This section defines the Event Notification content that the Printer sends when an Event occurs.

When an Event occurs, the Printer MUST find each Subscription object whose “notify-events” attribute “matches” the Event. See section 5.3.2.2 for details on “matching”. For each matched Subscription Object, the Printer MUST create an Event Notification with the content and format that the Delivery Method Document specifies. The content contains the value of attributes specified by the Delivery Method Document. The Printer obtains the values immediately after the Event occurs. For example, if the “printer-state” attribute changes from ‘idle’ to ‘processing’, the Event ‘printer-state-changed’ occurs and the Printer puts various attributes into the Event Notification, including “printer-up-time” and “printer-state” with the values that they have immediately after the Event occurs, i.e., the value of “printer-state” is ‘processing’.

If two different Events occur simultaneously, or nearly so (e.g., “printer-up-time” has the same value for both), the Printer MUST create a separate Event Notification for each Event, even if the associated Subscription Object is the same for both Events. However, the Printer MAY combine these distinct Event Notifications into a single Compound Event Notification if the Delivery Method supports Compound Event Notifications For example, suppose that two nearly-simultaneously Events represent two successive ‘printer-state-changed’ Events, one from ‘idle’ to ‘processing’ and another from ‘processing’ to ‘stopped’. These two Events have the same name but are different instances of the Event. Then the Printer MUST create a separate Event Notification for each Event and SHOULD accurately report the “printer-state” of the first Event as ‘processing’ and the second Event as ‘stopped’.

~~If the same Event occurs several times in quick succession (e.g., ‘job-progress’), the Printer MUST create a separate Event Notification for each Event unless the Delivery Method Document specifies that the Event is moderated. Events might be moderated by a time interval (e.g., every 10 seconds) or by the number of Events (every 10th occurrence of the Event).~~

If a Subscription Object contains more than one Subscribed Event, and several ~~matching~~ Events occur in quick succession each matching a different Subscribed Event in the Subscription Object, the Printer MUST NOT generate a ~~separate~~ single Event Notification ~~for~~ from ~~each~~ several of these Events, but ~~. Depending on the Delivery Method, the Printer~~ MAY combine ~~several~~ distinct Event Notifications into a single C~~c~~ompound Event Notification if the Delivery Method supports Compound Event Notifications.

After the Printer has created the Event Notification, the Printer delivers it via either a:

Push Delivery Method: The Printer sends the Event Notification shortly after an Event occurs. For some Push Delivery Methods, the Notification Recipient MUST send a response; for others it MUST NOT send a response.

Pull Delivery Method: The Printer saves Event Notifications for some event-lease time and expects the Notification Recipient to request Event Notifications. The Printer returns the Event Notifications in a response to such a request.

If an error that meets the following conditions occurs, the Printer MUST cancel the Subscription Object.

a) the error occurs during the sending of an Event Notification generated from Subscription Object S AND

b) the error would continue to occur every time the Printer sends an Event Notification generated from Subscription Object S in the future.

From example, if the address of the "notify-recipient-uri" of Subscription Object A references a non-existent target and the Printer determines ~~that~~ this fact, it MUST delete Subscription Object A.

The next two sections describe the values that a Printer sends in the content of Machine Consumable and Human Consumable Event Notifications, respectively.

The tables in the sub-sections of this section contain the following columns:

a) **Source Value:** the name of the attribute that supplies the value for the Event Notification. Asterisks in this field refer to a note below the table.

b) **Sends:** if the Printer supports the value (column 1) on the Source Object (column 3) the Delivery Method MUST specify:

MUST: that the Printer MUST send the value.

SHOULD: either that the Printer MUST send the value or that the value is incompatible with the Delivery Method.

MAY: that the Printer MUST, SHOULD, MAY, MUST NOT, SHOULD NOT, or NEED NOT send the value. The Delivery Method specifies the level of conformance for the Printer.

c) **Source Object:** the object from which the source value comes. If the object is "Event Notification", the Printer fabricates the value when it sends the Event Notification. See section 8.

9.1 Content of Machine Consumable Event Notifications

This section defines the attributes that a Delivery Method MUST mention in a Delivery Method Document when specifying the Machine Consumable Event Notification's contents.

This document does not define the order of attributes in Event Notifications. However, Delivery Method Documents MAY define the order of some or all of the attributes.

A Delivery Method Document MUST specify additional attributes (if any) that a Printer implementation sends in a Machine Consumable Event Notification.

Notification Recipients MUST be able to accept Event 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.

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

a) for all Events

b) for Job Events only

c) for Printer Events only

9.1.1 Event Notification Content Common to All Events

This section lists the attributes that a Delivery Method MUST specify for all Events.

Table 5 lists potential values in each Event Notification.

Table 5 – Attributes in Event Notification Content

Source Value	**Sends**	**Source Object**
notify-subscription-id (integer(1:MAX))	MUST	Subscription
notify-printer-uri (uri)	MUST	Subscription
notify-subscribed-event (type2 keyword)	MUST	Event Notification
printer-up-time (integer(MIN:MAX))	MUST	Printer
printer-current-time (dateTime) *	MUST	Printer
notify-sequence-number (integer (0:MAX))	SHOULD	Subscription
notify-charset (charset)	SHOULD	Subscription
notify-natural-language (naturalLanguage)	SHOULD	Subscription
notify-user-data (octetString(63)) **	SHOULD	Subscription
notify-text (text)	SHOULD	Event Notification
attributes from the “notify-attributes” attribute ***	MAY	Printer
attributes from the “notify-attributes” attribute ***	MAY	Job
attributes from the “notify-attributes” attribute ***	MAY	Subscription

* A Printer MUST send this value only if and only if it supports the Printer’s “printer-current-time” attribute.

** If the Subscription Object does not contain a “notify-user-data” attribute and the Delivery Method document REQUIRES the Printer to send the “notify-user-data” source value in the Event Notification, the Printer MUST send an octet-string of length 0.

*** The last three rows represent additional attributes that a client MAY request via the “notify-attributes” attribute. A Printer MAY support the “notify-attributes” attribute. The Delivery Method MUST say that the Printer MUST, SHOULD, MAY, MUST NOT, SHOULD NOT, or NEED NOT support the “notify-attributes” attribute and specific values of this attribute. The Delivery Method MAY say that support for the “notify-attributes” is conditioned on support of the attribute by the Printer or it MAY say that Printer MUST support the “notify-attributes” attribute if the Printer supports the Delivery Method.

9.1.2 Additional Event Notification Content for Job Events

This section lists the additional attributes that a Delivery Method MUST specify for Job Events. See Table 6.

Table 6 – Additional Event Notification Content for Job Events

Source Value	Sends	Source Object
job-id (integer(1:MAX))	MUST	Job
job-state (type1 enum)	MUST	Job
job-state-reasons (1setOf type2 keyword)	MUST	Job
job-impressions-completed (integer(0:MAX)) *	MUST	Job

* The Printer MUST send the “job-impressions-completed” attribute in an Event Notification only for the combinations of Events and Subscribed Events shown in Table 7.

Table 7 – Combinations of Events and Subscribed Events for “job-impressions-completed”

Job Event	Subscribed Job Event
‘job-progress’	‘job-progress’
‘job-completed’	‘job-completed’
‘job-completed’	‘job-state-changed’

9.1.3 Additional Event Notification Content for Printer Events

This section lists the additional attributes that a Delivery Method MUST specify for Printer Events. See Table 8.

