PWG WORKING DRAFT
<ipp-ntfy-http-protocol>

Hugo Parra
Novell, Inc.
September 20, 1999

Internet Printing Protocol: **HTTP-Based IPP Notification Protocol**

Status of this Memo

This document proposes an HTTP-based mechanism for delivering IPP notification. The intent is for this document to become a Printer Working Group (PWG) DRAFT.

Abstract

The IPP notification specification [ipp-ntfy] requires the availability of one or more delivery methods for dispatching notification reports to interested parties. This document describes the semantics and syntax of a protocol that a delivery method may use to deliver IPP notifications using HTTP for a transport.

Table of Contents

1. Introduction

IPP printers that support IPP notification either a) accept, store, and use notification subscriptions to generate notification reports and implement one or more delivery methods for notifying interested parties, or b) use the services of a Notification Delivery Service to perform some of these tasks. An IPP printer or a Notification Delivery Service may use the protocol specified in this document to send IPP notification reports over HTTP to a notification recipient.

2. Model and Operation

The HTTP-Based IPP Notification Protocol, hereafter referred to as HTTP notification protocol, is a client/server protocol. The ‘client’ in this relationship invokes operations supported by the HTTP notification protocol to communicate IPP Notification contents to the ‘server’. The ‘server’ only conveys information to the ‘client’ in the form of responses to the operations initiated by the ‘client’. Typically, it will be IPP printers and Notification Delivery Services who will implement HTTP notification clients. Conversely, it will be entities with IPP clients (potential IPP notification subscribers), who will normally implement HTTP notification servers. To avoid confusion with the overuse of the ‘client’ and ‘server’ terms, this document refers to an implementation of the HTTP notification client as an HTTP notification delivery method and an implementation of the HTTP notification server as an HTTP notification listener.

HTTP notification requests will be issued as HTTP POST operations and their corresponding HTTP notification responses will be returned in the responses those HTTP POST operations. Hence, HTTP notification delivery methods will need to include an HTTP client stack while HTTP notification listeners will need to implement an HTTP server stack (see section 4 for more details).

2.1 HTTP Notification Operations

The job of an HTTP notification delivery method is to use the contents of an IPP Notification as defined in [ipp-ntfy] to compose and invoke the appropriate HTTP notification operation and submit it to the specified HTTP notification listener.

The HTTP notification protocol makes extensive use of the operations model defined by IPP [rfc2566]. This includes, the use of a URI as the identifier for the target of each operation, the inclusion of a version number, operation-id, and request-id in each request, and the definition of attribute groups. The HTTP notification protocol uses the Operation Attributes group, but currently has no need for the Unsupported Attributes, Printer Object Attributes, and Job-Object Attributes groups. However, it defines a new attribute group, the Notification Report Attributes group.

In its 1.0 version, the HTTP notification protocol is composed of a single operation, but may be extended in the future as needed (e.g., to find out specific capabilities of an HTTP notification listener). The operation currently defined is Report-Ipp-Notifications.

<ISSUE 1: *Can we get away without describing the HTTP notification listener as a formal object?>*

2.1.1 Report-Ipp-Notifications

This REQUIRED operation allows an HTTP notification delivery method to send one or more notification reports to an HTTP notification listener. The operation has been tailored to enable simple mapping of the data available in an IPP Notification onto a Report-Ipp-Notifications Request.

Both ‘machine-consumable’ and ‘human-consumable’ notifications may be send to an HTTP notification listener through this operation.

2.1.1.1 Report-Ipp-Notifications Request

The following groups of attributes are part of the Report-Ipp-Notifications Request:

Group 1: Operation Attributes

Natural Language and Character Set:

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

Target:

The URI of the HTTP notification listener.

Group 2 to N: Notification Report Attributes

<ISSUE 2*: I see two options here, I’ll include both for discussion - HParra>*

<Option 1>

“machine-consumable-report” (collection of IPP Notifications)

The HTTP delivery method OPTIONALLY supplies this attribute. The contents of an IPP Notification that lend themselves to be machine-consumable are defined as a collection. This attribute is populated if the delivery method is capable of generating machine-consumable Notifications.

