Subj: IPP Bake Off 3 Issues
From: Peter Zehler
File: Issues-raised-at-Bake-Off3-001201.doc
Version: 1.0
Date: 12/1/2000

This version incorporates the discussion on the mailing list resolving the IPP/1.1 issues raised at Bake Off 3.

Please feel free to add additional alternatives or disagree with our suggested clarifications or additions via e-mail so that the group may have the widest possible set of alternatives from which to choose.

Status of Issues and Summary

This section lists the status of each issue and a brief summary. The next section is the detailed description of the issue and the resolution or alternatives, if the issue is still OPEN. Please review this status and the detailed issues to see if you agree or disagree with the status so far. Silence will be interpreted as agreement.

Status codes:

AGREED - agreement on the mailing list or telecons on the suggested clarification, suggested change, or resolution. Subsequence silence on the DL will be interpreted as agreement. If you disagree, please indicate this to the ipp@pwg.org DL with the subject line containing: "IPP Bake-Off 3 Issue #" where '#' is the Issue number.

OPEN - still being discussed at future telecons and on the DL.

OPEN issues remaining: 2 and 4.

Issue 3.1: AGREED

IPP Client failed when an unexpected HTTP "100 continue" was received. Some printers sent a "100 continue" even before the Client sent a request.

Proposed Resolution:

An IPP Client must accept and handle an HTTP "100 continue" whenever it is encountered.

Action:

The following caveat will be added to the IIG.
"IPP Clients must be prepared at any time to receive an interim response with a status code of '100 Continue' This includes receiving this response prior to sending an IPP request."

Issue 3.2: OPEN

Some IPP Clients issues a zero length HTTP Post. The Client assumed that this would force a challenge if security is enabled on the Printer. The Client would have a problem if a subsequent print operation were challenged.

Proposed Resolutions:

There are two competing resolutions.

Resolution 1 is that a challenge should be issued whenever an HTTP operation is received on a particular URL. (assuming the URL is part of an authentication space) The client must accept and respond to a challenge the first time a URL is accessed.

Resolution 2 allows the vendor to determine when a challenge is issued. The vendor is free to use the contents of the HTTP request to determine if the operation mandates a challenge. The client must accept and respond to a challenge at any time.

The Client should use the IPP operation "validate-job" to check if a job will be accepted. This operation will cause the Printer to issue a challenge and check the print request before sending the data. The IPP Client should also be able to handle a challenge when issuing an IPP operation since there is no guarantee the connection has not been torn down.
Furthermore, a Printer should accept an empty HTTP post and issue a challenge based on the URL of the post.

Resolution 1:

From Bob Herriot:
I raised the issue about whether a Printer should perform the authentication challenge based solely on the URL or whether it could react differently to an empty request than to a Validate-Job request.

I asked an HTTP expert and received the following information.

1) An HTTP server can have any policy.
This means that resolution 2 is allowable.
2) It is best for a client if it can associate the URL tree with the authentication space.
This means that our decision could be better. That is, we should require an IPP Printer to decide whether to issue an authentication challenge by examining the URL and nothing else, e.g. a Printer receiving a request for a particular URL, gives the same challenge to an empty request as to a Validate-Job request.
This solution allows a client to use Validate-Job to request a challenge as we decided to allow. It also allows a client to use the empty request.
The important difference between our decision and what I am proposing is that the Printer must perform an authentication challenge consistently for a URL regardless of the contents of the message body. This rule make IPP behavior consistent with good HTTP policy.

Resolution 2:

From Peter Zehler:
Allowing IPP Printers to use the contents of an IPP resuest to determine if a challenge should be issued allows for increased usability. The client does not have to keep track of multiple instances of the same

printer and select the appropriate one based on the operation to be performed. The printer is free to determine when authentication is required. This allows the client to use a single URL and authenticate himself when the printer places restrictions on operations or features.
This resolution does not prohibit challenges based statically on a URL. Resolution 2 does require a client to be ready at any time to receive a challenge. This should be done anyway since the client application may be unaware that an HTTP connection has dropped after authenticating the connection, resulting in a new challenge. Some HTTP servers have security realms that apply only to a transaction as well as being connetion based.

Issue 3.3: AGREED

Do the values for "notify-uri-schemes-supported" include the ':' character?

Proposed Resolution:

No. See rfc2911 section4.1.6 uri scheme data type variables

Action:

Added the following text to the ipp-not specification.
"Note: According to [RFC1738] the ":" terminates the scheme and so is not part of the scheme. Therefore, values of this attribute do not include the ":"."