Table 8 – Additional Event Notification Content for Printer Events

Source Value	Sends	Source Object
printer-state (type1 enum)	MUST	Printer

Source Value	Sends	Source Object
printer-state-reasons (1setOf type2 keyword)	MUST	Printer
printer-is-accepting-jobs (boolean)	MUST	Printer

9.2 Content of Human Consumable Event Notification

This section defines the information that a Delivery Method MUST mention in a Delivery Method Document when specifying the Human Consumable Event Notifications contents or the value of the “notify-text” attribute.

Such a Delivery Method MUST specify the following information and a Printer SHOULD send it:

a) the Printer name (see Table 9)
b) the time of the Event (see Table 11)
c) for Printer Events only:
    i) the Event (see Table 10) and/or Printer state information (see Table 14)
d) for Job Events only:
    i) the job identity (see Table 12)
    ii) the Event (see Table 10) and/or Job state information (see Table 13)

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

A Delivery Method Document MUST specify additional information (if any) that a Printer implementation sends in a Human Consumable Event Notification or in the “notify-text” attribute.

A client MUST NOT request additional attributes via the “notify-attributes” attribute because this attribute works only for Machine Consumable Event Notifications.

Notification Recipients MUST NOT expect to be able to parse the Human Consumable Event Notification contents or the value of the “notify-text” attribute.

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

a) for all Events

b) for Job Events only

c) for Printer Events only

9.2.1 Event Notification Content Common to All Events

This section lists the source of the information that a Delivery Method MUST specify for all Events.

There is a separate table for each piece of information. Each row in the table represents a source value for the information and the values are listed in order of preference, with the first one being the preferred one. An implementation SHOULD use the source value from the earliest row in each table. It MAY use the source value from another row instead, or it MAY combine the source values from several rows. An implementation is free to determine the best way to present this information.

In all tables of this section, all rows contain a "MAY" in order to state that the Delivery Method specifies the conformance.~~ The tables in this section and following contain the following columns for each piece of information:~~

~~The tables in this section do not contain a "Sends" column because all rows would have a "**SHOULD**" as defined in section 0~~.

Table 9 lists the source of the information for the Printer Name. The "printer-name" is more user-friendly unless the Notification Recipient is in a place where the Printer name is not meaningful. For example, an implementation could have the intelligence to send the value of the "printer-name" attribute to a Notification Recipient that can access the Printer via value of the "printer-name" attribute and otherwise send the value of the "notify-printer-uri" attribute.

Table 9 – Printer Name in Event Notification Content

Source Value	**Sends**	**Source Object**
printer-name (name(127))	MAY	Printer
notify-printer-uri (uri)	MAY	Subscription

Table 10 lists the source of the information for the Event name. A Printer MAY combine this information with state information described for Jobs in Table 13 or for Printers in Table 14.

Table 10 – Event Name in Event Notification Content

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

Table 11 lists the source of the information for the time that the Event occurred. A Printer can send this value only if it supports the Printer's "printer-current-time" attribute. If a Printer does not support the "printer-current-time" attribute, it MUST NOT send the "printer-up-time" value instead, since it is not an allowed option for human consumable information.

Table 11 – Event Time in Event Notification Content

Source Value	**Sends**	**Source Object**
printer-current-time (dateTime)	MAY	Printer

9.2.2 Additional Event Notification Content for Job Events

This section lists the source of the additional information that a Delivery Method MUST specify for Job Events.

Table 12 lists the source of the information for the job name. The “job-name” is likely more meaningful to a user than “job-id”.

Table 12 – Job Name in Event Notification Content

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

Table 13 lists the source of the information for the job state. If a Printer supports the “job-state-message” and “job-detailed-state-message” attributes, it SHOULD use those attributes for the job state information, otherwise, it should fabricate such information from the “job-state” and “job-state-reasons”. For some Events, a Printer MAY combine this information with Event information.

Table 13 – Job State in Event Notification Content

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

9.2.3 Additional Event Notification Content for Printer Events

This section lists the source of the additional information that a Delivery Method MUST specify for Printer Events.

Table 14 lists the source of the information for the printer state. If a Printer supports the “printer-state-message”, it SHOULD use that attribute for the job state information, otherwise it SHOULD fabricate such information from the “printer-state” and “printer-state-reasons”. For some Events, a Printer MAY combine this information with Event information.

Table 14 – Printer State in Event Notification Content

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

Source Value	**Sends**	**Source Object**
printer-is-accepting-jobs (boolean)	MAY	Printer

10 Delivery Methods

A Delivery Method is the mechanism, i.e., protocol, by which the Printer delivers an Event Notification to a Notification Recipient. There are several potential Delivery Methods for Event Notifications, standardized, as well as proprietary. This document does not define any of these delivery mechanisms. Each Delivery Method MUST be defined in a Delivery Method Document that is separate from this document. New Delivery Methods will be created as needed using an extension to the registration procedures defined in [ipp-mod]. Such documents are registered with IANA (see section 13).

The following sorts of Delivery Methods are expected:

- The Notification Recipient polls for Event Notifications at intervals directed by the Printer

- The Printer sends Event Notifications to the Notification Recipient using http as the transport.

- The Printer sends an email message.

This section specifies how to define a Delivery Method Document and what to put in such a document.

A Delivery Method Document MUST contain an exact copy of the following paragraph, caption and table. In addition, column 2 of the table in the Delivery Method Document MUST contain answers to questions in column 1 for the Delivery Method. Also, the Delivery Method document MUST contain a reference to this document and call that reference [ipp-ntfy] because the table contains an [ipp-ntfy] reference.

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

Table 15 – Information about the Delivery Method

Document Method Conformance Requirement	**Delivery Method Realization**
1. What is the URL scheme name for the Delivery Method?	
2. Is the Delivery Method REQUIRED, RECOMMEND, or OPTIONAL for an IPP Printer to support?	
3. What transport and delivery protocols does the Printer use to deliver the Event Notification Content, i.e., what is the entire network stack?	

4. Can several Event Notifications be combined into a C~~c~~ompound Event Notification?	
5. Is the Delivery Method initiated by the Notification Recipient (pull), or by the Printer (push)?	
6. Is the Event Notification content Machine Consumable or Human Consumable?	
7. What section in this document answers the following question? For a Machine Consumable Event Notification, what is the representation and encoding of values defined in section 9.1 of [ipp-ntfy] and the conformance requirements thereof? For a Human Consumable Event Notification, what is the representation and encoding of pieces of information defined in section 9.2 of [ipp-ntfy] and the conformance requirements thereof?	
8. What are the latency and reliability of the transport and delivery protocol?	
9. What are the security aspects of the transport and delivery protocol, e.g., how it is handled in firewalls?	
10. What are the content length restrictions?	
11. What are the additional values or pieces of information that a Printer sends in an Event Notification content and the conformance requirements thereof?	
12. What are the additional Subscription Template and/or Subscription Description attributes and the conformance requirements thereof?	
13. What are the additional Printer Description attributes and the conformance requirements thereof?	

11 Operations for Notification

This section defines all of the operations for Notification. Section 7.1 assigns of the “operation-id” for each operation. The following two sub-sections define Subscription Creation Operations, and other operations.