“human-readable-report” (text)

The HTTP delivery method OPTIONALLY supplies this attribute. A text string generated by the IPP printer or Notification Delivery Service from the contents of the IPP Notification suitable for humans to consume.

<End of Option 1>

<Option 2>

“Version-number” (integer (0:32767))

Blah-blah-blah

“status-code” (integer (0:32767))

Blah-blah-blah

“request-id” (integer (0:MAX))

Blah-blah-blah

“attributes-charset” (charset)

Blah-blah-blah

“attributes-natural-language” (naturalLanguage)

Blah-blah-blah

“printer-uri” (uri)

Blah-blah-blah

“printer-name” (name(127))

Blah-blah-blah

“job-id” (integer(1:MAX))

Blah-blah-blah

“job-name” (name(MAX))

Blah-blah-blah

“trigger-event” (type2 keyword)

Blah-blah-blah

“trigger-time” (integer(MIN:MAX))

Blah-blah-blah

“trigger-date-time” (dateTime)

Blah-blah-blah

“subscription-id” (integer(1:MAX))

Blah-blah-blah

“subscriber-user-name” (name(MAX))

Blah-blah-blah

“subscriber-user-data” (octetString(63))

Blah-blah-blah

“job-state” (type1 enum)

Blah-blah-blah

“job-state-reasons” (1setOf type2 keyword)

Blah-blah-blah

“job-k-octets-processed” (integer(0:MAX))

Blah-blah-blah

“job-impressions-completed” (integer(0:MAX))

Blah-blah-blah

"job-media-sheets-completed" (integer(0:MAX))

Blah-blah-blah

"job-collation-type" (type2 enum)

Blah-blah-blah

"sheet-completed-copy-number" (integer(-2:MAX))

Blah-blah-blah

"sheet-completed-document-number" (integer(-2:MAX))

Blah-blah-blah

"impressions-interpreted" (integer(-2:MAX))

Blah-blah-blah

"impressions-completed-current-copy" (integer(-2:MAX))

Blah-blah-blah

"printer-state" (type1 enum)

Blah-blah-blah

"printer-state-reasons" (1setOf type2 keyword)

Blah-blah-blah

"printer-is-accepting-jobs" (boolean)

Blah-blah-blah

"human-readable-report" (text)

The HTTP delivery method OPTIONALLY supplies this attribute. A text string generated by the IPP printer or Notification Delivery Service from the contents of the IPP Notification suitable for humans to consume.

<End of Option 2>

<ISSUE 3: *Should the Operation Attributes "attributes-charset" and "attributes-natural-language" be removed as each Notification Report specifies its own?*>

2.1.1.2 Report-Ipp-Notifications Response

The HTTP notification listener returns a status code for the entire operation and one for each Notification Report in the request if the operation's status code is other than "success-ok". If the HTTP notification listener receives a Notification report that it can't pair up with a subscription it know about, it can return an error status-code to indicate that evens associated with that subscription should no longer be sent to him

Group 1: Operation Attributes

Natural Language and Character Set:

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

Group 2 to N: Notification Report Attributes

“notification-report-status-code” (type2 enum)

Indicates whether the HTTP notification listener was able to consume the nth Notification Report.

2.2 HTTP Notification Protocol URI Scheme

<ISSUE 4: *Should the URI scheme for this protocol be “http://”, “ipp://”, or something else like “ipp-ntfy://”. If we intent this proposal to go to the IESG, something along the lines of the third option might be our only alternative*>

3. Encoding of the Operation Layer

The HTTP notification protocol uses the same operation layer encoding model and syntax as IPP [ipp-pro] with two extensions:

a) A new attribute tag is defined:

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

b) The following status codes are defined

0xYYYY - unknown-notification-recipient.
0xZZZZ - unable-to-delivery-notification-report

The encoding for the Report-IPP-Notification Request consists of:

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

Where:

version-number is made up of a major-version-number of %d1 and a minor-version-number of %d0 indicating the 1.0 version of the HTTP notification protocol.

