# IPP Printer versus Device object and operation semantics

From: Tom Hastings, Peter Zehler, Bob Herriot, Don Fullman, Mike Shepherd, Ira McDonald
Date: 12/03/99
File: ipp-printer-and-device-semantics-991201.doc

We had a good IPP telecon on Wednesday, November 17, 1999 on trying to disambiguate the semantics of the IPP Printer operations on the one hand that affect the software abstraction in each of its possible configurations and (new) Device operations that affect what [ipp-mod] calls the "output device" ("lump of metal") on the other hand. This telecon was building on the ideas suggested at the Raleigh IPP WG meeting October 28-29 that relates to the review of the Set2 and Set3 operations.

In reviewing the semantics of the operations, we suggest that there is not a need for a Device object, only Device operations. Not having a Device object significantly simplifies the semantics: there are no Device attributes and no Get-Device-Attributes and Set-Device-Attributes operations. If some Printer MIB attributes are needed, they can be added to the Printer object, but this should be done on a case by case basis with good justification.

This is a talking paper, not a complete spec. If we get agreement on this talking paper, we can produce the detailed spec very quickly.

## TABLE OF CONTENTS

# 1 Summary of the idea

In the IPP model, there are two distinct concepts: the Printer object and the output device. The output device is the “lump of metal” that makes marks on paper. The Printer object with its attributes represents a software abstraction. The IPP Printer object may be embedded in the output device or is hosted in a server. When the Printer object is hosted in a server, the server is connected by some means (serial, parallel, network-any protocol) to one or more output devices, each of which may or may not contain an IPP Printer object.

An Xxxx-Printer operation affects an IPP Printer object, i.e. the software abstraction, and an Xxxx-Device operation affects an output-device.

# 2 IPP and its effects on other protocols

The Printer operations (“xxx-Printer”) MUST affect only the IPP Printer object (Job operations are considered separately in section 11). They do not affect Printer abstractions for other protocols. The Device operations (“xxx-Device”) affect an output-device, and thus affect all protocols that the output-devices supports. This separation makes it clear which operations affect only the IPP Printer object and which affect all protocols.

Note: with this clear separation there is no need for an attribute, such as "printer-controls-other-protocols" to specify whether an “xxx-Printer” operations affects other protocols on an output-device.

# 3 Target for the Device operations

The URI supplied for any Device operation is the same as for a Printer operation, namely, the Printer object's URI. There is no need to introduce a Device-URI for Device operations. Each “xxx-Device” operation is OPTIONAL for a Printer object to support, whether the Printer object is (1) embedded in an output-device, (2) in a server that controls a single output-device, (3) in a server that controls a single subordinate Printer object, (4) in a server that controls many output-devices ("output-device fan-out"), or (5) in a server that controls many IPP Printer objects ("IPP Printer fan-out"). If a Printer object in any of these configurations supports an “xxx-Device” operation, it causes the “xxx-Device” operation to occur on all fanned-out output-devices at once.

In order to give an operator more control over or more explicit state information about each output device in a fan-out configuration, the implementation SHOULD have an associated *subordinate* Printer object for each such output device as agreed in Raleigh and described in the posted 11/16/99 Set2 specification. A

subordinate Printer object (of Printer A) MAY be supported on the output-device or on the server that hosts Printer object A. Therefore, the Device operations do NOT need a separate URI or "device-name" operation attribute to indicate which output device is intended. Consequently, the Device operations have the same target "printer-uri" as do the Printer operations.

