

Minutes of IPP Working Group Meeting

February 9-10, 2000

1. Meeting Attendees

Shigeru Ueda	Canon
Lee Farrell	Canon Information Systems
Bill Wagner	DPI/Net Silicon
Michael Wu	Heidelberg Digital
Dave Kuntz	Hewlett Packard
Sandra Matts	Hewlett Packard
Trevor Wells	Hewlett Packard
Ron Bergman	Hitachi-Koki
Harry Lewis	IBM
Henrik Holst	i-data
Peter Lefkin	IEEE ISTO
Stuart Rowley	Kyocera
Don Wright	Lexmark
Raymond Lutz	MFPA/Cognisys
Hugo Parra	Novell
Gail Songer	Peerless
Satoshi Fujitami	Ricoh
Chuck Adams	Xerox
Tom Hastings	Xerox
Bob Herriot	Xerox
Carl-Uno Manros (Chair)	Xerox
Peter Zehler	Xerox

2. Day 1

Carl Uno-Manros opened the IPP meeting and provided the suggested agenda topics:

- Proposed charter for the IETF IPPEXT WG and the latest organizational changes in IETF
- How to organize the IETF vs. PWG parts of the IPP work
- New IPP Operations
- How to proceed with our earlier drafts on Set 2 and Set 3 Operations
- IPP Notifications – delivery methods and Collection syntax
- Possible input for IPP/1.1 Implementer's Guide
- PWG – Production Printing

[NOTE: At the October meeting, it was decided that Tom Hastings would pursue the registration of different vendor document print languages as MIME types—and that he would send an e-mail on the progress of this effort. No progress has yet been reported.]

2.1 IPP Web Master/Mistress

Gail Songer graciously volunteered to take on the responsibility of maintaining the IPP website.

2.2 IPP Document Priorities

Carl-Uno reviewed the various documents and their intended completion schedule:

- Job and Printer Set Operations – March
- Set 2 and Set 3 Operations – June
- Notifications – ???

The group agreed on the following order for addressing the documents:

1. Set Operations
2. Notifications
3. Set 2/Set 3

2.3 IETF vs. PWG: Different Parts of IPP Work

Carl-Uno is interested in addressing the logistical issues of separating IETF related work vs. PWG related work. Specifically, the Production Printing items are probably only of interest to a smaller subset of the IPP group. How should we proceed with regard to addressing this topic?

More than half of the group indicated interest in the general topic of Production Printing (as related to IPP.) Chuck Adams asked about the Requirements list, but none had been identified yet.

Carl-Uno requested that a new reflector should be set up for the Production Printing effort—separate to the current IPP mailing list. This should only be for IPP-related activity that is *not* intended for IETF-based IPP work.

2.4 New IPP “Set” Operations

Tom Hastings led a review of the Job and Printer Set Operations document:

ftp://ftp.pwg.org/pub/pwg/ipp/new_OPS/ipp-job-printer-set-ops-000130.pdf

Three optional operations are defined:

- Set Printer Attributes
- Set Job Attributes
- Get Printer Supported Values

Tom provided an overview of the various sections of the document.

It was noted that the values for the Operation-Ids still need to be assigned.

The group discussed the issue of side effects caused by setting attribute values. It was suggested that an implementation should try to set *all* values specified in the operation before checking for valid results. If any conflict results, then the operation should fail (be rejected.)

ISSUE: What number of attributes should be established for compliant implementations as the minimum supported? Is only one attribute sufficient?

[At least one example has been cited where an operation requires three “parallel” attributes: printer uri supported, uri security supported, and uri authentication supported. During the discussion, this set of attributes was referred to as “the three musketeers.”]

It was noted that the error code for “Request entity is too large” could be used to indicate “too many attributes,” but this might be inefficient if an implementation only supports a maximum of one attribute. What logic would be used to pick the (correct) smaller number? Perhaps a more informative error message should be defined.

After long discussions, the group agreed that the specification should not require an implementation to support more than one attribute. [However, everyone also agreed that it would be a poor implementation.]

