

Notes and Agreements on Admin operations, 7/167/99

From: Bob Herriot and Tom Hastings

Date: 07/1608/1999

File: ipp-ops-admin-agreements-99071608.doc

This updated document includes agreements reached at the 7/14/99 IPP telecon. Agreements that have no comments indicate that the telecon also agreed with the WG meeting.

Bob Herriot led the IPP review of the Additional Administrative Operations document, dated June 30, 1999, 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 specification. **Unresolved ISSUES are highlighted like this in the .doc and .pdf files.**

The document being reviewed is available at:

<ftp://ftp.pwg.org/pub/pwg/ipp/proposed-registrations/operations/ipp-ops-admin-990630.pdf>

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

1. ISSUE: What happens if the value for the from-operator-message for job or printer is either a blank or an empty string for the set operation or other operations that set this attribute?
Suggested resolution: If the operation attribute or the explicit "printer-message-from-operator" (or "job-message-from-operator") is supplied in the request, but with either a zero-length or white space only content, it replaces the existing Printer or Job attribute as expected. This is the way that the operation can indicate that there is no longer a message from the operator for the Printer or Job object. However, if the operation attribute or the explicit "printer-message-from-operator" (or "job-message-from-operator") is NOT supplied in the request, the corresponding Printer or Job attribute is unchanged.
2. Printer and job operator message should also have a tick time, which is required. Date-time value should be required if the printer implements the current-date-time.
3. **ISSUE: Can a client determine the values of "when" that are supported for operations (Pause-Printer, Reset-Printer, and Shutdown-Printer)?**
We did not resolve this issue, since there could be different values of "when" for the three different operations. Adding three "when-xxx" Printer Description attributes is a possibility, but does seem overkill.
4. ISSUE 1: The 'after-current-job' value of the "when" operation attribute must be supported.
5. **ISSUE: In 13.1.5.8 server-error-printer-is-in-standby-mode, when Printer has been shutdown in 'standby-mode' (as opposed to "shutdown-function" = 'power-down'), : Restart-Printer and Get-Printer-Attributes work, but does Set work?**
Suggested resolution: We agreed that Restart-Printer and Get-Printer-Attributes must work. We also agreed, that the purpose of Shutdown-Printer to 'standby' mode, was long-term, not just a short time to, say, change a Printer attribute or change a loaded medium. However, on Restart-Printer (the next day), the operation or system administrator might want to change some attributes before allowing jobs to be

submitted or already submitted jobs to be scheduled. Therefore, we have the following issue:

ISSUE: What state the Printer comes back up on Restart-Printer and how can the client control?

Possible alternatives:

a. Restart-Printer always brings the Printer up Disabled ("printer-is-accepting-jobs" = 'false') and Paused ("printer-state" = 'stopped', and "printer-state-reasons" = 'paused'). Then the operator issues Enable-Printer and Resume-Printer when want to restore normal operation. The client can automatically issues these operations depending on GUI options.

Advantages: This is the simplest to implement, allows new states to be added without changing the Restart-Printer operation, and can have the same GUI interface as b:

b. Add a REQUIRED operation attribute to Restart-Printer, something like "printer-condition" with values: 'disabled-and-paused', 'enabled-and-paused', and 'enabled-and-idle'.

c. Have Restart-Printer restore the enable/disable and paused/idle state to what it was when the Shutdown-Printer was issued. The operation has to plan ahead at Shutdown-Printer time.

6. ISSUE: Perhaps only Restart should work when a printer is shut down, not Get-Printer-Attributes?
Suggested resolution: We disagreed. Get-Printer-Attributes must also work, so that a user can determine that the printer exists and find out that is it shutdown, why, and/or when it might come back up by querying the "printer-message-from-operator."
7. Tables for Set-Printer-Attributes and Set-Job-Attributes operations: Change all occurrences of "MUST NOT if supported" to "MUST NOT"
8. In the tables that show what attributes are settable for Set-Printer-Attributes and Set-Job-Attributes operations: should the table be replaced with a list the attributes that are REQUIRED to be read-only. Perhaps there is an attribute that lists all settable attributes.
ISSUE: Can we add just one Printer Description attribute: "settable-attributes" or do we need a "printer-settable-attributes" and a "job-settable-attributes" Printer Description attributes? What if the Interpreter and/or the Document object become real objects, would we need to add "interpreter-settable-attributes" and "document-settable-attributes" Printer Description attributes?
9. ISSUE: Do "operations-supported" and "versions-supported" reflect software support or does the software examine the administratively set values to determine behavior. That is, are they read-only or read/write?

