Network Working Group
Request for Comments: 3997
Category: Informational

T. Hastings, Ed.
Xerox Corporation
R. K. deBry
Utah Valley State College
H. Lewis
IBM Corporation

March 2005

Internet Printing Protocol (IPP): Requirements for IPP Notifications

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document is one of a set of documents that together describe all aspects of the Internet Printing Protocol (IPP). IPP is an application-level protocol that can be used for distributed printing on the Internet. There are multiple parts to IPP, but the primary architectural components are the Model, the Protocol, and an interface to Directory Services. This document provides a statement of the requirements for notifications as an optional part of an IPP Service.

Table of Contents

1. Introduction. 2
2. Terminology 2
3. Scenarios 6
4. Requirements. 10
5. Security Considerations for IPP Notifications Requirements. . 12
6. Internationalization Considerations 13
7. IANA Considerations 13
8. References. 14
   8.1. Normative References. 14
   8.2. Informative References. 14
9. Appendix A: Description of the Base IPP Documents 15
Authors' Addresses 16
Full Copyright Statement 17

1. Introduction

This document is one of a set of documents that together describe all aspects of the Internet Printing Protocol (IPP). IPP is an application level protocol that can be used for distributed printing on the Internet. There are multiple parts to IPP, but the primary architectural components are the Model, the Protocol, and an interface to Directory Services. This document provides a statement of the requirements for notifications as an optional part of an IPP Service. See section 10 for a description of the base IPP documents.

The scope of this requirements document covers functionality used by the following kinds of IPP Users: End Users, Print Administrators, and Operators. See [RFC3995] for the extensions to the Internet Printing Protocol/1.0 (IPP) [RFC2565], [RFC2566], IPP/1.1 [RFC2911], [RFC2910], and future versions.

2. Terminology

It is necessary to define a set of terms to be able to clearly express the requirements for notification services in an IPP System.

2.1. Job-Submitting End User

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

2.2. Administrator

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

2.3. Operator

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

2.4. Job-Submitting Application

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

2.5. Security Domain

For the purposes of this discussion, the set of network components that can communicate without going through a proxy or firewall. A security domain may be geographically very large; for example, anywhere within example.com.

2.6. IPP Client

The software component that sends IPP requests to an IPP Printer object and accepts IPP responses from an IPP Printer.

2.7. Job Recipient

A human who is the ultimate consumer of the print job. In many cases this will be the same person as the Job-Submitting End User, but this need not always be the case. For example, if I use IPP to print a document on a printer in a business partner's office, I am the Job-Submitting End User, and the person whom I intend the document for in my business partner's office is the Job Recipient. Since one of the goals of IPP is to be able to print near the Job Recipient, we would normally expect that person to be in the same security domain as, and geographically near, the Printer. However, this may not always be the case. For example, I submit a print job across the Internet to a XYZ's print shop. I am both the Submitting End User and the Job Recipient, but I am neither near nor in the same security domain as the Printer.

2.8. Job Recipient Proxy

A person acting on behalf of the Job Recipient. The Job Recipient Proxy physically picks up the printed document from the Printer if the Job Recipient cannot do so. The Proxy is by definition geographically near and in the same security domain as the printer. For example, I submit a print job from home to be printed on a printer at work. I'd like my secretary to pick up the print job and put it on my desk. In this case, I am acting as both a Job-Submitting End User and a Job Recipient. My secretary is acting as a Job Recipient Proxy.

2.9. Notification Subscriber

A client that requests the IPP Printer to send Event Notifications to one or more Notification Recipients. A Notification Subscriber may be a Job-Submitting End User or an End User, an Operator, or an Administrator that is not submitting a job.

2.10. Notification Source

The entity that sends Event Notifications.

2.11. Notification Recipient

The entity that receives IPP Notifications about Job and/or Printer events. A Notification Recipient may be a Job Submitting End User, a Job-Submitting Application, a Job Recipient, a Job Recipient Proxy, an Operator, an Administrator, etc., and his or her representative, log file, usage statistics-gathering application, or other active or passive entities.

2.12. Notification Recipient Agent

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

2.13. Event

An Event is an occurrence (either expected or unexpected) within the printing system of a change of state, condition, or configuration of a Job or Printer object.