The group agreed that the specification should be modified to say that a printer *should* (not **MUST**) accept the Set Printer Attributes operation in any state.

Bob Herriot suggested that the strategy to be used by a client implementation when it receives a “Request entity is too large” error will be unnecessarily complex. He thinks the entire issue could be simplified by having the operation only cause a single “set” at a time.

Hugo Parra suggested that parallel attribute updates should be handled by the use of a collection as a single “pseudo-attribute.” Carl-Uno noted that this would create a dependency of the Set Operations document on the Collection Syntax document.

The group (temporarily) agreed to modify the set operation such that it can set one and only one attribute. The printer must reject (“bad request”) any operation that contains more than one attribute. A new “multi-attribute” will be defined as a collection containing the parallel attributes printer uri supported/uri security supported/uri authentication supported.

Subsequent discussion caused the group to reject the agreement above.

Tom, Henrik, and Bob agreed to take the issue offline and work on developing a recommended solution to this issue.

While discussing “job-message-from-operator” and “printer-message-from-operator,” it was agreed that using “no-value” is acceptable, and it has the same effect and semantic meaning as a zero length string.

There was some question about whether the “default” out-of-band value could be eliminated. It was agreed that “no-value” could be used instead—at least for Set Printer Attributes. However, there was some debate about whether this should be used as a means of removing an existing attribute for Jobs. The group agreed that a separate out-of-band value called “delete-attribute” should be defined for Set Job Attributes.

Bob presented his proposal for a new collection attribute that includes the three parallel attributes mentioned above. His proposal will be distributed to the e-mail list for further review and comment.

During Bob's presentation, Carl-Uno suggested that the CONNEG WG syntax could be used in connection with the parallel attributes. No conclusion was reached on this suggestion.

[Even though there were no Issues explicitly identified in the document, the various discussions that occurred as a result of the review process had occupied the entire afternoon.]

2.5 Collections

Tom Hastings led a review of the 'collection' attribute syntax document:

ftp://ftp.pwg.org/pub/pwg/ipp/new_COL/ipp-collection-attr-syntax-991208.pdf

There was much discussion about maintaining compatibility with earlier implementations (prior to collection syntax) while still providing collection syntax for use by later implementations. Various alternative methods for determining the client capability were considered. It was generally agreed that an attribute should be defined that indicates "collection-attributes-supported."

3. Day 2

3.1 PWG IPP Archives

Don Wright suggested that the individual editors of past documents should clean up the ftp site by removing old document versions that are not .doc file format (e.g. .pdf, .txt, .htm, etc.) Tom, Gail, and Carl-Uno will work together to investigate the benefits of establishing a different hierarchy organization of the file directories.

3.2 Possible Input for IPP/1.1 Implementer's Guide

Peter Zehler discussed some of the issues/questions that have been raised on the mail list regarding the implementation of IPP. He reviewed his answers with the group, and solicited opinions on whether the current documents should be modified to provide further clarification and whether the information should be included in the Implementer's Guide.

1. For Language and String, should the max length of 1024 be for the String alone, or the *combined* length of both?

Evidently, different implementations have used both interpretations. The group agreed that the specification should explicitly clarify which interpretation is intended. After some discussion of which companies will be affected by which choice is ultimately selected, the group agreed that the max length will apply to the String only—independently of the size of the Language. This text clarification will be done to the Model document, and also explained in the Implementer's Guide.

2. Is the responsibility of the "Resume-Printer" operation limited to changing the Printer objects 'printer-state-reasons' or does it have any bearing on the 'Printer-state' also?

The basis of this question is not well understood. After reading the text in the RFC about Resume Printer Operation, the group decided that the Model document should be modified to say that the Printer "is free to" (i.e. it is not mandatory to) transition itself into a processing or idle state.

3. Assume a client has performed a Print-URI operation with a reference to a large document .The request is processed checking for the uri-scheme and the accessibility of the document, and the client is sent a response with a valid job-id. While the IPP server is downloading the document mid way, the client does a get-job-attributes. We would return the job-state as pending-held , but what the job-state-reasons say. There is a job-state-reason 'job-incoming', but is limited to create-job.

