

Attendees: Jerry Thrasher, Pete, Gail, Dave, Ira, Bob, Tom, Dennis, Harry, Lee

Agenda:

1. Feature creep for the Document object spec, IPPFAX, and PSI
2. IPP/1.2
Week after next.
3. Split document (see agreement 6 below)?
 - a. Document object - needed by PSI, FSG JTAPI, FSG PAPI, FSG Driver API
 - b. Non-document features (operations and attributes) needed by IPPFAX and PSI
 - c. Catch all that can take more time and that can have both Job and Document attributes.
4. 4+2+1 conformance increases
No support. So this be IPP/1.2.
5. Michael Sweet's objections to Document Object spec.
Week after next.
Ira send email to Michael to provide more details about objections to the current April 7 spec.

From: Dennis Carney [dcarney@us.ibm.com]
Sent: Monday, April 28, 2003 06:27
To: ipp@pwg.org
Subject: Re: IPP> 6 agreements from the IPP Document Object Spec review, April 24, 2003

My comments below, marked with <dmc></dmc>. (The comments are only on items 3, 5, and 6.)

Dennis Carney
IBM Printing Systems

"Hastings, Tom N"
<hastings@cpl0.es .xerox.com>
Sent by: the IPP Document Object Spec review, April 24, 2003
owner-ipp@pwg.org

To: ipp@pwg.org
cc:
Subject: IPP> 6 agreements from

04/25/03 05:49 PM

Attendees: Gail, Bob, Pete, Lee, Dave, Jerry T, Tom (did I miss anyone?)

We'll have one more page by page review next Thursday, May 1. PETER: OK?
Or do you want to look at the updated document (which isn't quite done)?

We reached the following agreements:

1. Add "document-format-version-detected" Job Description attribute to go with "document-format-detected" and "document-format-details-detected" Job Description attributes.

[Agreed to add.](#)

2. Remove the feature that "job-mandatory-attributes" can supply the keyword names for the 7 document Operation/Description attributes ("compression", "document-charset", "document-digital-signature", "document-details", "document-format", "document-format-version", and "document-natural-language"). So none of these 7 document Operation/Description attributes can be validated with Create-Job, because none of them can be submitted with Create-Job. These 7 MUST be submitted with Document Creation operations when they are supplied. All of these 7 document Operation/Description attributed can be supplied in a Validate-Job.

However, the Printer will only reject Validate-Job for the two that [rfc2911] REQUIRES: "document-format" and "compression". For the other 5 if unsupported attributes or values are supplied, the Printer MUST return them in the Unsupported Attributes response with a 'successful-ok-ignored-or-substituted-attributes' status code.

[Agreed to remove.](#)

3. The comment that we need a way for a Printer to say it will accept any value for a particular attribute was discussed, including adding a general 'any'. The problem with adding 'any' as a value of the "xxx-supported" Printer attributes is that the Printer validation now needs a special case check when comparing "xxx" attributes supplied by the client with "xxx-supported" Printer attribute. Also clients have to know that they can't display 'any' as a value and have to know that they can't send 'any' in the request, even though the value is one of the values of the Printer's "xxx-supported" attribute.

So we agreed that the way already defined in [pwg5100.3] section 6.1): user-defined-values-supported (lsetOf type2 keyword) Printer Description attribute lists the Job Template (and Document Template) attributes that the Printer will accept any value. However, with the current definition, it doesn't allow the Document Operation/Description attributes to be named. So there is no way for the Printer to say it will accept any "document-format" value. However, the Printer MUST accept any other value of any of the other 7 Document Operation/Description attributes. We also agreed that for the SM Schema it was preferable to indicate by some additional attribute that the Printer would accept any value for an attribute, rather than introducing an 'any' value.

ISSUE: So we may still have an issue if there is a need for a Printer to accept any document format. One solution would be to extend the "user-defined-values-supported" to allow the "document-format" and "compression" keyword attribute values.

<dmc>

What does it mean for a printer to accept any document-format/compression?

It will accept formats it has never heard of and attempt to print them anyway, presumably without any "formatting" or decompression? Sort of like equating any unknown document format to just 'text/plain'?
</dmc>

Agree: Don't need to add to "user-defined-values-supported".

Fix the spec so that Printer can accept more versions that in the implemented list. Some of the versions. MAY not be inclusive. Only come back in the Unsupported list if rejecting the job. If accepting the job, then version doesn't come back in the Unsupported list.

"document-charset" - MUST NOT accept an unsupported charset. Note: for Printer's that don't support "document-charset" would ignore this attribute.

"document-format" - MUST NOT accept an unsupported charset

"compression" - MUST NOT accept an unsupported charset

"document-digital-signature" - MAY accept/ignore or reject unsupported values for document formats that permit the digital signature to be skipped, such as PDF. MUST reject for the other document formats.

"document-format-version" - MAY accept/ignore unsupported values or reject

"document-source-xxx" - MAY accept/ignore unsupported values or reject

"document-natural-language" - MAY accept/ignore unsupported values or reject

"document-format-device-id" - MAY accept/ignore unsupported values or reject

ISSUE: OK to extend "user-defined-values-supported" to include 'document-format' and 'compression' attribute values?

No. Don't extend.

ISSUE: If we also allow the "user-defined-values-supported" to include 'document-format-details' and its member attributes, e.g., 'document-format-details.document-source-application-version', then the Printer can say that it implements "any" version. Doesn't this solve Bob Taylor's truth in advertizing requirement? So a Printer that doesn't want to bother clients with returning versions that aren't in its implemented list, the Printer can include the 'document-format-details.document-source-application-version' value is the Printer's "user-defined-values-supported".

