

Notes and Agreements on Notification, 7/7/99-7/8/99

From: Bob Herriot and Tom Hastings

Date: 07/08/1999

File: notification-agreements-990708.doc

Bob Herriot led the IPP review of the Notification Requirements document and the Notification specs at the 7/7/99-7/8/99 IETF IPP WG meeting on Copenhagen. He generated the following notes and agreements that were reached. The unnumbered issues are new issues raised. The number issues refer to the numbered issues in the specifications.

As always, these agreements are being sent to the IPP DL for further discussion and final consensus.

Notification requirements

The following agreements were reached and issues raised on the "Notification Requirements" Internet-Draft <draft-ietf-ipp-not-02.txt>, dated June 24, 1999:

ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-notification-requirements-990624.pdf

1. Section 2.15 need better wording for subscription, e.g.
 - any set of traps for a specific job and
 - any set of traps for a printer and jobs with no control over which jobs.
2. ISSUE: Section 2.19 Queued notification definition is flawed. Look at Jini for ideas.
3. ISSUE: Section 2.20 Is it "Reliable Transport" or just Reliable that is important?
4. Section 3. item 3 last sentence doesn't make sense. Also reword to include printer subscription for all events and for specific job events.
5. Section 3 item 6. Strike the word "that"
6. Section 3 item 8 and 9, combine that 9, which qualifies 8 comes first and is part of the same item.
7. Section 3 item 11: change "must" to "may".
8. Section 3 item 13 change "Event Source" to "Notification Source"
9. Section 3 item 14: remove. It is not an IPP issue. It is a value-add implementation issue.

10. Section 4, item 3, change "form" to "firm".
11. Section 4, item 4, why is "type" "email" instead of "queued"
12. Section 4, item 7, delete word "audit" and change "accounting" to "usage statistics".
13. There is no requirement for batching events or keeping them separate. Note that sequence numbering of events is harder with batching. We decided to abandon batching and send events one by one.

2. Notification Specification

The following agreements were reached and issues raised on the "IPP Event Notification" document, dated May 18, 1999:

ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-notification-990518.pdf

14. Move section 3 to just before section 1 so that model is defined before details.
15. Events are not limited by scheme of delivery method.
16. Who is "device-trigger-date-time" optional for. Isn't it required if the printer implements current-date-time?
17. Section 3. B is subscribed to by "first" party not "third" party.
18. Section 3 change "device" to "printer".
19. Rename "event report" to "notification"

Section 4: New Subscription Operation attributes

20. Section 4: #2 need more details for "restart" operation. E.g. how does sequence number get communicated. Explain how job-description attributes say how to generate a new subscription. Reference to ipp-ops-set1 should be updated to IPP/1.1.
21. ISSUE 1, yes you might have to drop events. Subscriber cannot state whether to allow dropping of events, but an implementation may allow an admin to configure policy on what to drop, but it is beyond the protocol. There should not be an event-dropping event because it would just add to the congestion and might not get delivered. Instead, a subscriber SHOULD start a separate timer for renewing leases in order to be really robust.
22. Section 4.1.1 fix 2nd paragraph. Maybe delete it.
23. Section 4.1. keep http but remove issue

24. Section 4.1 move snmp traps to some separate document
25. Section 4.1 remove ndps-notify
26. Section 4.1 remove sense-notify
27. ISSUE: Why does Jini have synchronous notify using RMI, i.e., notify event report is delivered to the Notification Recipient as a request and the Notification Recipient returns a response? Should we change the notification delivery for IPP to be a request with a response? For which delivery methods? Is this a Quality of Service option that the subscriber can choose?
28. ISSUE: For TCP/IP, what about leaving notification connection open versus having to reestablish a connection for each event?
29. ISSUE: Ok to add an octet stream for an attribute for subscription, like Marshalled object in Jini? It provides a way for a subscriber to pass opaque parameters to the recipient of the event.
30. Need to get back a 1setOf sequence numbers in CreateJob/PrintJob response, if we can subscribe to more than one event in an operation.

Section 5: Event Report Content

31. There is no requirement for batching events or keeping them separate. Note that sequence numbering of events is harder with batching. We decided to abandon batching and send events one by one in each report. In order to reduce the number of event reports sent, each event is defined to be disjoint. So the 'job-completed' event does not include 'job-state-changed' events. Also 'job-created' event does not include 'job-state-changed' events. Also 'job-state-reasons-changed' is merged with 'job-state-changed' event (see 34 below). Etc. Same for Printer events.
32. Issue 7: yes, have a sequence number and put it into the 32-bit "request-id" field. The event sequence number will be kept by event for each Job and Printer object. So change "job-trigger-events" (1setOf type2 keyword) to "job-trigger-events" (type2 keyword). Same for "printer-trigger-events".
33. Request-id is not zero any longer. It is an integer sequence number (0:MAX).
34. Probably should remove state-reason change event and have state-change return events for state and/or state-reason changes. The values returned should include previous-state, state, state-reasons, plus reasons-added and reason-deleted (instead of previous reasons). The latter sentence describes ideas not fully resolved.
35. Device-powered-up/down should be printer-shutdown and printer-restarted. Actually they should be merged into the state-change event. Likewise, printer-accepting-jobs should be a state-change event rather than config change. The document needs to be more precise about which event is associated with each attribute change.