### 3.1 Add "output-devices-supported (1setOf name(127))

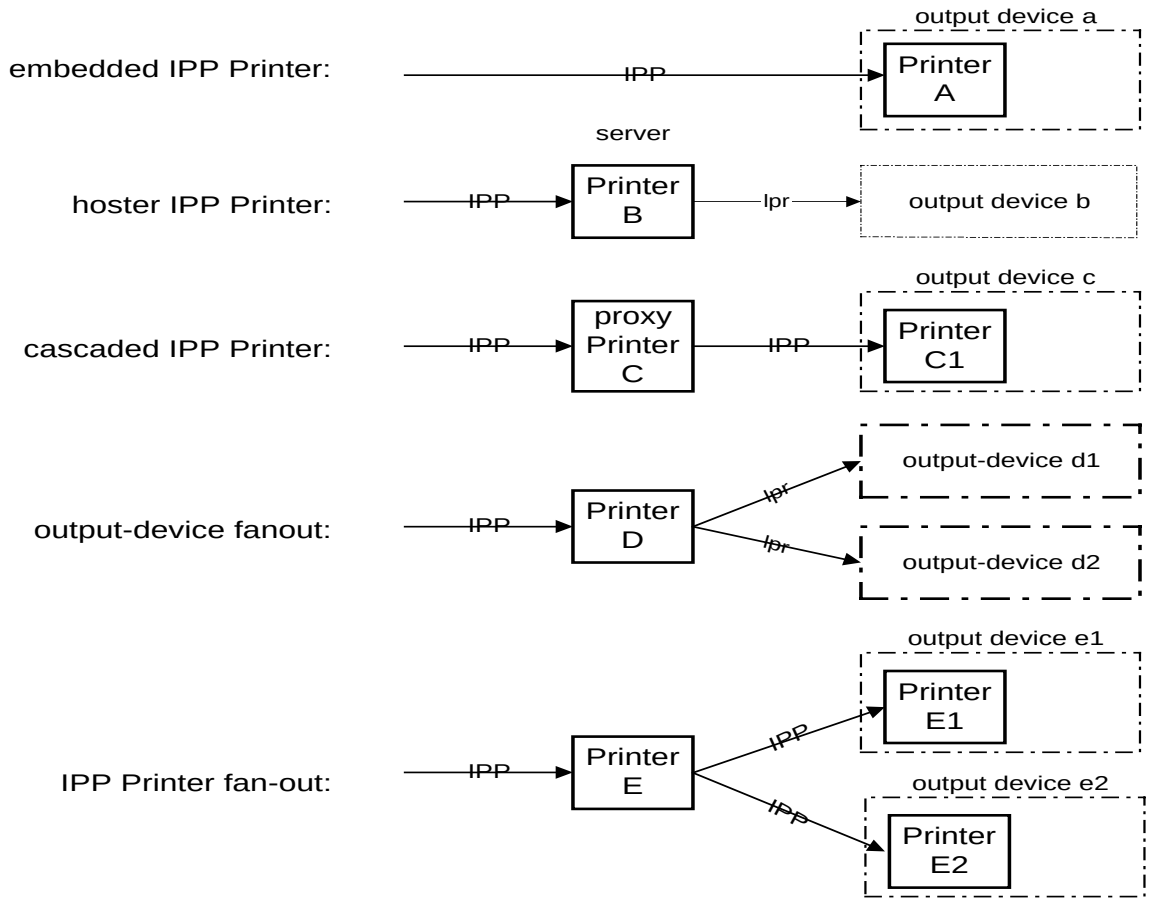
In order to know what output devices a Printer object supports, add a Printer Description attribute that identifies the names of the output devices that it supports. If a Printer object supports a Device operation as indicated in the Printer's "operations-supported" attribute, the Printer MUST forward that operation to all of the output-devices listed in its "output-devices-supported" attribute. This Printer Description attribute complements the IPP/1.0 "output-device-assigned" Job Description attribute (see [ipp-mod] section 4.3.13) that indicates to which output device the job was assigned for the output-device fan-out case.

The following sections discuss the clarifications on each Printer and Device operation.