Agree. Fix in the spec, don't add to Printer attribute semantics to indicate.

4. Add '[job-]errors-detected' value to "job-state-reasons" and "document-state-reasons", but do not add "[job-]errors-count" Job/Document Description attribute. Knowing the number of errors isn't more helpful, to just knowing that one or more errors occurred. Losing data is an error, while substituting some other font is only a warning, since no information was lost.

Audit trail would like to have number of errors, not just that one or more errors occurred.

Agree to add "job-errors-count" (and "job-warnings-count") Job Description attribute to catch all spec.

Agree to add "errors-count" Document Description attribute to catch all spec. Probably specify independently.

5. ISSUE: For a conversion service or a Print Service that converts the document format, there isn't a way to indicate the desired final format and there isn't a way to represent the current document format for a document that is being converted, where the current format might be different from

either the supplied format or the desired format.

<dmc>

Maybe I just have to get my mind reset, but IPP as a conversion or Print service? Isn't IPP tailored to *print*? Wouldn't we need (a bunch?) more attributes detailing where the job/document was supposed to go when it had been "converted"? I'm also wondering whether there would be (a bunch of?) attributes/state-reasons that are either meaningless, whose meaning is totally different, or whose meaning is unclear/ambiguous in the conversion case. Do we *really* want to try to go there?

</dmc>

AGREED:

[Move from the Document object spec to the Non-document spec](#)

a. Add the following 2 attributes as Job Template/Document Template attributes (but not to the "document-format-details" collection Operation attribute). [In the PWG Semantic Model these will be Processing attributes, not Description attributes](#):

"document-format-~~requested~~target" and "document-format-version-~~requested~~target"
Job

Template/Document Template attribute, so that there are also

"document-format-~~target~~requested-default" and

"document-format-version-~~target~~requested-default" Printer attributes and also

"document-format-~~target~~requested-supported" and

"document-format-version-~~target~~requested-supported" Printer attributes. And

corresponding "document-format-~~target~~requested-actual" and

"document-format-version-~~target~~requested-actual" Job Description attributes.

[PSI needs and CUPS has this too. Put in the Non-document spec.](#)

[Use case: Client specified an FTP target device and the format to convert to.](#)

b. Add the following pair to the READ-ONLY Document Description attributes:

"document-format-current" and "document-format-version-current" Document Description attributes. The Printer sets these to the "document-format" and

"document-format-version" supplied by the client and changes them as the processing proceeds, eventually winding up with the values supplied in the "document-format-requested" and "document-format-version-requested" attributes.

[CUPS doesn't have this one. PSI does NOT need.](#)

[PSI GetNextJob gets Document Description attributes with the same values as if the Service had pushed the Job to the Device.](#)

[PSI GetNextJob get back 3 parameters:](#)

[Job URI on the Service](#)

[Job Description object - doesn't have a Job URI Description \(doesn't have any state attributes\)](#)

[Put in the catch-all spec.](#)

[PSI needs and CUPS also has the equivalent of "output-device-requested" operation attribute, "output-device-requested-supported" Printer Description attribute. Put in the non-document spec.](#)

6. Suggestion to move some of the Document object spec to a separate spec.

Proposals:

1. Separate operation and attributes into REQUIRED/CONDITIONALLY REQUIRE

versus OPTIONAL for a Printer to support.

2. Move out only those things which make sense to implement even when not supporting the Document object:

- A. "job-mandatory-attributes" Job Creation Operation attribute [b](#).
- B. "document-charset" Operation attr, "-default", "-supported" [b](#)
- C. "document-digital-signature" Operation attr, "-default", "-supported" [b](#)
- D. "document-format-details" Operation attr, "-default", "-supported", "-implemented" [b](#)
- E. "document-format-version" Operation attr, "-default", "-supported" [b](#)

(B-E would have corresponding Document Description attributes ~~in~~defined only in the Document Object spec, along with "compression", "document-format", and "document-natural-language" Document Description attributes).

F. Close-Job operation [b](#)

- G. job-copies (integer(1:MAX)) Job Template attribute [b](#)
- H. job-cover-back (collection) Job Template attribute [b](#)
- I. job-cover-front (collection) Job Template attribute [b](#)
- J. job-finishings (1setOf type2 enum) Job Template attribute [b](#)
- K. job-finishings-col (1setOf collection) Job Template attribute [b](#)

- L. media-size-name (type3 keyword | name(MAX)) [Job Template \(UPnP\) c](#)
- M. media-type (type3 keyword | name(MAX)) [Job Template \(UPnP\) c](#)

- N. "ipp-attribute-fidelity" Job Description attribute [b](#)
- O. "job-mandatory-attributes" Operation/Job Description ~~in~~ attributes [b](#)

P. "pdl-override-supported" new value 'guaranteed' (which is already in see [ippsave] section 8.1) [c](#)

AGREED: Move only G through M. Keep the rest in the IPP Document object spec because they are related to Document objects, even though they might be useful when not supporting the Document object.

<dmc>

Unless I'm missing something, G-K are only already-existing Job Template attributes that have been renamed with "job-" on the front. Why would anyone want to implement these renames unless they were doing the Document object?

And M is also already-existing, in 5100.3. What is new about it that we would need a new spec?

</dmc>

Please comment on the IPP mailing list about any of these agreements.

Tom