2.14. Event Notification

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

2.15. Notification Subscription

A Notification Subscription is a request by a Notification Subscriber to the IPP Printer to send Event Notifications to specified Notification Recipient(s) when an event occurs.

2.16. Notification Attributes

IPP Objects (for example, a print job) from which notification are being sent may have associated attributes. A user may want to have one or more of these returned along with a particular notification. In general, these may include any attribute associated with the object emitting the notification. Examples include the following:

number-of-intervening jobs
job-k-octets
job-k-octets processed
job impressions
job-impressions-interpreted
job-impressions-completed
impressionsCompletedCurrentCopy (job MIB)
sheetCompletedCopyNumber (job MIB)
sheetsCompletedDocumentNumber (job MIB)
Copies-requested
Copy-type
Output-destination
Job-state-reasons
Job ID
Printer URI
Subscription ID (for job independent subscription)

2.17. Notification Delivery Method (or Delivery Method for Short)

Event Notifications are delivered by using a Delivery Method. An example of a Delivery Method is email.

2.18. Immediate Notification

Notifications sent to the Notification Recipient or the Notification Recipient's agent in such a way that the notification arrives immediately, within the limits of common addressing, routing, network congestion, and quality of service.

2.19. Store-and-Forward Notification

Notifications that are not necessarily delivered to Notification Recipients immediately but are queued for delivery by an intermediate network application, for later retrieval. Email is an example of a store-and-forward notification delivery method.

2.20. Reliable Delivery of Notifications

Notifications that are delivered by a reliable delivery of packets or character stream, with acknowledgement and retry, so that delivery of the notification is guaranteed within determinate time limits. For example, if the Notification Recipient has logged off and gone home for the day, an immediate notification cannot be guaranteed, even when sent over a reliable transport, because there is nothing there to catch it. Guaranteed delivery requires both store-and-forward notification and a reliable transport.

2.21. Notification over Unreliable Transport

Notifications are delivered via the fundamental transport address and routing framework, but no acknowledgement or retry is required. Process-to-process communications, if involved, are unconstrained.

2.22. Human-Consumable Notification

Notifications intended to be consumed by human end users only. Email would be an example of a Human-Consumable Notification, though it could also contain Machine-Consumable Notification.

2.23. Machine-Consumable Notification

Notifications that are intended for consumption by a program only, such as an IPP Client. Machine-Consumable Notifications may not contain human-readable information. Do we need both human and machine? Machine readable is intended for application-to-application only. The Notification Recipient could process the machine-readable Event Notification into human-readable format.

2.24. Mixed Notification

A mixed notification contains both Human-Consumable and Machine-Consumable information.

3. Scenarios

1. Sitting in my office, I submit a print job to the printer down the hall. I am in the same security domain as the printer and, of course, geographically near. I want to know immediately when my print job will be completed (or if there is a problem) because the document I am working on is urgent. I submit the print job with the following attributes:

- Notification Recipient: Me
- Notification Events: All
- Notification Attributes: Job-state-reason
- Notification Type: Immediate

2. Working from home, I submit a print job to the same printer as in the previous example. However, I am not at work, I cannot physically get the print file or do anything with it. It can wait until I get to work this afternoon. However, I'd like my secretary to pick up the output and put it on my desk so that it doesn't get lost or misfiled. I'd also like a store-and-forward notification sent to my email so that when I get to work I can tell whether there was a problem with the print job. I submit a print job with the following attributes:

   - Notification Recipient: My secretary
   - Notification Events: Print complete
   - Notification Type: Immediate

   - Notification Recipient: Me
   - Notification Events: Print complete
   - Notification Attributes: Impressions completed
   - Notification Type: Store and forward

3. Sitting in my office, I submit a print job to a client at an engineering firm my company works with on a daily basis. The engineering firm is in Belgium. I would like my client to know when the print job is complete so that she can pick it up from the printer in her building. It is important that she review it right away and send her comments back to me. I submit the print job with the following attributes:

   - Notification Recipient: Client at engineering firm
   - Notification Events: Print complete
   - Notification Type: Immediate
   - Notification Language: French

