Title: Issues in IPP/1.0 & 1.1 Additional Administrative Operations, 7/19/1999
From: Tom Hastings
Date: 7/26/1999
File: ipp-ops-admin-issues-990726-th.doc

I've added Carl's and my suggested resolutions to all of these issues, preceded by CK> or TH>, respectively. The dates indicate a mail message on the subject.

Here are the collected issues that are embedded in the "IPP/1.0 & 1.1 Additional Administrative Operations, 7/19/1999, that was posted at:

ftp://ftp.pwg.org/pub/pwg/ipp/proposed-registrations/operations/ipp-ops-admin-990719.doc
ftp://ftp.pwg.org/pub/pwg/ipp/proposed-registrations/operations/ipp-ops-admin-990719.pdf
ftp://ftp.pwg.org/pub/pwg/ipp/proposed-registrations/operations/ipp-ops-admin-990719-rev.doc
ftp://ftp.pwg.org/pub/pwg/ipp/proposed-registrations/operations/ipp-ops-admin-990719-rev.pdf

ISSUE 1: Can a client determine the values of "when" that are supported for operations (Pause-Printer, Reset-Printer, Shutdown-Printer, and Pause-Current-Job)?

TH> Just add a "when-values-supported" that applies to all of the four operations supported, with the exception of Pause-Current-Job. For Pause-Current-Job, only the 'now' and 'after-current-copy' have any meaning, so the other two values ('after-current-job' and 'after-all') don't apply.

ISSUE 2: 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?

TH> Replace the "settable-attributes" Printer Description attribute, with three Printer Description attributes: "printer-settable-attributes", "job-settable-attributes", and "interpreter-settable-attributes". The latter for those implementations that have different values for Printer attributes in the Get-Printer-Attributes and Set-Printer-Attributes operations, depending on the value of the "document-format" operation attribute supplied by the client. If and when we get a Document object, then we can add a "document-settable-attributes" Printer Description attribute.

ISSUE 3: Ok that the "printer-controls-other-protocols" Printer Description attribute is just a boolean, or do we need a list of operations that affect non-IPP protocols?

TH> It would be more flexible to allow per-operation control, so change from "printer-controls-other-protocols" Printer Description attribute to "operations-affecting-other-protocols (1setOf type2 enum). The values are defined by the "operations-supported (1setOf type2 enum)" operation.

ISSUE 4: Is IPP intended for printer management. The issue is still undetermined?

TH> I think it is natural and so far there is not much overlap with any MIB. If and when we do get overlap, we need to make sure that the semantics are the same.

ISSUE 5: 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?

TH> Its sufficient for the client to perform the "all" function.

ISSUE 6: Ok to add the Interpreter object, even though it doesn’t solve all of the attribute constraint problems. At least it gives us a more object-based description of the semantics. When we do add some sort of Printer Description attribute that enumerates combinations of Job attribute values that may not be used in combination (like PPD file constraints), it can include the "document-format" attribute among them. So introducing the Interpreter object in no way precludes a complete constraint description mechanism in the future.

TH> Yes, its good modeling and doesn’t change the IPP/1.0 and IPP/1.1 syntax or semantics.

ISSUE 7: Why not have a separate out-of-band ’not-settable’ value, so that the client can distinguish between the cases of the attribute isn’t supported versus the attribute is supported, but is not settable? True, the client can query the "settable-attributes" to see which attributes can be set, before or after issuing the Set-Printer-Attributes operation.

TH> I think that providing a separate out-of-band code is useful, since a reply could contain both unsupported attributes and ones that were supported, but are not settable in this implementation.

ISSUE 8: 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)?

CK 7/19/99: Remove the concept of non-process-run-out from all operations, but add a new operation that performs non-process run-out. This operation will be useful for web printers. Call this operation: Run-Out-Printer?

ISSUE 9:   What state does 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’) which is the state that Shutdown-Printer leaves the Printer in. Then the operator issues Enable-Printer and Resume-Printer when want to restore normal operation. The client can automatically issues these addition 2 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’.