11.1 Subscription Creation Operations

This section defines the Subscription Creation Operations. The first section on Create-Job-Subscriptions gives most of the information. The other Subscription Creation Operations refer to the section on Create-Job-Subscriptions, even though the Create-Job-Subscriptions operation is the only OPTIONAL operation in this document (see section 12).

A Printer MUST support Create-Printer-Subscriptions and the Subscription Template Attributes Group in Job Creation operations. It MAY support Create-Job-Subscriptions operations.

11.1.1 Create-Job-Subscriptions Operation

The operation creates one or more Per-Job Subscription Objects. The client supplies one or more Subscription Template Attributes Groups each containing one or more of Subscription Template Attributes (defined in section 5.3).

Except for errors, the Printer MUST create exactly one Per-Job Subscription Object from each Subscription Template Attributes Group in the request, even if the newly created Subscription Object would have identical behavior to some existing Subscription Object. The Printer MUST associate each newly created Per-Job Subscription Object with the target Job, which is specified by the “notify-job-id” operation attribute.

The Printer MUST accept the request in any of the target job’s ‘not-completed’ states, i.e., ‘pending’, ‘pending-held’, ‘processing’, or ‘processing-stopped’. The Printer MUST NOT change the job’s “job-state” attribute because of this operation. If the target job is in any of the ‘completed’ states, i.e., ‘completed’, ‘canceled’, or ‘aborted, then the Printer MUST reject the request and return the ‘client-error-not-possible’ status code; the response MUST NOT contain any Subscription Attribute Groups.

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 this Printer (see [IPP-MOD] sections 1 and 8.5). Otherwise the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

11.1.1.1 Create-Job-Subscriptions Request

The following groups of attributes are part of the Create-Job-Subscriptions Request:

Group 1: Operation Attributes

Natural Language and Character Set:

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

Target:

The “printer-uri” attribute which defines the target for this operation as described in [ipp-mod] section 3.1.5.

Requesting User Name:

The “requesting-user-name” attribute SHOULD be supplied by the client as described in [ipp-mod] section 8.3.

notify-job-id (integer(1:MAX)):

The client MUST supply this attribute and it MUST specify the Job object to associate the Per-Job Subscription with. The value of “notify-job-id” MUST be the value of the “job-id” of the associated Job object. If the client does not supply this attribute, the Printer MUST reject this request with a ‘client-error-bad-request’ status code.

Group 2-N: Subscription Template Attributes

For each occurrence of this group:

The client MUST supply one or more Subscription Template Attributes in any order. See section 5.3 for a description of each such attribute. See section 5.2 for details on processing these attributes.

11.1.1.2 Create-Job-Subscriptions Response

The Printer MUST return to the client the following sets of attributes as part of a Create-Job-Subscriptions response:

Group 1: Operation Attributes

Status Message:

As defined in [ipp-mod].

The Printer can return any status codes defined in [ipp-mod] and section 16. The following is a description of the important status codes:

successful-ok: the Printer created all Subscription Objects requested.
successful-ok-ignored-subscriptions: the Printer created some Subscription Objects requested but some failed. The Subscription Attributes Groups with a “notify-status-code” attribute are the ones that failed.
client-error-ignored-all-subscriptions: the Printer created no Subscription Objects requested and all failed. The Subscription Attributes Groups with a “notify-status-code” attribute are the ones that failed
client-error-not-possible: For this operation and other Per-Job Subscription operations, this error can occur because the specified Job has already completed.

Natural Language and Character Set:

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

Group 2: Unsupported Attributes

See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes. This group does not contain any unsupported Subscription Template Attributes; they are returned in the Subscription Attributes Group (see below).

Group 3-N: Subscription Attributes

These groups MUST be returned if and only if the “status-code” parameter returned in Group 1 has the values: ‘successful-ok’, ‘successful-ok-ignored-subscriptions’, or ‘client-error-ignored-all-subscriptions’.

See section 5.2 for details on the contents of each occurrence of this group.

11.1.2 Create-Printer-Subscriptions operation

The operation is identical to Create-Job-Subscriptions with exceptions noted in this section.

The operation creates Per-Printer Subscription Objects instead of Per-Job Subscription Objects, and associates each newly created Per-Printer Subscription Object with the Printer specified by the operation target rather than with a specific Job.

The Printer MUST accept the request in any of its states, i.e., ‘idle’, ‘processing’, or ‘stopped’. The Printer MUST NOT change its “printer-state” attribute because of this operation.

Access Rights: To create Per-Printer Subscription Objects, the authenticated user (see [IPP-MOD] section 8.3) performing this operation MUST have Operator or Administrator access rights for this Printer (see [IPP-MOD] sections 1 and 8.5). Otherwise, the Printer MUST reject the operation and return: the ‘client-error-forbidden’, ‘client-error-not-authenticated’, or ‘client-error-not-authorized’ status code as appropriate.

11.1.2.1 Create-Printer-Subscriptions Request

The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.1) except that the Operation Attributes group MUST NOT contain the “notify-job-id” attribute. If the client does supply the “notify-job-id” attribute, then the Printer MUST treat it as any other unsupported Operation attribute and MUST return it in the Unsupported Attributes group.

11.1.2.2 Create-Printer-Subscriptions Response

The groups are identical to the Create-Job-Subscriptions (see section 11.1.1.2).

11.1.3 Job Creation Operation – Extensions for Notification

This document extends the Job Creation operations to create Subscription Objects as a part of the operation.

The operation is identical to Create-Job-Subscriptions with exceptions noted in this section.

Unlike the Create-Job-Subscriptions operation, this operation associates the newly created Subscription Objects with the Job object created by this operation. The operation succeeds if and only if the Job creation succeeds. If the Printer does not create some or all of the requested Subscription Objects, the Printer MUST return a 'successful-ok-ignored-subscriptions' status-code instead of a 'successful-ok' status-code, but the Printer MUST NOT reject the operation because of a failure to create Subscription Objects.

If the operation includes a Job Template group, the client MUST supply it after the Operation Attributes group and before the first Subscription Template Attributes Group.

If a Printer does not support this Notification specification, then it MUST treat the Subscription Attributes Group like an unknown group and ignore it (see [ipp-mod] section 5.2.2). Because the Printer ignores the Subscription Attributes Group, it doesn't return them in the response either, thus indicating to the client that the Printer doesn't support Notification.

Access Rights: To create Per-Job Subscription Objects, the authenticated user (see [IPP-MOD] section 8.3) performing this operation MUST either have permission to create Jobs on the Printer. Otherwise the Printer MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' status code as appropriate.

11.1.3.1 Job Creation Request

The groups for this operation are sufficiently different from the Create-Job-Subscriptions operation that they are all presented here. The following groups of attributes are supplied as part of a Job Creation Request:

Group 1: Operation Attributes

Same as defined in [ipp-mod] for Print-Job, Print-URI, and Create-Job requests.

Group 2: Job Template Attributes

The client OPTIONALLY supplies a set of Job Template attributes as defined in [ipp-mod] section 4.2.

Group 3 to N: Subscription Template Attributes

The same as Group 2-N in Create-Job-Subscriptions. See section 11.1.1.1.

Group N+1: Document Content (Print-Job only)

The client MUST supply the document data to be processed.