## 4 Summary of the Printer/Device operations in various configurations

This section summarizes the effect of each Printer operation that has a corresponding Device operation. The Printer operations that have no corresponding Device operation are not shown in this section.

| Configuration Name | Description of configuration |
|---|---|
| embedded Printer object | Printer A: An embedded Printer object in output device a |
| hosted output device | Printer B: A server with one LPR output-device b |
| cascaded Printer object | Printer C: A server with one IPP Printer object C1 embedded in output-device c |
| output device fan-out | Printer D: A server with two LPR output-devices d1 and d2 |
| IPP Printer fan-out | Printer E: A server with two IPP Printer objects E1 and E2 embedded in output devices e1 and e2 |




82

| | **Printer A** | **Printer B** | **Printer C** | **Printer C1** | **Printer D** | **Printer E** | **Printer E1** |
|---|---|---|---|---|---|---|---|
| Pause-Printer | Pause A | Pause B | Pause C | Pause C1 | Pause D | Pause E | Pause E1 |
| Pause-Device | Pause a | Pause b | Pause c | Pause c | Pause d1&d2 | Pause e1&e2 | Pause e1 |
| Disable-Printer | Disable A | Disable B | Disable C | Disable C1 | Disable D | Disable E | Disable E1 |
| Disable-Device | Disable a | Disable b | Disable c | Disable c | Disable d1&d2 | Disable e1&d2 | Disable e1 |
| Shutdown-Printer | Shutdown A | Shutdown B | Shutdown C | Shutdown C1 | Shutdow n D | Shutdown E | Shutdown E1 |
| Shutdown-Device | Shutdown a | Shutdown b | Shutdown c | Shutdown c | Shutdow n d1&d2 | Shutdown e1&e2 | Shutdown e1 |
| Purge-Jobs (Printer) | Purge A, not a | Purge B, not b | Purge C not C1*, not c | Purge C1, not c | Purge D only | Purge E only | Purge E1, not e1 |
| Purge-Device | Purge a | Purge b | Purge c | Purge c | Purge d1&d2 | Purge e1&e2 | Purge e1 |

83 Note: Any Printer object MAY either support an operation or not support an operation, depending on
84 implementation and/or configuration. An implementation and/or a system administrator could configure a

Printer so that it doesn't support a Device operation because the Printer object is controlling more than one output-device. So the table entries are showing what is possible, not what MUST be possible.

# 5 Forwarding Printer and Device operations

A Printer object, whether it has IPP Printer fan-out or not, MUST NOT forward Printer operations to its subordinate Printer objects. Instead, the client MUST discover the subordinate Printer URIs and perform any operations on them directly.

A Printer object that has output device fan-out, such as Printer D, or IPP Printer fan-out, such as Printer E, either (1) MUST support a Device operation by performing it uniformly on all fan-out output devices or (2) MUST reject a Device operation as not supported (either because of fan-out or because the operation isn't supported). This choice is an IMPLEMENTATION OPTION and MAY be configurable. In other words, if a Device operation is supported by a Printer object, the Printer object MUST forward the Device operation so that it (eventually) gets to the actual output device(s).

There are some reliability problems involved with forwarding Device operations. When a Printer object forwards a Device operation to subordinate Printer object(s), the operation may encounter problems, such as a network failure. Therefore, the responses to Device operations do not indicate that the operation has actually reached any or all of the output device(s). In other words, Device operations when forwarding is involved, is on a "best efforts" basis and there is no way for a client to query to see if the Device operation actually completed on each of the output devices. Therefore, rather than relying on forwarding of Device operations, it is RECOMMENDED that each output device have its own Printer object in order to reflect the state of the output device completely and to provide the best control for each of the output devices to an operator.

# 6 Pause-Printer versus Pause-Device

## 6.1 Current definition

The [ipp-mod] has the following implementation choice for Pause-Printer (see section 3.2.7):

> This OPTIONAL operation allows a client to stop the Printer object from scheduling jobs on all its devices. Depending on implementation, the Pause-Printer operation MAY also stop the Printer from processing the current job or jobs. Any job that is currently being printed is either stopped as soon as the implementation permits or is completed, depending on implementation. The Printer object MUST still accept create operations to create new jobs, but MUST prevent any jobs from entering the 'processing' state.

