From:	Hastings, Tom N [hastings@cp10.es.xerox.com]
Sent:	Monday, September 13, 1999 14:57
To:	ipp
Subject:	IPP> NOT - Proposed alternative to make Per-Job like Per-Printer subsc riptions

This mail message proposes making the Per-Job subscription mechanism more
like the Per-Printer subscription mechanism, now that we have agreed at the
IPP WG August meeting (and put into the first Internet-Draft) that the
Per-Printer mechanism is REQUIRED if doing notification, as well as the
Per-Job mechanism. The Per-Printer mechanism is more object-oriented, like
the rest of IPP.

This mail note summarizes the current notification proposal just posted with
a September 9, 1999 date (updated to resolve some of the issues in the
August 25, 1999 first Internet-Draft) and compares it with the proposed
alternative, dated September 10, 1999.

I've posted the complete proposed alternative at:

ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-not-spec-alt-990910.doc
ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-not-spec-alt-990910.pdf
ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-not-spec-alt-990910-rev.doc
ftp://ftp.pwg.org/pub/pwg/ipp/new_NOT/ipp-not-spec-alt-990910-rev.pdf

The revision marks show the changes from the September 9, 1999 version just
posted.
The same issues as the September 9, 1999 version with one addition:

ISSUE 5a - Shouldn't we add a "number-of-job-subscriptions" Job Description
attribute, since the client is NOT assured that it can determine the number
of outstanding Per-Job Subscriptions by doing Get-Subscriptions because of
access control and since there are no Job Descriptions attributes associated
with notification.

This alternative will be discussed at the Wednesday, 9/15, IPP telecon, at
the IPP WG meeting, 9/22-23 in Denver and on the mailing list.

        Current Notification Specification, Sept 9, 1999 (First
Internet-Draft)

The first Internet-Draft, dated August 25, reflects the agreements from the
August 18-19 IPP WG meeting in Alaska. The first Internet-Draft and the
Sept 9 spec (just posted) have the following:

1. The client creates Per-Job subscriptions by supplying the "job-notify"
operation attribute in a Job Creation operation (Create-Job, Print-Job,
Print-URI). The "job-notify" attribute uses the new 'collection' attribute
syntax. The attribute syntax of the "job-notify" attribute is '1setOf
collection', so that the client can supply one or more Per-Job subscriptions
when creating the Job. The Printer copies the "job-notify" attribute to the
Job object as a Job Description attribute which may be queried and modified
subsequently.

2. Other operations on Per-Job subscriptions are done using general
operations on the Job object:

a. add, remove, or modify subscriptions: the new Set-Job-Attributes operation (part of the Set 2 proposal a ka Admin operations) that replaces the entire "job-notify (1setOf collection)" Job Description attribute with a new set of collection values.

b. query: the existing Get-Job-Attributes and Get-Jobs requesting the "job-notify (1setOf collection)" Job Description attribute.

3. Per-Printer subscriptions have a completely separate set of operations, create a Subscription object, and don't use the 'collection' attribute syntax:

a. create and add subscriptions: Create-Printer-Subscription
b. query subscriptions: Get-Printer-Subscriptions and Get-Printer-Subscription-Attributes
c. remove: Cancel-Printer-Subscription
d. renew: Renew-Printer-Subscription

SUMMARY OF THE PROPOSED ALTERNATIVE:

Now that Per-Printer subscriptions have to be supported if Per-Job subscriptions are supported, we suggest an alternative that changes the Per-Job subscriptions to be much more parallel to the Per-Printer subscription.

We suggest that the "job-notify (1setOf collection)" operation attribute in the Job Creation operations (Create-Job, Print-Job, and Print-URI) create one Subscription object for each collection value, instead of being copied to the "job-notify (1setOf collection)" Job Description object. Then the "job-notify (1setOf collection)" would NOT be a Job Description attribute. Also the Subscription object would be used to represent both Per-Job and Per-Printer subscriptions.

We suggest that the names of the 5 Subscription object operations from the August spec have the word "Printer-" removed from their names, so that the Subscription object may be used for Per-Job subscriptions as well as Per-Printer subscriptions. Here are the 5 operations:

a. create and add subscriptions: Create-Subscription
For Per-Job subscriptions, the client supplies the "job-id" so this operation can happen only AFTER the job is created to add more subscriptions to a job.
The Printer returns a unique "subscription-id". Whether or not the id has structure to help distinguish between Per-Job and Per-Printer subscriptions depends on the implementation and is transparent to the client.

b. query subscriptions: Get-Subscriptions
The client supplies the "job-id" as a simple filter to indicate that all the Per-Job subscriptions are to be returned for that job. Absence of the "job-id" returns all of the Per-Printer subscriptions.

The following operations do not support a "job-id" input parameter. Instead, the "subscription-id" (and the "printer-uri") are sufficient to uniquely identify the Subscription object instance:

c. query: Get-Subscription-Attributes.
Has a "my-subscriptions" boolean filter, just like Get-Jobs.

d. remove: Cancel-Subscription

e. renew: Renew-Subscription
   Lease-time-requested MUST be 0 for Per-Job subscriptions, so that
subscriptions could be added for other objects in the future). [In the
posted proposal, its Renew-Printer-Subscription, but that name should be
changed so that the Subscription object can be any Subscription object.]

Advantages of this alternative proposal:

1. Per-Job and Per-Printer subscriptions are much more parallel. The
Subscription object is used for both Per-Job and Per-Printer subscriptions
with the same attributes. The same 5 Subscription object operations work
for both Per-Job and Per-Printer subscriptions. This should simplify code,
reduce the code size, and increase the understandability of the
specification for both Printer and client implementations.

2. Users will understand the semantics more easily, since the parallelism
between Per-Job and Per-Printer means learning one set of semantics, not
two.

3. Aligns with object oriented semantics, like those of Jini printing.

4. More implementer flexibility: the implementer can allocate Per-Job
subscription objects within each Job object or in the same pool as the
Per-Printer subscription objects transparently to the client.

5. Improved interoperability, since the way to change Per-Job subscriptions
was OPTIONAL in the August spec (Set-Job-Attributes was OPTIONAL). Now the
client knows that Create-Subscription and Cancel-Subscription will always
work for Per-Job operations.

Thanks,
Tom