4. From a hotel room, I send a print job to a Kinko's store in the town I am working in, in order to get a printed report for the meeting I am attending in the morning. As I'm going out to dinner after I get this job submitted, an immediate notification won't do me much good. However, I'd like to check in the morning before I drive to the Kinko's store to see whether the file has been printed. An email notification is sufficient for this purpose. I submit the print job with the following attributes:

- Notification Recipient: Me
- Notification Events: Print complete
- Notification Type: Store and forward

5. I am printing a large, complex print file. I want to have some immediate feedback on the progress of the print job as it prints. I submit the print job with the following attributes:

   - Notification Recipient: Me
   - Notification Type: Immediate
   - Notification Events: All state transitions
   - Notification Attributes: Impression completed

6. I am an operator and one of my duties is to keep the printer running. I subscribe independently from a job submission so that my subscription outlasts any particular job. I subscribe with the following attributes:

   - Notification Recipient: Me
   - Notification Type: Immediate
   - Notification Events: All Printer state transitions
   - Notification Attributes: Printer state, printer state reasons, device powering up, device powering down

7. I am a usage statistics gathering application. I subscribe independently from a job submission so that my subscription outlasts any particular job. My subscription may persist across power cycles. I subscribe with the following attributes:

   - Notification Recipient: Me
   - Notification Type: Immediate
   - Notification Events: Job completion
   - Notification Attributes: Impression completed, sheets completed, time submitted, time started, time completed, job owner, job size in octets, etc.

8. I am a client application program that displays a list of jobs currently queued for printing on a printer. I display the "job-name", "job-state", "job-state-reasons", "page-count", and "intervening-jobs", either for the user's jobs or for all jobs. The window displaying the job list remains open for an independent amount of time, and it is desired that it represent the current state of the queue. It is desired that the application only perform a slow poll in order to recover from any missed notifications. So the event delivery mechanism provides the means to update the screen on all needed changes, including querying for some attributes that may not be delivered in the Notification.

9. I am a client application program that displays a list of printers. For each Printer, I display the current state and configuration. The window displaying the printer list remains open for an independent amount of time, and it is desired that it represent the current state of each printer. It is desired that the application only need to perform a slow poll in order to recover from any missed notifications. So the event delivery mechanism provides the means to update the screen on all needed changes, including querying for some attributes that may not be delivered in the Notification.

10. I am an IPP Server that controls one or more devices and that implements an IPP Printer object to represent each device. I want to support IPP Notification for each of the IPP Printer objects that I implement. Many of these devices do not support notification (or IPP). So I need to support the IPP Notification semantics specified for each IPP Printer object myself on behalf of each of the devices that each of the IPP Printer objects represents. When I accept an IPP job creation requests, I convert it to what the device will accept. In some cases, I must poll the devices in order to be informed of their job and device state and state changes to be able to send IPP Notifications to subscribed Notification Recipients.

11. I am an IPP Server that controls one or more devices and that implements an IPP Printer object to represent each device. I want to support IPP Notification for each of the IPP Printer objects that I implement. These devices all support IPP, including IPP Notification. I would like the design choice for supporting IPP Notification for these objects either (1) by forwarding the notification to the IPP Printers that I, alone, control and have them send the notifications to the intended Notification Recipients without my involvement, or (2) by replacing the notification submitted with the Job to indicate me as the Notification Recipient; in turn I will forward Notifications to the Notification Recipients requested by my clients. Most of the rest of the contents of the IPP Job I send to the IPP Printers I control will be the same as those that I receive from my IPP clients.

12. I am an IPP Server that controls one or more devices and that implements an IPP Printer object to represent each device. I want to support IPP Notification for each of the IPP Printer objects that I implement. These devices all support IPP, including IPP Notification. Because these IPP Printers MAY also be controlled by other servers (using IPP or other protocols), I only want job events for the jobs that I send, but I do want Printer events all the time, so that I can show proper Printer

state to my clients. So I subscribe to these IPP Printers for Printer events with a long-standing subscription, with myself as the Notification Recipient. When I get a Job Creation request, I decide to which IPP Printer to send the job. When I do so, I also add a job subscription for Job events, with me as the Notification Recipient to the job's job subscriptions supplied by my clients (this usage is called "piggybacking"). These IPP Printers automatically remove their job subscriptions when the job finishes, as for all job subscriptions, so that I no longer get Job events when my jobs are completed.