Issue 3.4: AGREED

For get-printer-attributes operation submitted with an unsupported "requested-attributes" value what is the return code and should an unsupported attributes group be returned containing the requested-attributes attribute and the unsupported value. There are four possibilities of status code and unsupported attribute:

A) successful-ok/no attributes
B) successful-ok/unsupported requested-attributes returned
C) Successful-attribute-or-value-ignored/ no attributes
D) Successful-attribute-or-value-ignored/ unsupported requested-attributes returned

The standard currently allows C and D. Should the standard be relaxed to include C. The implementations at the Bake-Off supported were A-11, B-1, C-3, D-0

Proposed Resolution:

Recommend D, allow C and warn client implementers about A.

Action:

IIG updated with
"For the following success status codes, the requested attributes are returned in Group 3 in the response:

successful-ok: no operation attributes or values were substituted or ignored (same as Print-Job)and no requested attributes were unsupported.
Note to client implementers: If the client requests attributes that are not supported, the Printer is supposed to return 'successful-ok-ignored-or-substituted-attributes', rather than 'successful-ok'. However, a number of implementations have been found not to conform to this requirement, so clients should be tolerant of such Printers.

successful-ok-ignored-or-substituted-attributes: The "requested-attributes" operation attribute SHOULD be returned with the unsupported values in the Unsupported Attributes Group.

Note to client implementers: Although not recommended, the Unsupported Attribute Group and its contents may be omitted. Clients should be prepared for this behavior.

Issue 3.5: AGREED

In the subscription object is the does the mailto URL contain '//'. Is it <mailto://mumble> or <mailto:mumble> ?

Proposed resolution:

The mailto URL does not include '//'.

Action:

The mailto notify document will be updated with a caveat. Here is the complete updated text:

5.2.1 notify-recipient-uri (uri)

This section describes the syntax of the value of this attribute for the 'mailto' Delivery Method. The syntax for values of this attribute for other Delivery Method is defined in other Delivery Method Documents.

In order to support the 'mailto' Delivery Method, the Printer MUST support the following syntax for the 'mailto' Delivery Method when the Printer uses SMTP. The line below use RFC 822 syntax rules and terms.

"mailto:" mailbox

Note: the above syntax allows 1 occurrence of 'mailbox'. The occurrence of 'mailbox' represents an email address of a Notification Recipient.

For SMTP, the phrase 'address part' of the "notify-recipient-uri" attribute value refers to the 'mailbox' part of the value. Example:

mailto:jones@acme.com

Unlike other URLs, the mailto scheme MUST NOT use // after the colon.

The Printer MAY support other syntax for the 'address part' if it supports email protocols in addition to SMTP.

Issue 3.6: AGREED

Are there suffixes to "printer-state-reasons" value "none" (i.e. none-error & none-report)?

Proposed Resolution:

Recommend that no suffixes be used for the value "none".

Action:

Add the following text to the IIG.

"Is a suffix needed for the "printer-state-reasons" 'none' value (Issue 3.6)?
The values of the 'printer-state-reasons" MAY have suffixes of '-report', '-warning', and '-error'. If none of these suffixes is included, the meaning is the same as 'error', i.e., the Printer is stopped. However, for the 'none' value it is RECOMMENDED that no suffix be included, even though the Printer is not stopped. However, some implementations do include the '-report'

suffix, i.e., return ' none-report'. There is no semantic difference between the “printer-state-reasons” of ‘none’, ‘none-report’, and ‘none-error’. They all mean that no additional information on the printer’s state is available. “

Issue 3.7: AGREED

What is the attribute syntax for the “notify-status-code” attribute?

Proposed Resolution:

It should be a type2 enum (which is a 32-bit integer, but the values are constrained to 16 significant bits with the 16 high order bits always being zero, so that status codes values can be used here).

Action:

Add the following text to the IPP Notification specification in section 11.1.1.2:

“notify-status-code” (type2 enum):

Indicates the status of this subscription (see section **Error! Reference source not found.** for the status code definitions). Section **Error! Reference source not found.** defines when this attribute MUST be present in this group.

Issue 3.8: AGREED

When MUST Subscription Attributes groups be returned in Subscription Creation responses and when MUST the they not be returned? The current spec is too constraining on when they MUST NOT be returned.

Proposed Resolution:

Require them to be returned unless the entire request cannot be interpreted.

Action:

Add the following text to the IPP Notification specification in section 11.1.1.2:

Group 3-N: Subscription Attributes

These groups MUST be returned unless the Printer is unable to interpret the entire request, e.g., the “status-code” parameter returned in Group 1 has the value: ‘client-error-bad-request’.