11.1.3.2 Job Creation Response

The Printer MUST return to the client the following sets of attributes as part of a Print-Job, Print-URI, and Create-Job Response:

Group 1: Operation Attributes

Status Message:

As defined in [ipp-mod] for Print-Job, Print-URI, and Create-Job requests.

The Printer can return any status codes defined in [ipp-mod] and section 16. The following is a description of the important status codes:

successful-ok: the Printer created the Job and all Subscription Objects requested.

successful-ok-ignored-subscriptions: the Printer created the Job and not all of the Subscription Objects requested. This status-code hides ‘successful-ok-xxx’ status-codes that could reveal problems in Job creation. The Printer MUST not return the ‘client-error-ignored-all-subscriptions’ status code for Job Creation operations because the Printer returns an error status-code only when it fails to create a Job.

Natural Language and Character Set:

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

Group 2: Unsupported Attributes

See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes. This group does not contain any unsupported Subscription Template Attributes; they are returned in the Subscription Attributes Group (see below).

Group 3: Job Object Attributes

As defined in [ipp-mod] for Print-Job, Print-URI, and Create-Job requests.

Group 4 to N: Subscription Attributes

These groups MUST be returned if and only if the client supplied Subscription Template Attributes and the operation was accepted.

See section 5.2 for details on the contents of each occurrence of this group.

11.2 Other Operations

This section defines other operations on Subscription objects.

11.2.1 Validate-Job Operation - Extensions for Notification

A client can test whether one or more Subscription Objects could be created using the Validate-Job operation. The client supplies one or more Subscription Template Attributes Groups (defined in section 5.3), just as in a Job Creation request.

A Printer MUST support this extension to this operation.

The Printer MUST accept requests that are identical to the Job Creation request defined in section 11.1.3.1, except that the request MUST not contain document data.

The Printer MUST return the same groups and attributes as the Print-Job operation (section 11.1.3.1) with the following exceptions. The Printer MUST NOT return a Job Object Attributes Group because no Job is created. The Printer MUST NOT return the "notify-subscription-id" attribute in any Subscription Attribute Group because no Subscription Object is created.

If the Printer would succeed in creating a Subscription Object, the corresponding Subscription Attributes Group either has no 'status-code' attribute or a 'status-code' attribute with a value of 'successful-ok-too-many-events' or 'successful-ok-ignored-or-substituted-attributes' (see sections 5.2 and 17). The status-codes have the same meaning as in Job Creation except the results state what "would happen".

The Printer MUST validate Subscription Template Attributes Groups in the same manner as the Job Creation operations. ~~However, to cause the Printer to validate as many Subscription Template Attributes as possible, the Printer MUST assume that is can create up to the number of Subscription Objects equal to the value of "notify-max-job-subscriptions-supported".~~

11.2.2 Get-Printer-Attributes - Extensions for Notification

This operation is extended so that it returns Printer attributes defined in this document.

A Printer MUST support this extension to this operation.

In addition to the requirements of [ipp-mod] section 3.2.5, a Printer MUST support the following additional values for the "requested-attributes" Operation attribute in this operation and return such attributes in the Printer Object Attributes group of its response.

1. **Subscription Template Attributes:** Each supported attribute in column 2 of Table 1.

2. **New Printer Description Attributes:** Each supported attribute in section 6.

3. **New Group Name:** The 'subscription-template' group name, which names all supported Subscription Template Attribute in column 2 of Table 1. This group name is also used in the Get-Subscription-Attributes and Get-Subscriptions operation with an analogous meaning.

4. **Extended Group Name:** The 'all' group name, which names all Printer attributes according to [ipp-mod] section 3.2.5. In this extension 'all' names all attributes specified in [ipp-mod] plus those named in items 1 and 2 of this list.

11.2.3 Get-Subscription-Attributes operation

This operation allows a client to request the values of the attributes of a Subscription Object.

A Printer MUST support this operation.

This operation is almost identical to the Get-Job-Attributes operation (see [ipp-mod] section 3.3.4). The only differences are that the operation is directed at a Subscription Object rather than a Job object, and the returned attribute group contains Subscription Object attributes rather than Job object attributes.

11.2.3.1 Get-Subscription-Attributes Request

The following groups of attributes are part of the Get-Subscription-Attributes request:

Group 1: Operation Attributes

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as described in section [ipp-mod] 3.1.4.1.

Target:
The "printer-uri" attribute which defines the target for this operation as described in [ipp-mod] section 3.1.5.

"notify-subscription-id" (integer (1:MAX)):
The client MUST supply this attribute. The Printer MUST support this attribute. This attribute specifies the Subscription Object from which the client is requesting attributes. If the client omits this attribute, the Printer MUST reject this request with the 'client-error-bad-request' status code.

Requesting User Name:
The "requesting-user-name" attribute SHOULD be supplied by the client as described in [ipp-mod] section 8.3.

"requested-attributes" (1setOf keyword):
The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. This attribute specifies the attributes of the specified Subscription Object that the Printer MUST return in the response. Each value of this attribute is either an attribute name (defined in sections 5.3 and 5.4) or an attribute group name. The attribute group names are:

- 'subscription-template': all attributes that are both defined in section 5.3 and present on the specified Subscription Object (column 1 of Table 1).
- 'subscription-description': all attributes that are both defined in section 5.4 and present on the specified Subscription Object (Table 2).
- 'all': all attributes that are present on the specified Subscription Object.

A Printer MUST support all these group names.

If the client omits this attribute, the Printer MUST respond as if this attribute had been supplied with a value of ‘all’.

11.2.3.2 Get-Subscription-Attributes Response

The Printer returns the following sets of attributes as part of the Get-Subscription-Attributes Response:

Group 1: Operation Attributes

Status Message:
Same as [ipp-mod].

Natural Language and Character Set:
The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod] section 3.1.4.2. The “attributes-natural-language” MAY be the natural language of the Subscription Object, rather than the one requested.

Group 2: Unsupported Attributes

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

The response NEED NOT contain the “requested-attributes” operation attribute with any supplied values (attribute keywords) that were requested by the client but are not supported by the Printer. If the Printer does return unsupported attributes referenced in the “requested-attributes” operation attribute and that attribute included group names, such as ‘all’, the unsupported attributes MUST NOT include attributes described in the standard but not supported by the implementation.

Group 3: Subscription Attributes

This group contains a set of attributes with their current values. Each attribute in this group:

a) MUST be specified by the “requested-attributes” attribute in the request, AND

b) MUST be present on the specified Subscription Object AND

c) MUST NOT be restricted by the security policy in force. For example, a Printer MAY prohibit a client who is not the creator of a Subscription Object from seeing some or all of its attributes. See [ipp-mod] section 8.

The Printer can return the attributes of the Subscription Object in any order. The client MUST accept the attributes in any order.

11.2.4 Get-Subscriptions operation

This operation allows a client to retrieve the values of attributes of all Subscription Objects belonging to a Job or Printer.

A Printer MUST supported this operation.

This operation is similar to the Get-Subscription-Attributes operation, except that this Get-Subscriptions operation returns attributes from possibly more than one object.

This operation is similar to the Get-Jobs operation (see [ipp-mod] section 3.2.6), except that the operation returns Subscription Objects rather than Job objects.