The group determined that the original answer to this question provided by Peter on the mail list was incorrect. Tom Hastings will provide an updated response—including any necessary clarification.

4. Assume a user sends a Create-Job request and is successful. Without performing a send URI/Doc operation he issues a last-doc without any document data (so the IPP server would have a job that is created without any data.) What should be the response of the IPP server to this last-doc? Will a client-error-not-possible (0x0404) status-code suffice?

No, the operation should succeed with a successful-complete status.

5. Restart on a Print-URI request that was aborted while downloading the URI: If I issue a Restart-Job request on a job that is in state "aborted" (due to download failing), does the IPP Server need to download the file again or return "client-error-not-possible"?

It *shall* attempt to retrieve the job data again. (The 'print-uri' does *not* necessarily mean download. A client may upload the file to the printer via FTP. The URL used in 'print-uri' could have a method of 'file://')

The Implementer's Guide should clarify that Print URI can be a "push." Also, the state transition diagram should be added.

ACTION: Tom Hastings will attempt to create a set of state transition diagrams for inclusion in the document(s).

The other issues/questions that were raised did not require any changes to the existing documents. In some cases, Peter will need to modify his previous e-mail response(s).

Harry raised the issue that multiple documents in a single job could possibly be received—and printed—out of order. This concern was originally raised by Carl Kugler. The group agreed that Carl should write a proposal for addressing this issue—possibly by defining an attribute that indicates “document number within a job.”

3.3 IPP Notifications

Tom Hastings led a review of the IPP Event Notification Specification document:

ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-not-spec-000202.pdf

ISSUE 01 – Once a number of delivery solutions have been developed and evaluated, we may want to make one or several of them REQUIRED for implementation to ensure a minimum set of interoperability. Which one or ones should be REQUIRED?
Deferred.

ISSUE 02 – Ok that "job-uri" isn't defined for use with the Create-Job-Subscription operation?
No, we should be consistent with IPP/1.1; "job-uri" should be defined.

ISSUE 03 – Should Renew-Subscription Request have a Group 2 for its two Subscription attributes?
Yes.

ISSUE 04 – Should Renew-Subscription Response have a Group 3 for its two Subscription attributes?
Yes.

ISSUE 05 – The out-of-band 'none' value is also defined in the PWG Production Printing document.
Should we keep the definition here in the Notification document? And/or should we move the definition to the 'collection' specification (ipp-coll) which is also defining the tag for the 'collection' attribute syntax that is the province of the IPP Transport and Encoding document?
Put it in the Collection document.

Conformance Requirements Section – The method for determining notification support should be clarified more. Also, it was suggested that this information should be moved to section 8.1.1 and included as a "NOTE". Similar information should be supplied for Section 8.2.1.

ISSUE 06 - Should we add references to the notification delivery documents that are in progress?
Yes, keep the references up-to-date. The Notification documents should be submitted to the IETF as a complete suite.

In Section 15, the status code values still need to be defined. Also, in other sections, the operation code values should be defined in Section 8.

It was suggested that a common list of code values should be maintained in a central location on the PWG website to keep track of assigned numbers.

3.4 Notifications Over SMTP

Tom led a review of a first-draft version of the "ipp-notify-mailto" document:

ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-notify-mailto-000209.pdf

It was noted that some Firewalls currently restrict e-mail attachments from being delivered. As such, it is important to *at least* provide a method for in-line text/plain messages.

Issues/questions raised during the review:

- Should we allow machine consumable format for notification—either optional or not?
[The group agreed that Machine Consumable format should be allowed.]
- XML in text body? Disguised as text/plain or application/xml only?
- Send multiple MIME types?
[Yes, this is desirable—the example of a multimedia greeting card was used.]
- Require text/plain to be supported?
[Yes.]