4. Requirements

The following requirements are intended to be met by the IPP Notification specification (not the implementation). The following are true for the resulting IPP Notification Specification document:

1. It must indicate which of these requirements are REQUIRED and which are OPTIONAL for a conforming implementation to support. See [RFC2911], section 12.1, for the definition of these important conformance terms.

2. It must be designed so that an IPP Printer can transparently support the IPP Notification semantics by using third-party notification services that exist today or that may be standardized in the future.

3. It must define a means for a Job-Submitting End User to specify zero or more Notification Recipients when submitting a print job. A Submitter will not be able to prevent out-of-band subscriptions from authorized persons, such as Operators.

4. It must define a means, when specifying a Notification Recipient, for a Notification Subscriber to specify one or more notification events for that Notification Recipient, subject to administrative and security policy restrictions. Any of the following constitute Job or Printer Events for which a Job Submitting End User can specify that notifications be sent:

   - Any standard Printer MIB alert
   - Job Received (transition from Unknown to Pending)
   - Job Started (transition from Pending to Processing)
   - Page Complete (page is stacked)
   - Collated Copy Complete (last sheet of collated copy is stacked)

- Job Complete (transition from Processing or Processing-stopped to Completed)
- Job Aborted (transition from Pending, Pending-held,
- Processing, or Processing-stopped to Aborted)
- Job Canceled (transition from Pending, Pending-held,
- Processing, or Processing-held to Canceled)
- Other job state changes, such as paused, purged
- Device problems for which the job is destined
- Job (interpreter) issues

5. It must define how an End User or Operator subscribes for

   - any set of Job Events for a specific job, or
   - any set of Printer Events while a specific job is not complete.

6. It must define how an End User or Operator subscribes for the following without having to submit a Job:

   - Any set of Printer Events for a defined period.
   - Any set of Job Events for all jobs, with no control over which jobs.

7. It must define how the Notification Subscriber is able to specify either immediate or store-and-forward notification independently for each Notification Recipient. The means may be explicit, or implied by the method of delivery chosen by the Job Submitting End User.

8. It must define common delivery methods: e.g., email.

9. It must define how an IPP Printer validates its ability to deliver an Event by using the specified delivery scheme. If it does not support the specified scheme, or if the specified scheme is invalid for some reason, then the IPP Printer accepts and performs the request anyway and indicates the unsupported attribute values. There is no requirement for the IPP Printer receiving the print request to validate the identity of a Notification Recipient, or the ability of the system to deliver an event to that recipient as requested (for example, if the Notification Recipient is not at work today).

10. It must define a class of IPP event notification delivery methods that can flow through corporate firewalls. However, an IPP printer need not test to guarantee delivery of the notification through a firewall before accepting a print job.

11. It may define a means to deliver a notification to the submitting client when the delivery of an event notification to a specified Notification Recipient fails. A fallback means of subscribers to determine whether notifications have failed (i.e., polling) may be provided.

12. It must define a mechanism for localizing Human-Consumable Notifications by the Notification Source.

13. It may define a way to specify whether event delivery requires acknowledgement back to the Notification Source.

14. There must be a mechanism defined so that job-independent subscriptions do not become stale and do not require human intervention to be removed. However, a subscription must not be deemed stale only if it is unable to deliver an Event Notification, as temporary Notification delivery problems must be tolerated.

15. A mechanism must be defined so that an Event Subscriber is able to add an Event Subscription to a Job after the Job has been submitted.

16. A mechanism must be defined so that a client is able to cancel an Event Subscription on a job or printer after the job has been submitted.

17. A mechanism must be defined so that a client can obtain the set of current Subscriptions.

5. Security Considerations for IPP Notifications Requirements

By far the biggest security concern is the abuse of notification: sending unwanted notifications sent 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). Scenarios exist in which third-party notification is required (see scenarios 2 and 3). The fully secure solution would require active agreement of all recipients before anything is sent out. However, requirement 9 ("There is no requirement for an IPP Printer receiving the print request to validate the identity of an event recipient") argues against this. Certain systems may decide to disallow third-party notifications (a traditional fax model).