Suggested resolution: They are not included in the REQUIRED to be read-only list in the spec. Therefore, implementer's are free to make them either read-only, meaning that they reflect the implementation and the client is unable to change via the protocol, or may make them be read-write, meaning that the client can change the behavior of the system using the protocol. In the latter case, these attributes would be included in the new "printer-settable-attributes" Printer Description attribute.

In order to allow the client to determine the values for the settable attributes that are supported by the implementation, as opposed to the current settings, it was suggested to add an operation attribute that allows the client to get the factory defaults, as opposed to the current setting, for the settable attributes. If the "factory-defaults" (boolean) is supplied with a 'true' value, the factory defaults are returned, instead of the current values, for any requested settable attribute.

10. Fidelity should go away for the set operation. The set operation should be atomic.
11. Set-Printer-Attributes operation: The "document-format" operation attribute does have issues that were left unresolved. Issue of how to resolve with attributes that do and don't vary with regard to format. For example if I set n-up and media for a format of PostScript and then Get-Printer-Attributes with a document format of text, have I changed the values of n-up and media, or just n-up because its value depends on the format but not media which doesn't. The document format is a limited version of constraints.

ISSUE: Do we continue the error with Get-Printer-Attributes by adding "document-format" to Set-Printer-Attributes or do we acknowledge that an attribute holds all values, but some may be constrained by constraints which are another attribute to set.

Suggested solution: Add an Interpreter object to the IPP Object model. Those attributes that can have values depending on the interpreter, are modeled as Interpreter object attributes, instead of Printer attributes. Those attributes that are the same for all interpreters, continue as Printer attributes. In the Get-Printer-Attributes and Set-Printer-Attributes, the "document-format" operation attribute becomes part of the target specification for those attributes that are Interpreter attributes. Implementations that have multiple interpreters, but don't have different values when validating jobs, would have a single Interpreter object that represents all interpreters.

ISSUE: Do we need a way to get and/or set the "xxx" attribute for all Interpreter objects in one Get-Printer-Attributes or Set-Printer-Attributes operation? Or is it sufficient for a client to provide the equivalent functionality by stepping through all the values of the "document-formats-supported" with repeating the Get or Set operation?

12. Keep unsupported values rules for Set-Job-Attributes consistent to Create-Job rules instead of new ones with new 'read-only' out-of-band value
13. ISSUE: Make stronger what operations do with regard to other protocols, e.g., disable should disable queue for other protocols too. People are not unanimous on this. Some believe that admin operation affect only the IPP channel; other believe it affect the entire device and thus other protocols "feel" the change.

Suggested solution: While we agreed to RECOMMEND that IPP control other protocols, we also think it would help a lot to add a Printer Description attribute that indicates whether Printer operations affect other protocols or not. This attribute would not be indicated as REQUIRED to be read-only, so that some implementer's could even allow System Administrator's to use the Set-Printer-Attributes operation to change whether these operations do or don't affect other protocols.

14. ISSUE: Is IPP intended for printer management. The issue is still undetermined?

15. Shutdown-Printer operation's behavior should perhaps be left more implementation dependent with respect to Pause/Resume of job. It is hard for us to prescribe some printer dependent behavior as to whether a job can be resumed after a "now" type of shutdown.
16. ISSUE 6 (Shutdown-Printer operation) and ISSUE 11 (Pause-Job operation): The "synchronize" attribute: it is not clear why this would ever be false. Ok to get rid of from the Shutdown-Printer and Pause-Job operations?