## 6.2 Proposed definition

This proposal allows an implementation to support the first choice with the Pause-Printer operation and the second choice with the Pause-Device operation:

The Pause-Printer operation stops the IPP Printer object from sending additional IPP jobs to a subordinate Printer or to an output device. If the IPP Printer is implemented in a server, then a Pause-Printer operation keeps it from sending jobs to a subordinate IPP Printer which may be another server or a Printer in an output device. If the IPP Printer is implemented in an output device, then Pause-Printer keeps it from sending the job to the output device.) However, in any configuration the current job continues normally in the output device producing marked media. If the Printer object has already sent additional jobs to the output device, they also print. If the Printer object is in the middle of sending a job to the output device, it continues sending that job, but does not send any other jobs to the output device (otherwise a network time-out might occur that would abort the job). In other words, the Pause-Printer operation becomes more like the "when" attribute with a value of 'after-current-job'. However, we also propose to drop the "when" operation attribute for the Pause-Printer operation that is in the Set2 spec.

The Pause-Device operation stops the production of marked media *immediately*, i.e., in mid impression, after the current impression, or after the current sheet(s), depending on technology and implementation. It is like pushing the "pause" button locally on the device. The current job stops producing output, no matter whether it is an IPP job or a job submitted using some other job protocol, such as LPR/LPD.

The IPP Printer object that controls the output device will change the state of the Printer object's "printer-state" attribute and "printer-state-reasons" as specified in [ipp-mod]:

> The IPP Printer stops the current job(s) on its [output] device(s) that were in the 'processing' or 'processing-stopped' states as soon as the implementation permits. If the implementation will take appreciable time to stop, the IPP Printer adds the 'moving-to-paused' value to the Printer object's "printer-state-reasons" attribute (see section **Error! Reference source not found.**). When the device(s) have all stopped, the IPP Printer transitions the Printer object to the 'stopped' state, removes the 'moving-to-paused' value, if present, and adds the 'paused' value to the Printer object's "printer-state-reasons" attribute.

# 7 Disable-Printer versus Disable-Device

The Disable-Printer Printer operation disables only the IPP Printer object from accepting new jobs. The Disable-Device Device operation disables all job submission channels from accepting new jobs. All other operations are unaffected on each channel. There is no Disable-Device-Channel operation; only Disable all channels. Its too complicated to Disable individual job submission channels.

# 8 Purge-Jobs and Purge-Device

The Purge-Jobs operation is an IPP/1.1 Printer operation (see [ipp-mod]) that removes all IPP Job objects from the IPP Printer object, including the Job History. In order to perform this functionality the IPP Printer MAY have to cancel any jobs that it has sent to output device(s) by some means or other. For consistency, the IPP Printer does not forward the Purge-Jobs operation to its subordinate Printers.

The Purge-Device operation removes all jobs from the output device regardless of the job submission protocol used to submit the jobs.

# 9 Quiesce-Printer and Restart-Printer

Instead of having Shutdown-Printer have a way to restart the Printer, lets make a new operation that transitions the Printer object to a quiescent state, i.e., the Printer becomes a read-only object. The Printer stops accepting new jobs and stops sending jobs to the output device immediatelyHowever, the Printer completes any operations that were in process and completes sending any Jobs to the Printer that were in process. The Printer immediately stops accepting any operations, except the Printer MUST accept Restart-Printer and Get-Printer-Attributes and SHOULD accept Get-Job-Attributes and Get-Jobs. The current job is completed, so that this operation is a "graceful" stopping of the Printer. Since this may take some time, there are two "printer-state-reasons" values added: 'moving-to-quiescent' and 'quiescent'. There is no corresponding Quiesce-Device operation.

The Restart-Printer operation brings the Printer object back from the quiescent state (or 'moving-to-quiescent'). There is no corresponding Device operation.