11.2.4.1 Get-Subscriptions Request

The following groups of attributes are part of the Get-Subscriptions request:

Group 1: Operation Attributes

Natural Language and Character Set:
The “attributes-charset” and “attributes-natural-language” attributes as described in [ipp-mod] section 3.1.4.1.

Target:
The “printer-uri” attribute which defines the target for this operation as described in [ipp-mod] section 3.1.5.

Requesting User Name:
The “requesting-user-name” attribute SHOULD be supplied by the client as described in [ipp-mod] section 8.3.

“notify-job-id” (integer(1:MAX)):
If the client specifies this attribute, the Printer returns the specified attributes of all Per-Job Subscription Objects associated with the Job whose “job-id” attribute value equals the value of this attribute. If the client does not specify this attribute, the Printer returns the specified attributes of all Per-Printer Subscription Objects. Note: there is no way to get all Per-Job Subscriptions.

“limit” (integer(1:MAX)):
The client OPTIONALLY supplies this attribute. The Printer 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 the “my-subscriptions” attribute constrains which Subscription Objects are returned. The limit is a “stateless limit” in that if the value supplied by the client is ‘N’, then only the first ‘N’ Subscription Objects are returned in the Get-Subscriptions Response. There is no mechanism to allow for the next ‘M’ Subscription Objects after the first ‘N’ Subscription Objects. If the client does not supply this attribute, the Printer responds with all applicable Subscription Objects.

“requested-attributes” (1setOf type2 keyword):
The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. This attribute specifies the attributes of the specified Subscription Objects that the Printer MUST return in the response. Each value of this attribute is either an attribute name (defined in sections 5.3 and

5.4) or an attribute group name (defined in section 11.2.3.1). If the client omits this attribute, the Printer MUST respond as if the client had supplied this attribute with the one value: 'notify-subscription-id'.

"my-subscriptions" (boolean):

The client OPTIONALLY supplies this attribute. The Printer MUST support this attribute. If the value is 'false', the Printer MUST consider the Subscription Objects from all users as candidates. If the value is 'true', the Printer MUST return the Subscription Objects created by the requesting user of this request. If the client does not supply this attribute, the Printer MUST respond as if the client had supplied the attribute with a value of 'false'. The means for authenticating the requesting user and matching the Subscription Objects is similar to that for Jobs which is described in [ipp-mod] section 8.

11.2.4.2 Get-Subscriptions Response

The Printer returns the following sets of attributes as part of the Get-Subscriptions Response:

Group 1: Operation Attributes

Status Message:

Same as [ipp-mod].

Natural Language and Character Set:

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

Group 2: Unsupported Attributes

Same as for Get-Subscription-Attributes.

Groups 3 to N: Subscription Attributes

The Printer responds with one Subscription Attributes Group for each requested Subscription Object (see the "notify-job-id" attribute in the Operation Attributes Group of this operation).

The Printer returns Subscription Objects in any order.

If the "limit" attribute is present in the Operation Attributes group of the request, the number of Subscription Attributes Groups in the response MUST NOT exceed the value of the "limit" attribute.

It there are no Subscription Objects associated with the specified Job or Printer, the Printer MUST return zero Subscription Attributes Groups and it MUST NOT treat this case as an error, i.e., the status-code MUST be 'successful-ok' unless something else causes the status code to have some other value.

See the Group 3 response (Subscription Attributes Group) of the Get-Subscription-Attributes operation (section 11.2.3.2) for the attributes that a Printer returns in this group.

11.2.5 Renew-Subscription operation

This operation allows a client to request the Printer to extend the lease on a Per-Printer Subscription Object.

The Printer MUST support this operation.

The Printer MUST accept this request for a Per-Printer Subscription Object in any of the target Printer's states, i.e., 'idle', 'processing', or 'stopped', but MUST NOT change the Printer's "printer-state" attribute.

The Printer MUST reject this request for a Per-Job Subscription Object because it has no lease (see section 5.4.3). The status code returned MUST be 'client-error-not-possible'.

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

11.2.5.1 Renew-Subscription Request

The following groups of attributes are part of the Renew-Subscription Request:

Group 1: Operation Attributes

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod] section 3.1.4.1.

Target:
The "printer-uri" attribute which defines the target for this operation as described in [ipp-mod] section 3.1.5.

"notify-subscription-id" (integer (1:MAX)):
The client MUST supply this attribute. The Printer MUST support this attribute. This attribute specifies the Per-Printer Subscription Object whose lease the Printer MUST renew. If the client omits this attribute, the Printer MUST reject this request with the 'client-error-bad-request' status code.

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

Group 2: Subscription Template Attributes "notify-lease-duration" (integer(0:MAX)):

The client MAY supply this attribute. It indicates the number of seconds to renew the lease for the specified Subscription Object. A value of 0 requests an infinite lease (which MAY require Operator access rights). If the client omits this attribute, the Printer MUST use the value of the Printer's "notify-lease-duration-default" attribute. See section 5.3.7 for more details.

11.2.5.2 Renew-Subscription Response

The Printer returns the following sets of attributes as part of the Renew-Subscription Response:

Group 1: Operation Attributes

Status Message:

Same as [ipp-mod].

The following are some of the status codes returned:

successful-ok: The operation successfully renewed the lease on the Subscription Object for the requested duration..

successful-ok-ignored-or-substituted-attributes: The operation successfully renewed the lease on the Subscription Object for some duration other than the amount requested.

client-error-not-possible: The operation failed because the "notify-subscription-id" Operation attribute identified a Per-Job Subscription Object.

client-error-not-found: The operation failed because the "notify-subscription-id" Operation attribute identified a non-existent Subscription Object.

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod] section 3.1.4.2. The "attributes-natural-language" MAY be the natural language of the Subscription Object, rather than the one requested.

Group 2: Unsupported Attributes

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

Group 3: Subscription Attributes

The Printer MUST return the following Subscription Attribute:

"notify-lease-duration" (integer(0:MAX)):

The value of this attribute MUST be the number of seconds that the Printer has granted for the lease of the Subscription Object (see section 5.3.7 for details, such as the value of this attribute when the Printer doesn't support the requested value).

11.2.6 Cancel-Subscription operation

This operation allows a client to delete a Subscription Object and stop the Printer from sending more Event Notifications. Once performed, there is no way to reference the Subscription Object.

A Printer MUST supported this operation.

The Printer MUST accept this request in any of the target Printer's states, i.e., 'idle', 'processing', or 'stopped', but MUST NOT change the Printer's "printer-state" attribute.

If the specified Subscription Object is a Per-Job Subscription Object, the Printer MUST accept this request in any of the target Job's states, but MUST NOT change the Job's "job-state" attribute or affect the Job.

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 (see [IPP-MOD] sections 1 and 8.5). Otherwise, the Printer MUST reject the operation and return: the 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' status code as appropriate.

Note: There is no way to change any attributes on a Subscription Object, except the "notify-lease-duration" attribute (using the Renew-Subscription operation). In order to change other attributes, a client performs a Subscription Creation Operation and Cancel-Subscription operation on the old Subscription Object. If the client wants to avoid missing Event Notifications, it performs the Subscription Creation Operation first. If this order would create too many Subscription Objects on the Printer, the client reverses the order.

11.2.6.1 Cancel-Subscription Request

The following groups of attributes are part of the Cancel-Subscription Request:

Group 1: Operation Attributes

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod] section 3.1.4.1.

