INTERNET-DRAFT - 11 ISSUES are numbered and highlighted like this
<draft-ietf-ipp-ops-set3-00.txt>

Carl Kugler
IBM Corporation
T. Hastings
Xerox Corporation
H. Lewis
IBM Corporation
~~October 22~~December 8, 1999

Internet Printing Protocol/1.1: Set3 Operations

Status of this Memo

Abstract

This document specifies 12 additional OPTIONAL operations for use with the Internet Printing Protocol/1.0 (IPP) [RFC2565, RFC2566] and IPP/1.1 [ipp-mod, ipp-pro]. These Set3 operations are Device operations that operators/administrators may perform that directly affect the output device:

Disable-Device	Enable-Device
Pause-Device-Now	Pause-Device-After-Current-Copy
Pause-Device-After-Current-Job	Resume-Device
Deactivate-Device	Activate-Device
Purge-Device	Reset-Device
Power-Off-Device	

This document does not define any new objects and does not define any Job operations. A companion specified, entitled "Internet Printing Protocol/1.1: Set2 Operations [ipp-set2] defined Printer operations that affect the Printer object, rather than the output device. Both the Set2 Printer operations and the Set3 Device operations have the Printer object as the target, i.e., the client must supply the "printer-uri" operation attribute and must direct the operation to the network entity that is implied by that URI.

The scope of IPP, is characterized in RFC2526 "Design Goals for an Internet Printing Protocol". It is not the intent of this document to revise or clarify this scope or conjecture as to the degree of industry adoption or trends related to IPP within printing systems. It is the intent of this document to extend the original set of operations - in a similar fashion to the Set1 extensions which referred to IPP/1.0 and were later incorporated into IPP/1.1.

This document is intended for registration following the registration procedures of IPP/1.0 [RFC2566] and IPP/1.1 [ipp-mod] and to be published as an RFC that extends IPP/1.0 and IPP/1.1. The material will also be added to a new minor revision of IPP if and when such a minor version is published.

The full set of IPP documents includes:

Design Goals for an Internet Printing Protocol [RFC2567]
Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
Internet Printing Protocol/1.1: Model and Semantics (this document)
Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]
Internet Printing Protocol/1.1: Implementer's Guide [IPP-IIG]
Mapping between LPD and IPP Protocols [RFC2569]

The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed printing functionality, and it enumerates real-life scenarios that help to clarify the features that need to be included in a printing protocol for the Internet. It identifies requirements for three types of users: end users, operators, and administrators. It calls out a subset of end user requirements that are satisfied in IPP/1.0. A few OPTIONAL operator operations have been added to IPP/1.1.

The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document describes IPP from a high level view, defines a roadmap for the various documents that form the suite of IPP specification documents, and gives background and rationale for the IETF working group's major decisions.

The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the abstract operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It defines the encoding rules for a new Internet MIME media type called "application/ipp". This document also defines the rules for transporting over HTTP a message body whose Content-Type is "application/ipp". This document defines a new scheme named 'ipp' for identifying IPP printers and jobs.

The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some of the considerations that may assist them in the design of their client and/or IPP object implementations. For example, a typical order of processing requests is given, including error checking. Motivation for some of the specification decisions is also included.

The "Mapping between LPD and IPP Protocols" document gives some advice to implementers of gateways between IPP and LPD (Line Printer Daemon) implementations.

Table of Contents

TABLE OF FIGURES

TABLE OF TABLES

1. Introduction

The Internet Printing Protocol (IPP) is an application level protocol that can be used for distributed printing using Internet tools and technologies. IPP version 1.1 (IPP/1.1) focuses on end user functionality with a few administrative operations included. This document defines additional OPTIONAL operator and administrator operations used to control Jobs and Printers. This document is a registration proposal for an extension to IPP/1.0 and IPP/1.1 following the registration procedures in those documents.

2. Terminology

This section defines terminology used throughout this document.

