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

Hugo Parra
Novell, Inc.
October 19, 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) support a subset of these tasks and farm out the remaining tasks to a Notification Delivery Service. The protocol specified in this document may be used in a variety of notification scenarios. Its primary intended use is for IPP printers to send notifications to notification recipients over HTTP. However, it may also be used by IPP printers to send notification to Notification Services and by Notification Delivery Services to send notifications to notification recipients.

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 HTTP relationship is the notification source described in [ipp-ntfy] while the “server” is the notification recipient. The notification source invokes operations supported by the HTTP notification protocol to communicate IPP Notification contents to the notification recipient. The notification recipient only conveys information to the notification source in the form of responses to the operations initiated by the notification source.

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

2.1 HTTP Notification Operations

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

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 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 Send-Notifications.

2.1.1 Report-Ipp-Notifications

This REQUIRED operation allows a notification source to send one or more notifications to notification recipient using HTTP. The operation has been tailored to accommodate the current definition of IPP Notification.

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

2.1.1.1 Send-Notifications Request

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

Group 1: Operation Attributes

Natural Language and Character Set:

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

Target:

The URI of the HTTP notification recipient.

Group 2 to N: Notification Attributes

“human-readable-report” (text)

The HTTP notification source 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 human consumption.

“version-number” (integer (0:32767))
“status-code” (integer (0:32767))
“request-id” (integer (0:MAX))
“attributes-charset” (charset)
“attributes-natural-language” (naturalLanguage)
“printer-uri” (uri)
“printer-name” (name(127))
“job-id” (integer(1:MAX))
“job-name” (name(MAX))
“trigger-event” (type2 keyword)
“trigger-time” (integer(MIN:MAX))
“trigger-date-time” (dateTime)
“subscription-id” (integer(1:MAX))
“subscriber-user-name” (name(MAX))
“subscriber-user-data” (octetString(63))
“job-state” (type1 enum)
“job-state-reasons” (1setOf type2 keyword)
“job-k-octets-processed” (integer(0:MAX))
“job-impressions-completed” (integer(0:MAX))
“job-media-sheets-completed” (integer(0:MAX))
“job-collation-type” (type2 enum)
“sheet-completed-copy-number” (integer(-2:MAX))
“sheet-completed-document-number” (integer(-2:MAX))
“impressions-interpreted” (integer(-2:MAX))
“impressions-completed-current-copy” (integer(-2:MAX))
“printer-state” (type1 enum)
“printer-state-reasons” (1setOf type2 keyword)
“printer-is-accepting-jobs” (boolean)

These attributes communicate the same information as the notification attributes by the same name described in sections 7.4, 7.5, and 7.6 of [ipp-ntfy]. The rules that govern when each individual attribute MUST or MAY be included in this operation precisely mirror those specified in [ipp-ntfy].

2.1.1.2 Send-Notifications Response

The HTTP notification recipient 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 knows about, it can return an error status-code to indicate that events associated with that subscription should no longer be sent to it.

Group 1: Operation Attributes

Natural Language and Character Set:

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

Group 2 to N: Notification Attributes

"notification-report-status-code" (type2 enum)

Indicates whether the HTTP notification listener was able to consume the n^{th} 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:

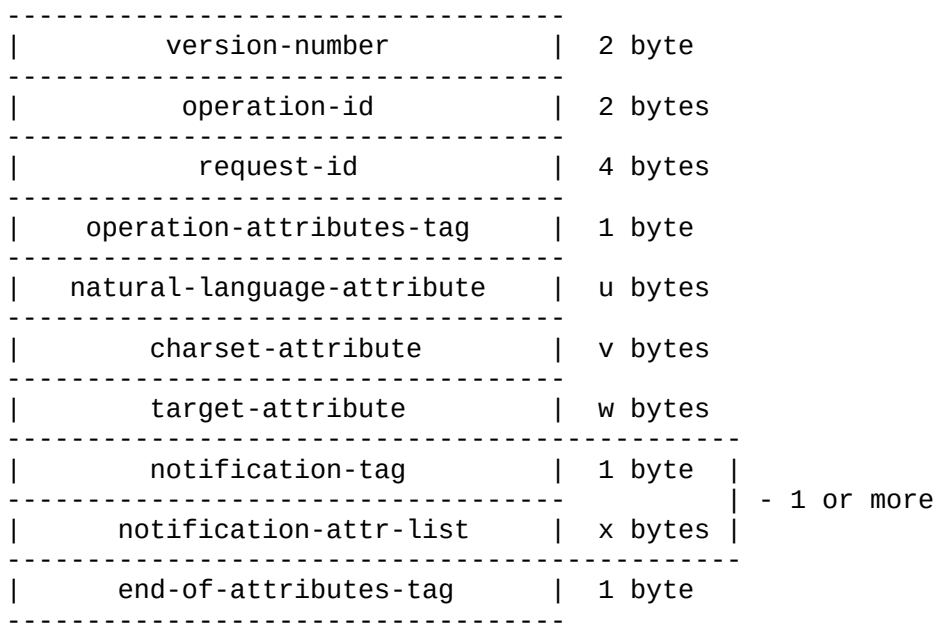
a) A new attribute tag is defined:

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:



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.