Target:
The "printer-uri" attribute which defines the target for this operation as described in [ipp-mod] section 3.1.5.

"notify-subscription-id" (integer (1:MAX)):
The client MUST supply this attribute. The Printer MUST support this attribute. This attribute specifies the Subscription Object that the Printer MUST cancel. If the client omits this attribute, the Printer MUST reject this request with the 'client-error-bad-request' status code.

Requesting User Name:
The "requesting-user-name" attribute SHOULD be supplied by the client as described in [ipp-mod] section 8.3.

11.2.6.2 Cancel-Subscription Response

The Printer returns the following sets of attributes as part of the Cancel-Subscription Response:

Group 1: Operation Attributes

Status Message:
Same as [ipp-mod].

The following are some of the status codes returned:

successful-ok: The operation successfully canceled (deleted) the Subscription Object..
client-error-not-found: The operation failed because the "notify-subscription-id" Operation attribute identified a non-existent Subscription Object.

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod] section 3.1.4.2. The "attributes-natural-language" MAY be the natural language of the Subscription Object, rather than the one requested.

Group 2: Unsupported Attributes

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

12 Conformance Requirements

It is OPTIONAL to implement this Event Notification specification.

If this Event Notification specification is implemented, Printers MUST:

1. meet the Conformance Requirements detailed in section 5 of [ipp-mod].
2. support all of the following attributes:
    a. REQUIRED Subscription Object attributes in section 5.
    b. REQUIRED Printer Description object attributes in section 6.
    c. REQUIRED attributes in Event Notification content in section 8.
3. send Event Notifications that conform to the requirements of the Delivery Method Document for each supported Delivery Method (the conformance requirements for Delivery Method Documents is specified in section 10).
4. support all operations as described in Table 16:

Table 16 – Conformance Requirements for Operations

Attribute	**Conformance requirements**
Subscription Attributes Group	REQUIRED
Create-Printer-Subscriptions (section 11.1.2)	REQUIRED
Create-Job-Subscriptions (section 11.1.1)	OPTIONAL
Get-Subscription-Attributes (section 11.2.2)	REQUIRED
Get-Subscriptions (section 11.2.4)	REQUIRED
Renew-Subscription (section 11.2.5)	REQUIRED
Cancel-Subscription (section 11.2.6)	REQUIRED

13 IANA Considerations

This section describes the procedures for registering Event Notification Delivery Method proposals with IANA to be used with this document. Such Delivery Method proposals can be IETF standards track documents or vendor-defined documents. In either case, they will be registered with IANA using procedures that extend those defined in [ipp-mod] section 6 and 11.

These extension procedures are aligned with the guidelines as set forth by the IESG [IANA-CON]. Section 13.1 defines the format and content for new registrations for consideration. IANA will reject registration proposals that leave out required information or do not follow the appropriate format described in Section 13.1.

Implementers can, at any time, define new Event Notification Delivery Methods by proposing the complete specification to IANA:

iana@iana.org

or by filling out the appropriate form on the IANA web pages (http://www.iana.org).

IANA will forward the registration proposal to the IPP Designated Expert who will review the proposal with a mailing list that the Designated Expert keeps for this purpose. Initially, that list will be the mailing list used by the IPP WG:

ipp@pwg.org

even after the IPP WG is disbanded as permitted by [IANA-CON]. The IPP Designated Expert is appointed by the IESG Area Director responsible for IPP, according to [IANA-CON].

When a Delivery Method Document is approved, the IPP Designated Expert becomes the point of contact for any future maintenance that might be required for that registration.

13.1 Format and Requirements for IPP Delivery Method Registration Proposals

This section defines the format and requirements for an IPP Event Notification Delivery Method Registration Proposal. A Delivery Method Registration Proposal:

1. MUST contain the following information:

   Type of registration: IPP Event Notification Delivery Method
   Name of this delivery method:
   Proposed URL scheme name of this delivery method:
   Name of proposer:
   Address of proposer:
   Email address of proposer:
   Is this delivery method REQUIRED or OPTIONAL for conformance to the IPP Event Notification Specification document:
   Is this delivery method defining Machine Consumable and/or Human Consumable content:

2. MUST meet the conformance requirements for Delivery Method Documents specified in section 10.

14 Internationalization Considerations

This IPP Notification specification continues support for the internationalization of [ipp-mod] of attributes containing text strings and names. Allowing a Subscribing Client to specify a different natural language and charset for each Subscription Object increases the internationalization support.

The Printer MUST be able to localize the content of Human Consumable Event Notifications and to localize the value of “notify-text” attribute in Machine Consumable Event Notifications that it sends to Notification Recipients. For localization, the Printer MUST use the value of the “notify-charset” attribute and the “notify-natural-language” attribute in the Subscription Object supplied by the Subscribing Client.

15 Security Considerations

By far the biggest security concern is the abuse of notification: sending unwanted Event 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 Event 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 Subscription Object is associated with a user. Only the creator or an Operator can cancel the Subscription Object. The system may limit the listing of items to only those items owned by the user. Some Subscription Objects (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 Delivery Method. IPP should use the security mechanism of the Delivery Method used. Some delivery mechanisms are more secure than others. Therefore, sensitive Event Notifications should use the Delivery Method that has the strongest security.

16 Status Codes

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

16.1 successful-ok-ignored-subscriptions (0x0003)

The Subscription Creation Operation was unable to create all requested Subscription Objects.

For a Create-Job-Subscriptions or Create-Printer-Subscriptions operation, this status code means that the Printer created one or more Subscription Objects, but not all requested Subscription Objects.

For a Job Creation operation, this status code means that the Printer created the Job along with zero or more Subscription Objects. The Printer returns this status code 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'.

16.2 client-error-ignored-all-subscriptions (0x0414)

This status code is the same as 'successful-ok-ignored-subscriptions' except that only the Create-Job-Subscriptions and Create-Printer-Subscriptions operation return it. They return this status code only when the Printer creates zero Subscription Objects.

17 Status Codes in Subscription Attributes Groups

This section contains values of the "notify-status-code" attribute that the Printer returns in a Subscription Attributes Group in a response when the corresponding Subscription Object:

1. is not created or

2. is created and some of the client-supplied attributes are not supported.

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

17.1 client-error-uri-scheme-not-supported (0x040C)

This status code is defined in [ipp-mod]. This document extends its meaning and allows it to be in a Subscription Attributes Group of a response.

The scheme of the client-supplied URI in a "notify-recipient-uri" Subscription Template Attribute in a Subscription Creation Operation is not supported. See section 5.3.1.

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

The number of Subscription Objects supported by the Printer would be exceeded if this Subscription Object were created (see section 5.2).

17.3 successful-ok-too-many-events (0x0005)

The client supplied more Events in the "notify-events" operation attribute of a Subscription Creation Operation than the Printer supports, as indicated in its "notify-max-events-supported" Printer attribute (see section 5.3.2).

17.4 successful-ok-ignored-or-substituted-attributes (0x0001)

This status code is defined in [ipp-mod]. This document extends its meaning to include unsupported Subscription Template Attributes and it can appear in a Subscription Attributes Group.

18 Encodings of Additional Attribute Tags

This section assigns values to two attributes tags as extensions to the encoding defined in [ipp-pro]).