36. Jobs are missing an event for monitor non-job-state changes, such as job-name or job-template attribute changes.
37. Remove last 3 printer events ('device-config-change', 'ready-for-job', and 'ready-for-just-in-time-job') for job submission because they make sense only in the printer subscription (Subscribe-Printer) case. But a later discussion to combine the two notification specs into a single spec may make it ok to leave these events in.
38. Issue 10 is moot because we have only single values for event reports.
39. Talk with Larry about multipart/report and other alternatives
40. Security issues not addressed.

Printer Notification

The following agreements were reached and issues raised on the IPP Event Notification document: Job Independent Subscriptions, dated May 19, 1999:

ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-notification-printer-990519.pdf

41. Section 1: Subscription should be generated by server, not client. This is correct in section 4 but not in summary. Fix summary.
42. Section 2: Pictures should eliminate job-creation subscriptions. They confuse the printer subscription. Change arrow from "job-creation subscriptions" to "job submissions".
43. Drop D and E and use words to describe them as examples of A.
44. Subscription leasing times should be handled by the client, i.e. It has an alarm.
45. Section 4.1. The second sentence needs rewording to clarify antecedent of "it" and "its".
46. ISSUE: Does a client need to be able to get a list of current subscriptions? (Carl-Uno's issue)
47. Issue 1: A printer MAY suppress sending duplicate notifications (same events to same notification recipient), but it MUST allow duplication subscriptions so that Unsubscribe-Printer (Unsubscribe-Job) doesn't unsubscribe both subscriptions. Since the sequence number is kept by event for each object, rather than for each subscription, duplicate notifications will get the same sequence number, so the recipient can tell if the duplicates aren't suppressed. Eventually, an older duplicate due to a subscriber crash and restart will be aged out using the lease mechanism, if the implementation doesn't filter out duplicate notifications. A subscriber could

record the subscription ID locally so that if the subscriber crashes and is restarted, it can renew the old subscription, rather than creating a new subscription.

48. Section 4.1 add "event-sequence-numbers" (1setOf integer (0:MAX) attribute to the Subscribe-Printer response. This is an attribute that is parallel to the "notify-events" (1setOf type2 keyword) operation attribute. Sequence numbers are kept for each event for each Job and Printer object. The returned value indicates the sequence number returned for the last event delivery, for each event being subscribed to. The next event delivery will supply a sequence-number in the event report with a value one more than this value. An event that has never occurred will return a sequence number value of 0 in the subscription response. Thus the first sequence number for each event/object pair that is actually delivered to a Notification Recipient will be 1.
49. Issue 2: Renew operation must not allow subscription-id to change. Therefore, don't need "subscription-id" to come back, so remove it from the response.
50. Renew has other errors, such as not –authorized and lease-denied.
51. Section 5.3 get rid of because client does renewal with alarm.
52. Issue 3: detailed-status-message solves the problem. Put in implementer's guide.
53. Issue 4: maybe have unsubscribe ALL. Group undecided.
54. Issue 5: NO
55. Issue 6: Expect that subscriptions would survive, but low end might lose everything. Rebooting tries to preserve lease and time of lease, which should be no shorter than before the crash.
56. Issue 7: yes
57. Add Subscribe-Job operation that allows a client to add a subscription to a previously submitted job. Change the name of the Subscribe operation to Subscribe-Printer to clarify the target of these two operations.
58. Add an OPTIONAL Unsubscribe-Job operation that would be useful for a user to stop getting events from a Job, such as sheet-completions, while the job was being processed. Add "subscription-id" operation attribute to the create job operations that returns an implicit subscription in create job operations (Create-Job, Print-Job, Print-URI) so that an Unsubscribe-Job works, if Unsubscribe-Job operation is supported.

Job Progress Attributes and Event Report Content

The following agreements were reached and issues raised on the "Job Progress Attributes and Event Report Content" document, dated May 19, 1999:

ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-job-prog-attr-990519.pdf

This specification was not covered at the meeting. It contains the following six issues which need to be resolved:

59. ISSUE - Should the "job-collation-type" (type2 enum) attribute syntax be 'type2 keyword' to go with "multiple-document-handling(type2 keyword)", instead of the Job MIB type2 enum syntax?
60. ISSUE - For the "job-collation-type" (type2 enum) attribute should we use the out-of-band 'unknown' value, instead of this Job Monitoring MIB '2' enum value for 'unknown'?
61. ISSUE - For "sheet-completed-copy-number" (integer(-2:MAX)) should we change the lower limit to 0 and use the IPP out-of-bound values: 'unknown', instead of the '-2' value?
62. ISSUE - For "sheet-completed-document-number" (integer(-2:MAX)) should we change the lower limit to 0 and use the IPP out-of-bound values: 'unknown' instead of the '-2' value?
63. ISSUE - For "impressions-interpreted" (integer(-2:MAX)) should we change the lower limit to 0 and use the IPP out-of-bound values: 'unknown' instead of the '-2' value?
64. ISSUE - For "impressions-completed-current-copy" (integer(-2:MAX)) should we change the lower limit to 0 and use the IPP out-of-bound values: 'unknown' instead of the '-2' value?