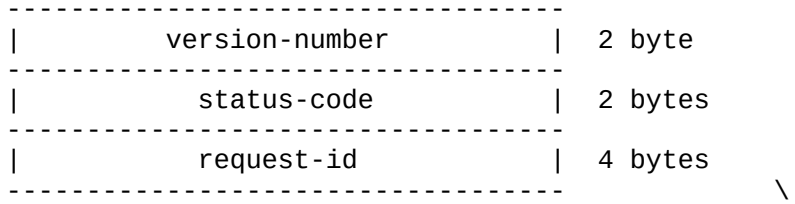
operation-id, in the 1.0 version of the protocol, can only be 0x00003, Report-IPP-Notification.

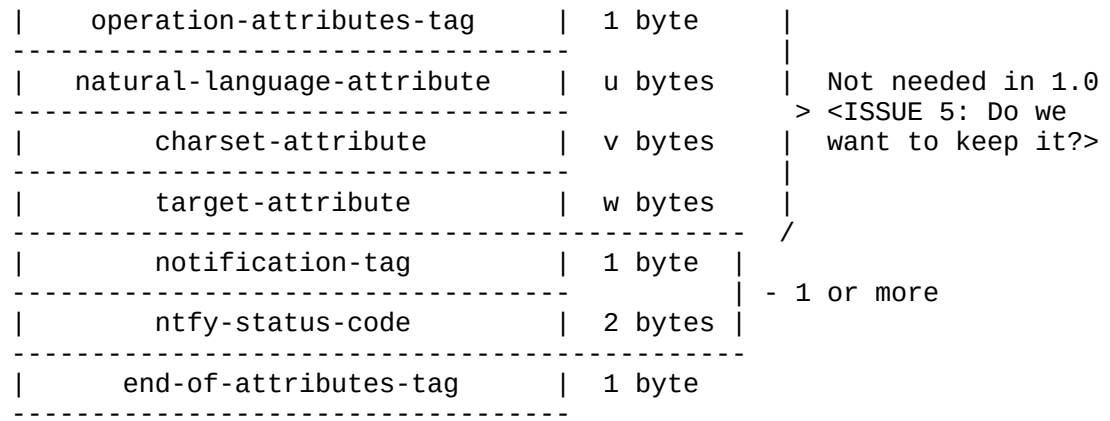
request-id is any 4 byte number provided by the notification source and must be matched by the notification recipient in the corresponding response to a request. It assists the notification source in associating operation responses with their corresponding requests. Note that this request id is independent 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-attr-list contains a list of the attributes that make up a single notification (see section 2 above) encoded using the syntax specified in [ipp-pro].

The encoding for the Send-Notification Response consists of:





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 recipient implementation support HTTP over the IANA assigned Well Known Port XXX (the HTTP notification protocol default port), though a notification recipient implementation may support HTTP over some other port as well.

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/ipp-ntfy". The message-body MUST contain the operation layer and MUST have the syntax described in section 3, "Encoding of Operation Layer". An HTTP notification source implementation MUST adhere to the rules for a client described for HTTP1.1 [rfc2068]. An HTTP notification recipient implementation MUST adhere the rules for an origin server described for HTTP1.1 [rfc2068].

An HTTP notification source sends a response for each request that it receives. If a notification recipient detects an error, it MAY send a response before it has read the entire request. If the HTTP layer of the notification recipient completes processing the HTTP headers successfully, it MAY send an intermediate response, such as "100 Continue", with no notification data before sending the notification response. HTTP notification sources MUST expect such a variety of responses from notification recipients. 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 source 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

The IPP Model and Semantics document [ipp-mod] discusses high level security requirements (Client Authentication, Server Authentication and Operation Privacy). Client Authentication is the mechanism by which the client proves its identity to the server in a secure manner. Server Authentication is the mechanism by which the server proves its identity to the client in a secure manner. Operation Privacy is defined as a mechanism for protecting operations from eavesdropping.

5.1 Security Conformance

Notification sources MAY support:
Digest Authentication [rfc2069].

MD5 and MD5-sess MUST be implemented and supported.

The Message Integrity feature NEED NOT be used.

Notification recipienst MAY support:
Digest Authentication [rfc2069].

MD5 and MD5-sess MUST be implemented and supported.

The Message Integrity feature NEED NOT be used.

Notification recipients MAY support TLS for client authentication, server authentication and operation privacy. If a notification recipient supports TLS, it MUST support the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite as mandated by RFC 2246 [rfc2246]. All other cipher suites are OPTIONAL. Notification recipients MAY support Basic Authentication (described in HTTP/1.1 [rfc 2068]) for client authentication if the channel is secure. TLS with the above mandated cipher suite can provide such a secure channel.

6. References

[ipp-mod]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and Semantics", <draft-ietf-ipp-model-v11-02.txt>, May, 1999.

[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.