The "subscription-attributes-tag" delimits Subscription Template Attributes Groups in requests and Subscription Attributes Groups in responses.

The "event-notification-attributes-tag" delimits Event Notifications in Delivery Methods that use an IPP-like encoding.

The following table specifies the values for the delimiter tags:

Tag Value (Hex)	**Meaning**
0x06	“subscription-attributes-tag”
0x07	“event-notification-attributes-tag”

19 References

[IANA-CON]
Narte, T. and Alvestrand, H.T.: Guidelines for Writing an IANA Considerations Section in RFCs, Work in Progress, draft-iesg-iana-considerations-04.txt, May 21, 1998.

[ipp-mod]
deBry, R., , Hastings, T., Herriot, R., Isaacson, S., Powell, P., “Internet Printing Protocol/1.1: Model and Semantics”, <draft-ietf-ipp-model-v11-07.txt>, work in progress, May 22, 2000.

[ipp-not-req]
deBry, R., Lewis, H., Hastings, T., “Internet Printing Protocol/1.1: Requirements for IPP Notifications”, <draft-ietf-ipp-not-04~~3~~.txt>, work in progress, ~~August 11~~July 6, 2000~~, 1999~~.

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

[ipp-prog]
Hastings, T., Bergman, R., Lewis, H., “~~Proposed~~ IPP Job Progress Attributes ~~for IPP~~”, <draft-ietf-ipp-job-prog-00.txt> work in progress, ~~February 2~~July 6, 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-02~~1~~.txt>, work in progress, March 23~~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.~~

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

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

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

[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.

20 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

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

Phone: 650-813-7696
Fax: 650-813-6860
Email: robert.herriot@pahv.xerox.com

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 - Model for Notification with Cascading Printers

With this model (see Figure 2), there is an intervening Print server between the human user and the output-device. So the system effectively has two Printers. There are two cases to consider.

1. When the Printer 1 (in the server) generates Events, the system behaves like the client and Printer in Figure 1. In this case, Printer 1 sends Event Notifications that are shown as Event Notifications (A) of Figure 2,.

2. When the Printer 2 (in the output-device) generates Events, there are two possible system configurations:

   a) Printer 1 forwards the client-supplied Subscription Creation Operations to the downstream Printer 2 and lets Printer 2 send the Event Notifications directly to the Notification Recipients supplied by the Client (Event Notifications(C) in the diagram).

   b) Printer 1 performs the client-supplied Subscription Creation Operations and also forwards the Subscription Creation Operations to Printer 2 with the Notification Recipient changed to be the Printer 1. When an Event occurs in Printer 2, Printer 2 sends the Event Notification (B) to Notification Recipient of Printer 1, which relays the received Event Notification (B) to the client-supplied Notification Recipient (as Event Notifications(A) in the diagram). Note, when a client performs a Subscription Creation Operation, Printer 1 need not forward the Subscription Creation Operation to Printer 2 if it would create a duplicate Subscription Object on Printer 2.

Note: when Printer 1 is forwarding Subscription Creation Operations to Printer 2, it may request Printer 2 to create additional Subscription Objects (called "piggy-backing"). Piggy-backing is useful when:

- Device A is configured to accept (IPP or non-IPP) requests from other servers.
- Server S wants to receive Job Events that the client didn't request and Server S wants these Events for jobs it submits and not for other jobs.

```
                           server S                       device A
                        +------------+                 +------------+
                        |            |                 |            |
+--------+ Subscription | ###########|                 | ###########|
| client |--Creation ---->#Printer #|  Subscription   | # Printer #|
+--------+  Operation   | # Object 1#|---Creation------|># Object 2#|
                        | ###|#######|   Operation     | ####|#|####|
                        +----|---^---+                 +-----|-|----+
+--------+     Event         |   |                           | |
|Notific-|<-Notifications(A)-+   +-- Event Notifications(B)--+ |
|ation Re|<-------------Event Notifications(C)-----------------+
|cipient |
+--------+
```

Figure 2 – Model for Notification with Cascading Printers

B. Appendix - Distributed Model for Notification

A Printer implementation could use some other remote notification service to provide some or most of the service. For example, the remote notification service could send Event Notifications using Delivery Methods that are not directly supported by the output device or server. Or, the remote notification service could store Subscription Objects (passed to it from the output device in response to Subscription Creation requests), accept Events, format the Event Notification in the natural language of the Notification Recipient, and send the Event Notifications to the Notification Recipient(s).

Figure 3 shows this partitioning. The interface between the output device (or server) and the remote notification service is outside the scope of this document and is intended to be transparent to the client and this document. The combination of the output device (or server) and the notification service together constitute an IPP Printer conforming to this Notification document.

```
                                         ***********************
                                         *
                                         * Printer (including
                                         * the distributed
                                         * Notification Service)
                                         *
                                         * output device or server
                                         * +---------------+
   PDA, desktop, or server               * +  ###########  +
        +--------+                       * |  # partial #  |
        | client |---IPP Subscription-------->#  Printer #  |
        +--------+   Creation operation  * |  # Object  #  |
                                         * |  #####|#####  |
                                         * +-------|-------+
                                         *         | Subscriptions
                                         *         | OR Event
     +------------+                      *         | Notifications
     |Notification|   IPP-defined        *  +------v--------+
     |Recipient   |<--Event Notifications---| Notification  |
     +------------+                      *  | Service       |
                                         *  +---------------+
                                         *
                                         *************************
     *** = Implementation configuration opaque boundary
```

Figure 3 – Opaque Use of a Notification Service Transparent to the Client

C. Appendix - Extended Notification Recipient

The model allows for an extended Notification Recipient that is itself a notification service that forwards each Event Notification to another recipient (called the Ultimate Notification Recipient in this section). The Delivery Method to the Ultimate Recipient is probably different from the Delivery Method used by the Printer to the extended Notification Recipient.

This extended Notification Recipient is transparent to the Printer but not to the client.

When a client performs a Subscription Creation Operation, it specifies the extended Notification Recipient as it would any Notification Recipient. In addition, the client specifies the Ultimate Notification Recipient in the Subscription Creation Operation in a manner specified by the extended Notification Recipient. Typically, it is either some bytes in the value of “notify-user-data” or some additional parameter in the value of “notify-recipient-uri”. The client also subscribes directly with the extended Notification Recipient (by means outside this document), since it is a notification service in its own right.

The IPP Printer treats the extended Notification Recipient like any other Notification Recipient and the IPP Printer is not aware of the forwarding. The Delivery Method that the extended Notification Recipient uses for delivering the Event Notification to the Ultimate Notification Recipient is beyond the scope of this document and is transparent to the IPP Printer.

Examples of this extended Notification Recipient are paging, immediate messaging services, general notification services, and NOS vendors’ infrastructure. Figure 4 shows this approach.

```
   PDA, desktop, or server                    server or output device
                                                   +---------------+
       +--------+                                  |  ###########  |
       | client |---Subscription Creation ----------->#  Printer #  |
       +--------+       Operation                  |  # Object  #  |
                                                   |  #####|#####  |
+------------+     +------------+   IPP-defined    +-------|-------+
|Ultimate    | any |Notification|<--Event Notifications----+
|Notification|<----|Recipient   |
|Recipient   |     +------------+
+------------+     (Notification Service)
```

Figure 4 – Use of an Extended Notification Recipient transparent to the Printer

D. Appendix - Details about Conformance Terminology

The following paragraph provide more details about conformance terminology.