17. ISSUE: It isn't clear which type of checkpointing is being suggested for synchronize: checkpoint a stream or checkpoint in a job that is on a disk file in the printer.
18. ISSUE 9 (Pause-Job operation), ISSUE 12 (Pause-Current-Job operation), and ISSUE 13 (Resume-Job operation): The 'processing-stopped' state seems like the right job state for Pause-Job and Pause-Current-Job, rather than 'pending-held'.
19. Do we really need Pause-Current-Job. Most people felt that Pause-Job was sufficient.
20. Promote-Job seems to have no support. No one can see a reason. Also, it would seem that a GUI would want a drag and drop interface that would allow a job to be move to any position in a queue. This implies that we must define a model for how a queue is ordered and what moving jobs does to the ordering. For example a queue might be order by time of arrival, but the "move" operation would disrupt this temporarily. It would not change the fact that arriving jobs would still go to the end of the queue.
We disagreed with removing Promote-Job. We also agreed that the Get-Jobs operation must return the promoted job(s) first, since jobs are returned in the "expected time to complete" order. In addition we agreed that a Printer can have more than one job in the promoted condition at a time, even though the Promote-Job operation only accepts one "job-id" at a time. So if an implementation does have job queues, then the job is moved to the front of the queue. Therefore, subsequent Promote-Job operations before the previously promoted job started processing would just go in front of the previously promoted job. The trick will be to specify these semantics in such a way as to not require a queue, since that is implementation dependent.
21. ISSUE: Does we really need a Space-Current-Job. This seems very specific to roll-fed printers.
Suggested resolution: No. But do add a REQUIRED "job-id" operation attribute that a client MAY supply. If supplied, then the job-id must match the current job or the one that still has paper in the paper path. Printers that implement Space-Printer will also have a notion of what job is in the paper path, even though the job has finished marking. Remember that the job is supposed to transition to the 'completed' state when "all of the job media sheets have been successfully stacked in the appropriate output bin(s).
22. ISSUE: Is Space-Current-Job reasonable to do in the 'processing' state when paper is still moving?
Suggested resolution: This is now moot, since Space-Current-Job is to be removed.
23. ISSUE: In the 'processing-stopped' state, is there a "current" job?
Suggested resolution: This is now moot, since Space-Current-Job is to be removed.
24. Should be Space-Job with job-id if in spec at all. Space-Current-Job is not to be in the spec at all.

The following numbered ISSUES were not addressed in the notes, so I've copied them here so that we have one set of issues and agreements:

25. ISSUE 2: In the Reset-Printer operation, is the "non-process-run-out" operation attribute really needed at all or can the default behavior for Reset-Printer be defined to be to perform non-process run out (for continuous and cut sheet printers)?

26. ISSUE 3: In the Restart-Printer operation, is the "non-process-run-out" operation attribute really needed at all or can the default behavior for Restart-Printer be defined to be to perform non-process run out (for continuous and cut sheet printers)?
27. ISSUE 4: In the Space-Printer operation, is the "non-process-run-out" operation attribute really needed at all or can the default behavior for Space-Printer be defined to be to perform non-process run out (for continuous and cut sheet printers)?
28. ISSUE 5: In the Shutdown-Printer operation, is the "non-process-run-out" operation attribute really needed at all or can the default behavior for Shutdown-Printer be defined to be to perform non-process run out (for continuous and cut sheet printers)?
29. ISSUE 7: On Shutdown-Printer with "when" = 'now', is the current job automatically restarted when the Printer is restarted? Or does some client have to issue a Restart-Job operation?
30. ISSUE 8: On Cancel-Current-Job, why isn't non-process-run-out automatic on a continuous form printer? When would an operator want to cancel the job and NOT run out the last sheets.? It would be simpler to require process-run-out when canceling the current job (for continuous and cut sheet printers).
31. ISSUE 10: For the Pause-Job operation, is the "non-process-run-out" operation attribute really needed at all or can the default behavior for Pause-Job be defined to be to perform non-process run out (for continuous and cut sheet printers)?