TH 7/26: Remove the Shutdown-Printer ’standby’ mode concept. Shutdown-Printer always powers off eventually. Also remove Restart-Printer operation all together. Instead change the Disable-Printer and Enable-Printer operations to Disable-Operations and Enable-Operations, so that individual operations are enabled and disabled independent of the state of the Printer and don't otherwise affect the state of the Printer.

ISSUE 10: 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)?

CK 7/19/99: Remove the concept of non-process-run-out from all operations, but add a new operation that performs non-process run-out. This operation will be useful for web printers. Call this operation: Run-Out-Printer?

ISSUE 11 - Need to look at life cycle of the Printer versus standby/power-down and the other operations that can be accepted. There can be appreciable time between acceptance of this operation and when the final state of the printer, either standby or powered down is reached. Is it ok for non-submission operations to continue to be accepted during this time? May need 'moving-to-shutdown'. What about 'moving-to-standby'?

TH> Add 'moving-to-shutdown' which the Shutdown-Printer sets immediately (analogous to 'moving-to-paused'). Then the 'shutdown' values means that the shutdown has completed (and is only meaningful to a server implementation that hosts the Printer object). Thus the server can still respond to a Get-Printer-Attributes operation after the Printer is shutdown as stated in IPP/1.1.

ISSUE 12: 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)?

CK 7/19/99: Remove the concept of non-process-run-out from all operations, but add a new operation that performs non-process run-out. This operation will be useful for web printers. Call this operation: Run-Out-Printer?

ISSUE 13: 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.

CK 7/19/99: Suggest that it means checkpoint the job to disk, such that processing can be resumed in the future. Get all the counters and whatever from the output device and save that information persistently so that the job might be resumed in the future.

ISSUE 14: Do we really need the "synchronize" operation attribute for the Shutdown-Printer operation or can synchronization be the default behavior of the Shutdown-Printer operation? It is not clear why this would ever be false? If it never makes sense to be 'false', can we remove it altogether?

CK 7/19/99: "synchronize" is set to false in exceptional conditions; like when a previous attempt to shutdown with "synchronize" 'true' hung (should never happen but apparently it does sometimes). This could be automated with some kind of timeout mechanism, but we chose to give the operator direct control.

So change the default behavior from 'false' to 'true' when the client omits the "synchronize" operation attribute.

ISSUE 15: Is the current job automatically restarted when the Printer is restarted? Or does some client have to issue a Restart-Job operation?

The question is moot, if we remove the Restart-Job operation, as suggested:

TH 7/26: Remove the Shutdown-Printer ’standby’ mode concept. Shutdown-Printer always powers off eventually. Also remove Restart-Printer operation all together. Instead change the Disable-Printer and Enable-Printer operations to Disable-Operations and Enable-Operations, so that individual operations are enabled and disabled independent of the state of the Printer and don’t otherwise affect the state of the Printer.

ISSUE 16: 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).

CK 7/19/99: Remove the concept of non-process-run-out from all operations, but add a new operation that performs non-process run-out. This operation will be useful for web printers. Call this operation: Run-Out-Printer?

ISSUE 17: Is the "non-process-run-out" operation attribute really needed at all or can the default behavior for Pause-Current-Job be defined to be to perform non-process run out (for continuous and cut sheet printers)?

CK 7/19/99: Remove the concept of non-process-run-out from all operations, but add a new operation that performs non-process run-out. This operation will be useful for web printers. Call this operation: Run-Out-Printer?

ISSUE 18: Do we really need the "synchronize" operation attribute or can synchronization be the default behavior of the Pause-Current-Job operation?

CK 7/19/99: "synchronize" is set to false in exceptional conditions; like when a previous attempt to shutdown with "synchronize" ’true’ hung (should never happen but apparently it does sometimes). This could be automated with some kind of timeout mechanism, but we chose to give the operator direct control.

So change the default behavior from ’false’ to ’true’ when the client omits the "synchronize" operation attribute.

ISSUE 19: Is the Space-Current-Job reasonable to perform when the current job is in the ’processing’ state when paper is still moving?

CK 7/20/99: We can do Space-Current-Job while the printer is stopped, paused, or running.