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 RECOMMENDED 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.

E. Appendix - Object Model for Notification

This section describes the Notification object model that adds a 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 5 – 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.

E.1 Appendix - Object relationships

This sub-section defines the object relationships between the Printer, Job, and Subscription Objects by example. 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:

E.2 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. Each Per-Printer Subscription Object (s1, s2, and s3) is contained in (or is associated with) exactly one Printer object (p1).

E.3 Job Object and Per-Job Subscription Objects

1. A Job object (j1, j2, j3) 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. Each Per-Job Subscription Object is associated with exactly one Job object.

F. Appendix - Per-Job versus Per-Printer Subscription Objects

Per-Job and Per-Printer Subscription Objects are quite similar. Either type of Subscription Object can subscribe to Job Events, Printer Events, or both. Both types of Subscription Objects can be queried using the Get-Subscriptions and Get-Subscription-Attributes operations and canceled using the Cancel-Subscription operation. Both types of Subscription Objects create Subscription Objects which have the same Subscription Object attributes defined. However, there are some semantic differences between Per-Job Subscription Objects and Per-Printer Subscription Objects. A Per-Job Subscription Object is established by the client when submitting a job and after creating the job using the Create-Job-Subscriptions operation by specifying the “job-id” of the Job with the “notify-job-id” attribute. A Per-Printer Subscription Object is established between a client and a Printer using the Create-Printer-Subscriptions operation. Some specific differences are:

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

2. For Per-Job Subscription Objects, the Subscription Object is only valid while the job is “not-complete” (see sections 5.4.3) while for the Per-Printer Subscription Objects, the Subscription Object is valid until the time (in seconds) that the Printer returned in the “notify-lease-expiration-time” operation attribute.

3. Job Events in a Per-Job Subscription Object apply only to “one job” (the Job created by the Job Creation operation or references by the Create-Job-Subscriptions operation) while Job Events in a Per-Printer Subscription Object apply to ALL jobs contained in the IPP Printer.

G. Appendix: Change History (to be removed for Internet-Draft)

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.

G.1 Changes to the June 30, 2000 version to create the July 13, 2000 version

The following changes were made to the June 30, 2000 version to create the July 13, 2000 version based on the agreements reached at the July IPP WG meeting:

1. Deleted the “notify-max-job-subscriptions” and “notify-max-printer-subscriptions” Printer Description attributes, since the maximum cannot be guaranteed.

2. Added the “notify-time-interval (integer(0:MAX)) Subscription Template attribute to give Subscribing Client control over moderation of ‘job-progress’ Event Notifications that MUST be implemented if and only if the ‘job-progress’ event is implemented. There are no default or supported Printer attributes.

3. Removed the idea that a Delivery Method MAY allow the Printer to moderate certain high frequencey events.

4. Clarified that the Printer MUST treat the address part of the “notify-recipient-uri” attribute value as opaque.

5. Added the REQUIRED ‘printer-stopped’ event and the OPTIONAL ‘job-stopped’ event.

6. Deleted the ‘job-purged’ event.

7. Deleted the “notify-persistence” Subscription Template attribute.

8. Clarified the concept of Compound Event Notifications used by both ‘mailto’ and ‘indp’.

9. Clarified that a Printer MUST cancel a Subscription if it gets hard errors when sending that will never change.

10. Clarified Figure 3 – Opaque Use of a Notification Service Transparent to the Client to indicate that the Printer includes the Notification Service.

G.2 Changes to the May 10, 2000 version to create the June 30, 2000 version

The following changes were made to the May 10, 2000 version to create the June 30, 2000 version based on the agreements reached at the May IPP WG meetings and subsequent teleconferences:

1. Editorially reorganized and revised the document so that information is stated only once. Moved supplementary material to appendices.

2. Cleaned up the terminology so that it is used consistently throughout the document; capitalized such terms. Simplified the descriptions of each term.

3. Recast the Subscription attributes to be Subscription Template and Subscription Description attributes following the IPP/1.1 model for Jobs. Therefore, a few attribute names were changed to make them consistent.

4. Reworked the operation descriptions to align with the style in [ipp-mod].

5. Made the validation and processing of Subscription Template attributes be the same for Job Creation Operations, Create-Job-Subscriptions, and Create-Printer-Subscriptions operations (and defined in one place) and as similar to validation of jobs as possible (though there are some differences since one request can generate multiple Subscription objects.

6. Clarified the error handling for all operations.

7. Removed the “notify-text-format” and “notify-additional-formats” Subscription Template attributes and added the “notify-job-id” Subscription Description attribute.

8. The client can supply one or more Subscription Template Attribute Groups in all Subscription Creation requests and the printer returns Subscription Object Attributes groups for each Subscription object created. Consequently, an “s” was added to Create-Job-Subscriptions and Create-Printer-Subscriptions operations.

9. Reorganized the Events, so that some of the Events represent a group of events and the rest are sub-events. This reduces the number of Subscribed Events that a Printer needs to support in one Subscription from 5 to 2. It also means that the event that is delivered is one of the Subscribed events, not necessarily the trigger event, so “notify-trigger-event” was renames to “notify-subscribed-event” in the Event Notification.

10. Added the ‘printer-full’ and ‘printer-not-almost-idle’ Events to go along with the ‘printer-no-longer-full’ and ‘printer-almost-idle’ Events. Renamed the ‘printer-queue-changed’ Event to ‘printer-queue-order-changed’.

11. Clarified what MUST be in a Delivery Method Document.

12. Removed “persistent-jobs-supported” Printer Description attribute, since it has nothing to do with Notifications and is not needed to describe Subscription object persistence.

13. Changed notify-max-printer-subscriptions-supported (integer(0:MAX)) and notify-max-job-subscriptions-supported (integer(0:MAX)) so that MAX means no limit and 0 means no subscriptions are (currently) allowed, so as to give a way to turn off accepting new subscriptions.

G.3 Changes to the March 8, 2000 version to create the May 10, 2000 version

The following changes were made to the March 8, 2000 version to create the May 10, 2000 version based on the agreements reached at the April IPP WG meetings and subsequent teleconferences:

1. Change “notify-format” to “notify-text-format” and made it apply only to the format of the “notify-text” (formerly called “human-readable-report”) and Human Consumable form. A new attribute “notify-additional-formats” specifies the formats for the Machine Consumable contents of Delivery Methods that support multiple formats.

2. Change the “job-notify” collection attribute in Job Creation operations to be multiple “notify-xxx” attributes. This change eliminates the need for collection values. It also means that a Job Creation operation can create only one Subscription Object.

3. Change the Machine Consumable form to be transport independent.

4. Reduce the set of REQUIRED attributes in the Machine Consumable form and add the OPTIONAL “notify-attributes” attribute that allows a client to request additional attributes.

5. Specify the information that SHOULD be in the Human Consumable form

G.4 Changes 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.

G.5 Changes 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-Subscriptions, and Create-Printer-Subscriptions.
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 “notify-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 “notify-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.

G.6 Changes 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.

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-Subscriptions and Create-Printer-Subscriptions 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.

H. Appendix: Full Copyright Statement