# 10 Shutdown-Printer versus Shutdown-Device

The Shutdown-Printer operation only shuts down the IPP Printer object abstraction and has no direct effect on the output device and no effect on power anywhere. The Shutdown-Printer is a drastic operation. It MAY require more privileges than the Quiesce-Printer operation. Typically, the process(es) that are supporting IPP are exited after saving any necessary state. If the Printer object is processing, that is abandoned, since Shutdown-Printer happens immediately. In order to achieve a more orderly shutdown, the operator performs a Quiesce-Printer operation first and waits until the Printer adds the 'quiescent' value to the "printer-state-reasons" attribute.

There is no way to bring the IPP Printer back from the shutdown condition using the IPP protocol. However, the output device continues to do the work that it already has and any additional work that it might receive through other job submission channels.

On the other hand, the Shutdown-Device operation affects the output device, disables all the output device's channels, and powers the output device off. There is no way using the IPP protocol to power it back up. In order to shutdown both the output device and the Printer, the Shutdown-Device operation MUST be issued first, followed by the Shutdown-Printer, since the Printer cannot accept any operations after the Shutdown-Printer operation.

# 11 Extensions to the Job object

This section discusses the extension of semantics to the Job object.

## 11.1 Effect of IPP Operations on jobs from other protocols

The IPP Job operations MAY affect jobs submitted with other job submission protocols, depending on implementation. Thus a Get-Jobs or Get-Job-Attributes MAY return other jobs or it MAY NOT depending on implementation.

### 11.2 Add "device-submission-channel-type" Job Description attribute

In order to indicate which, we suggest adding a "device-submission-channel-type" (type2 keyword) Job Description attribute which indicates the type of channel the job was submitted on. The IPP keywords are derived by transforming the Printer MIB enum symbols registered for the Printer MIB job submission channel. The transformation is to remove the leading 'ch', add a hyphen in front of any uppercase letter or number that had a preceding lower case letter, and lower case all letters. So the Printer MIB chAppleTalkPAP enum symbol would become the IPP 'apple-talk-pap' keyword and the chIEEE1284Port enum symbol would become the IPP 'ieee-1284-port' keyword.

### 11.3 Unneeded Device operations on jobs

Here is the complete list of job operations to check to see if we all agree that they are unneeded as Device operations:

| Operations that affect jobs | Corresponding Device operation equivalent? |
|---|---|
| Create-Job | no |
| Print-Job | no |
| Print-URI | no |
| Validate-Job | no |
| Purge-Jobs | Purge-Device |
| Send-Document | no |
| Send-URI | no |
| Cancel-Job | no |
| Get-Job-Attributes | no |
| Get-Jobs | no |
| Hold-Job | no |
| Release-Job | no |
| Restart-Job | no |
| Set-Job-Attributes | no |
| Reprocess-Job | no |
| Cancel-Current-Job (though the target is the Printer object) | no |
| Pause-Current-Job (though the target is the Printer object) | no |
| Resume-Job | no |
| Promote-Job | no |

## 12 Get-Device-Attributes and Set-Device-Attributes

We suggest that we need neither a Get-Device-Attributes nor a Set-Device-Attributes operation, so we don't have a Device object either. If Printer MIB attributes are needed, they should be added to the Printer object. The problem with having both a Printer object and a Device object, is that it would be hard to remember whether an attribute was a Printer attribute or a Device attribute.

# 13 List of the Printer and Device operations

The list of Printer and Device operations are:

| Printer operation | Corresponding Device operation equivalent? |
|---|---|
| Pause-Printer | Pause-Device |
| Resume-Printer | Resume-Device |
| Purge-Jobs | Purge-Device |
| Get-Printer-Attribute | no |
| Set-Printer-Attributes | no |
| Disable-Printer | Disable-Device |
| Enable-Printer | Enable-Device |
| Quiesce-Printer | No why not Quiesce-Device? |
| Restart-Printer | not possible - the device is powered off |
| Shutdown-Printer | Shutdown-Device |