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

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

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

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

The encoding for the Report-IPP-Notification Response consists of:

```
   -----------------------------------
   |         version-number          |  2 byte
   -----------------------------------
   |           status-code           |  2 bytes
   -----------------------------------
   |           request-id            |  4 bytes
   -----------------------------------             \
   |    operation-attributes-tag     |  1 byte     |
   -----------------------------------             |
   |   natural-language-attribute    |  u bytes    |  Not needed in 1.0
   -----------------------------------              > <ISSUE 5: Do we
   |        charset-attribute        |  v bytes    |  want to keep it?>
   -----------------------------------             |
   |        target-attribute         |  w bytes    |
   ----------------------------------------------  /
   |     notification-report-tag     |  1 byte  |
   -----------------------------------          | - 1 or more
   |        ntfy-status-code         |  2 bytes |
   ----------------------------------------------
   |      end-of-attributes-tag      |  1 byte
   -----------------------------------
```

4. Encoding of Transport Layer

HTTP/1.1 [rfc2068] is the transport layer for this protocol.
The operation layer has been designed with the assumption that the transport layer contains the following information:

- the URI of the target job or printer operation.

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

It is REQUIRED that an HTTP notification listener implementation support HTTP over the IANA assigned Well Known Port XXX (the HTTP notification protocol default port), though a listener implementation may support HTTP over some other port as well.

<ISSUE 6: *Does this protocol need a Well Known Port or, since the subscriber provides its delivery URL with each subscription, can we use dynamically allocated ports?*>

Each HTTP operation MUST use the POST method where the request-URI is the object target of the operation, and where the "Content-Type" of the message-body in each request and response MUST be "application/???". The message-body MUST contain the operation layer and MUST have the syntax described in section 3, "Encoding of Operation Layer”. An HTTP notification delivery method implementation MUST adhere to the rules for a client described for HTTP1.1 [rfc2068]. An HTTP notification listener implementation MUST adhere the rules for an origin server described for HTTP1.1 [rfc2068].

An HTTP notification listener sends a response for each request that it receives. If a notification listener detects an error, it MAY send a response before it has read the entire request. If the HTTP layer of the listener completes processing the HTTP headers successfully, it MAY send an intermediate response, such as "100 Continue", with no IPP data before sending the IPP response. An HTTP notification delivery method MUST expect such a variety of responses from an notification listener. For further information on HTTP/1.1, consult the HTTP documents [rfc2068].

An HTTP server MUST support chunking for HTTP notification requests, and an HTTP notification delivery method MUST support chunking for HTTP notification responses according to HTTP/1.1[rfc2068]. Note: this rule causes a conflict with non-compliant implementations of HTTP/1.1 that don't support chunking for POST methods, and this rule may cause a conflict with non-compliant implementations of HTTP/1.1 that don't support chunking for CGI scripts

5. Security Considerations

Blah, blah, blah

6. References

[ipp-ntfy]
Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., “Internet Printing Protocol/1.0 & 1.1: IPP Event Notification Specification”, <draft-ietf-ipp-not-spec-01.doc>, September 9, 1999.

[ipp-pro]
Herriot, R., Butler, S., Moore, P., Tuner, R., “Internet Printing Protocol/1.1: Encoding and Transport”, draft-ietf-ipp-protocol-v11-01.txt, May, 1999.

[rfc2068]
R Fielding, et al, “Hypertext Transfer Protocol – HTTP/1.1” RFC 2068, January 1997.

[rfc2566]
deBry, R., Hastings, T., Herriot, R., Isaacson, S., Powell, P., “Internet Printing Protocol/1.0: Model and Semantics”, RFC 2566, April 1999.