2.1 Conformance Terminology

Capitalized terms, such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED NOT, and OPTIONAL, have special meaning relating to conformance. These terms are defined in [ipp-mod] section 12.1 on conformance terminology, most of which is taken from RFC 2119 [RFC2119].

The following specialization of these terms apply to this document:

REQUIRED: if an implementation supports the extensions described in this document, it MUST support a REQUIRED feature.

OPTIONAL: if an implementation supports the extensions described in this document, it MAY support an OPTIONAL feature.

2.2 Other terminology (copied from Set2)

This document uses terms such as "attributes", "keywords", and "support". These terms have special meaning and are defined in the model terminology [ipp-mod] section 12.2.

IPP Printer object (or Printer for short) - a software abstraction defined by [ipp-mod].

~~O~~uutput ~~D~~ device - the physical imaging mechanism that an IPP Printer controls.

~~O~~output ~~D~~ device fan-out - a configuration in which an IPP Printer controls more that one output ~~-~~ device.

Printer fan-out - a configuration in which an IPP Printer object controls more than one subordinate IPP Printer object.

output device fan-in - a configuration in which an output device is controlled by more than one IPP Printer object.

Printer fan-in - a configuration in which an IPP Printer object is controlled by more than one IPP Printer object.

Subordinate Printer - an IPP Printer object that is controlled by another IPP Printer object. Such a Subordinate Printer object MAY have one or more Subordinate Printers.

Leaf Printer - a Subordinate Printer object that has no Subordinate Printer objects.

Non-Leaf Printer - an IPP Printer object that has one or more Subordinate Printer objects.

Chained Printer - a Non-Leaf Printer object that has exactly one Subordinate Printer object.

Job Creation operations - IPP operations that create a Job object: Print-Job, Print-URI, and Create-Job.

Embedded Printer - a Printer object that is implemented as part of the output device and shares the same power supply.

Hosted Printer - a Printer object that is implemented as part of some host that is separate from the output device, or at least as a separate power supply, and uses some connection mechanism, such as a serial port, a parallel port, or a network connection to communicate with the output device.

3. Requirements and Use Cases (copied from Set2)

The following requirements and usage cover both the Set2 [ipp-set2] and Set3 [~~ipp-set3~~this document] operations. They are presented here together to show the parallelism.

1. Have separate operations for affecting the IPP Printer versus affecting the output ~~-~~device, so its clear what the intent of each is and implementers can implement one or the other or both.

2. Support fan-out of Printer objects.

3. Support fan-out of output ~~-~~devices.

4. Support fan-in of Printer objects, as long as it doesn't make the semantics more complicated when not supporting fan-in.

5. Support fan-in of output objects, as long as it doesn't make the semantics more complicated when not supporting fan-in.

6. Instead of having operation attributes that alter the behavior of the operation significantly, have separate operations, so that it is simple and clear to a client which semantics the Printer is supporting (by querying the "operations-supported" attribute) and it is simple to describe the capabilities of a Printer implementation in written documentation (just list the OPTIONAL operations supported).

7. Need a Printer operation to prevent a Printer object from accepting new IPP jobs, but currently accepted jobs continue unaffected to be scheduled and processed. Need a companion one to restore the Printer object to accept new IPP jobs.

   Usage: Operator is preparing to take the IPP Printer out of service or to change the configuration of the IPP Printer.

   Suggested name and operations: **Disable-Printer** and **Enable-Printer**

8. Need a Device operation to prevent an output device from accepting any new jobs from any job submission protocol and a companion one to restore the output device to accepting any jobs.

Usage: Operator is preparing to take the output device out of service.

Suggested name and operations: **Disable-Device** and **Enable Device**

9. Need a Printer operation to stop the processing after the current IPP job completes and not start processing any additional IPP jobs (either by scheduling the jobs or sending them to the output device), but continue to accept new IPP jobs. Need a companion operation to start processing/sending IPP jobs again.