The same security issues are present when Clients submit notification requests to the IPP Printer as when they submit 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. Creating a notification subscription is associated with a user. Only the creator or an operator can cancel the subscription. The system may limit the listing of items to items owned by the user. Some subscriptions (e.g., those that have a lifetime longer than a job) can be done only by privileged users (operators and/or administrators), if that is the authorization policy.

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

6. Internationalization Considerations

The Human-Consumable Notification must be localized to the natural language and charset that Notification Subscriber specifies within the choice of natural languages and charsets that the IPP Printer supports.

The Machine-Consumable Notification data uses the "application/ipp" MIME media type. It contains attributes whose text values are required to be in the natural language and charset that the Notification Subscriber specifies within the choice of natural languages and charsets that the IPP Printer supports. See [RFC2566].

7. IANA Considerations

Some notification delivery methods have been registered with IANA for use in URLs. These will be defined in other documents.

8. References

8.1. Normative References

[RFC2910] Herriot, R., Butler, S., Moore, P., Turner, R., and J. Wenn, "Internet Printing Protocol/1.1: Encoding and Transport", RFC 2910, September 2000.

[RFC2911] Hastings, T., Herriot, R., deBry, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.1: Model and Semantics", RFC 2911, September 2000.

[RFC3995] Herriot, R. and T. Hastings, "Internet Printing Protocol (IPP): Event Notifications and Subscriptions", RFC 3995, March 2005.

8.2. Informative References

[RFC2565] Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.0: Encoding and Transport", RFC 2565, April 1999.

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

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

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

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

[RFC2639] Hastings, T., Manros, C., Zehler, P., Kugler, C., and H. Holst, "Internet Printing Protocol/1.1: Implementor's Guide", RFC 3196, November 2001.

9. Appendix A: Description of the Base IPP Documents

The base set of IPP documents includes the following:

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 [RFC2911]
Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
Internet Printing Protocol/1.1: Implementer's Guide [RFC3196]
Mapping between LPD and IPP Protocols [RFC2569]

"Design Goals for an Internet Printing Protocol" takes a broad look at distributed printing functionality, and it enumerates real-life scenarios that help 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 [RFC2566], [RFC2565]. A few OPTIONAL operator operations have been added to IPP/1.1 [RFC2911], [RFC2910].

"Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" describes IPP from a high-level view, defines a roadmap for the various documents that form the suite of IPP specification documents, and gives background and rationale for the IETF IPP working group's major decisions.

"Internet Printing Protocol/1.1: Model and Semantics" describes a simplified model with abstract objects, their attributes, and their operations. The model introduces a Printer and a Job. The Job supports multiple documents per Job. The model document also addresses security, internationalization, and directory issues.

"Internet Printing Protocol/1.1: Encoding and Transport" is a formal mapping of the abstract operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It also 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 the "ipp" scheme for identifying IPP printers and jobs.

"Internet Printing Protocol/1.1: Implementer's Guide" gives insight and advice to implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some of the considerations that may assist them in the design of their client and/or IPP object implementations. For example, a typical order of processing requests is given, including error checking. Motivation for some of the specification decisions is also included.

"Mapping between LPD and IPP Protocols" gives some advice to implementers of gateways between IPP and LPD (Line Printer Daemon) implementations.

Authors' Addresses

Tom Hastings (editor)
Xerox Corporation
701 S Aviation Blvd, ESAE 242
El Segundo, CA   90245

Phone: 310-333-6413
Fax: 310-333-6342
EMail: hastings@cp10.es.xerox.com

Roger deBry
Utah Valley State College

Phone: (801) 863-8848
EMail: debryro@uvsc.edu

Harry Lewis
IBM Corporation
6300 Diagonal Hwy
Boulder, CO 80301

Phone: (303) 924-5337
EMail: harryl@us.ibm.com

Full Copyright Statement

Intellectual Property

Acknowledgement

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