The group agreed to rename the notification format attribute to “notify-format” *without* specifying human consumable or machine consumable in the name. The MIME media type will determine the specific format of the message—including multipart MIME. If the notify-format is omitted, the implementation MAY send multiple MIME types, but MUST send at least text/plain.

It was agreed that when the subscribing user selects the ‘mailto:’ delivery scheme, the client MUST provide the user’s mail address.

Tom Hastings and/or Henrik Holst will update the document to reflect the decisions reached during the review and post it for group access.

3.5 Notification via Polling

Harry Lewis led a review of the IPP Notification Polling document:

[ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-notify-poll-000202.pdf](http://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-notify-poll-000202.pdf)

The concept of “server-directed polling” was explained. The Printer will provide the client with information that indicates a suggested “time window” for issuing the next event poll.

There was a long discussion about the merits and drawbacks of using multipart MIME for handling multiple events.

ISSUE 01 – Should it be possible for a client to ask for the Per-Job Subscriptions for a particular job using a “job-id”, instead of the subscription-id, which currently isn’t returned by a Job Creation operation?

Yes.

ISSUE 02 – Is there anything useful that we could define for the rest of the “notification-recipient” (uri) attribute, since there is no recipient address needed after the ‘ipp-notify-poll://’ since the recipient(s) poll?

Possibly. For example, IP address or Domain Name of the (authenticated) host.

Bob Herriot suggested that we should add job-id as an attribute to Get Notifications.

ISSUE 03 – [There is no ISSUE 03 in the document.]

ISSUE 04 – Or MUST the Printer discard events that occurred earlier than the sliding time window specified by the difference between these two values? Otherwise, the clients may get back a lot of duplicate events on subsequent requests.

No. The Printer MUST keep the events at least as long as the time window.

ISSUE 05 – If we don’t want to have Job Creation operations return subscription id’s, then allow a “job-ids” operation attribute in the Get-Notifications request in addition to the “subscription-ids” operation attribute.

Yes.

3.6 Notification Delivery Method and Protocol

Hugo Parra led a review of the ‘ipp-notify-send’ document:

ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-notify-send-000202.pdf

Hugo described his proposed protocol for an IPP Printer communicating with a “Notification Delivery Service” that is responsible for sending notification messages to the subscribing client. This configuration simplifies the notification service support required in the Printer.

ISSUE 01 – What should the name of this delivery method and protocol be that we use in the title of this document?

IPP Notification Delivery Protocol (indp).

ISSUE 02 – What should the scheme name be? Consider ‘ipp-notify-send’ a working title, until we see several schemes. The ‘ipp-notify-poll’ delivery method is another example. The IETF likes words or well-recognized acronyms, not abbreviations in scheme names, so let’s include “notify”?

[It was noted that hyphens are not allowed in IETF scheme names (hyphens are reserved to separate the responsible organization name from the scheme name.)] The group agreed to call the scheme “indp”.

ISSUE 03 – Should the scheme name be used in the title?

Yes.

ISSUE 04 – Because “human-readable-report” has been added to the Notification Model document, is it ok to change this description to be a reference to “human-readable-report” in that document?
???

The group agreed that a separate notification attribute—“human-readable-report-format”—will be added.

ISSUE 05 – Should we move the status codes into the Notification Model document in order to have the same status codes for any other delivery method that might be defined?

???

Carl-Uno suggested that the document should be split into two separate drafts—one for the protocol and one for the delivery method. He believes that having separate documents would make future extensions easier.

ACTION: Hugo Parra will try to get at least the Delivery Method document published before the IETF deadline.

3.7 Collections – continued

Bob Herriot presented some information about using the CONNEG syntax to describe collections of attributes. He explained several samples that are contained in the CONNEG Content Features Schema specification and then provided some suggested encodings to handle the “parallel attributes” that were discussed during the Set Operation document review.

IPP Meeting, February 9-10, 2000

After a long discussion about the parallel attributes (and “the three musketeers”), the group agreed that the Set Operations document will reference *neither* collections nor the CONNEG syntax as methods for setting or validating—at least for now.

IPP Meeting adjourned.