   Usage: Operator wants to gracefully stop the IPP Printer at~~s~~ the next job boundary. The~~i~~s Pause-Printer-After-Current-Job operation is also invoked implicitly by the Deactivate-Printer and the Shutdown-Printer operations.

   Suggested name and operations: **Pause-Printer-After-Current-Job**, **Resume-Printer**

10. Need a Device operation to stop the processing the current job "immediately", no matter what protocol. Its like the Pause button on the output device. This operation is for emergencies. The stop point depends on implementation, but can be mid page, end of page, end of sheet, or after a few sheets for output devices that can't stop that quickly. The paper path isn't run out. Need a companion operation to start processing the current any-protocol job without losing any thing.

    Usage: Operator sees something bad about to happen, such as the paper is about to jam, or the toner is running out, or the device is overheating or wants to add more paper.

    Suggested name and operations: **Pause-Device-Now**, **Resume-Device**

11. Need a Printer operation to stop the processing of IPP jobs after all of the currently accepted jobs that have been processed, but any newly accepted jobs go into the 'processing-held' state.

    Usage: This allows an operator to reconfigure the output device in order to let jobs that are held waiting for resources, such as special media, to get a chance. Then the operator uses Resume-Printer after reconfiguring. He repeats the two operations to restore the output device to its normal media.

    Suggested name and operations: **Pause-Device-After-All-Current-Jobs**, **Resume-Device**

12. Need a Device operation to stop the processing the current any-protocol job at a convenient point, such as after the current copy (or end of job if last or only copy). Need a companion operation to start processing the current any-protocol job or next job without losing any thing.

    Usage: The operator wants to empty the output bin that is near full. The paper path is run out.

    Suggested name and operations: **Pause-Device-After-Current-Copy**, **Resume-Devic**e

13. Need a Device operation that always pauses on a job boundary, no matter how many copies, in order to not break up a job. Need a companion operation to start processing the current any-protocol job or next job without losing any thing.

Usage: The operator wants to empty the output bin that is near full, but he doesn't want to break up a job in case it has multiple copies. The paper path is run out.

Suggested name and operations: **Pause-Device-After-Current-Job**, **Resume-Device**

14. Need a Printer operation that combines Disable-Printer, Pause-Printer-After-Current-Job, and rejects all other Job, Printer, and Device operations, except Job and Printer queries, System Administrator Set-Printer-Attributes, and the companion operation to resume activity. In other words, this operation makes the Printer a read-only object in a graceful manner for end-users and the operator.

    Usage: The administrator wants to reconfigure the Printer object using the Set-Printer-Attributes operation without disturbing the current in process work, but wants to make sure that the operator isn't also trying to change the Printer object as part of running the Printer.

    Suggested name and operation: **Deactivate-Printer, Activate-Printe**r

15. Need a Device operation that combines Disable-Device, Pause-Device-After-Current-Job, and rejects all other Device operations, except Job and Printer queries and the companion operation to resume activity. In other words, this operation makes the output device a read-only object in a graceful manner.

    Usage: The field service person wants to open up the device without disturbing the current in process work, perhaps to replace staples, or replace the toner cartridge.

    Suggested name and operation: **Deactivate-Device, Activate-Device**

16. Need a Printer operation to recover from the IPP Printer software that has gotten confused (run out of heap memory or gotten into a state that it doesn't seem to be able to get out of). This is a condition that shouldn't happen, but does in real life. Any volatile information is saved if possible before the software is re-initialized. No companion operation is needed to undo this. We don't want to go back to the "confused" state :-).

    Usage: The IPP Printer software has gotten confused or isn't responding properly.

    Suggested name and operation: **Restart-Printer**

17. Need a Device operation to recover from the output device hardware and software that has gotten confused (gotten into a state that it doesn't seem to be able to get out of, run out of heap memory, etc.). This is a condition that shouldn't happen, but does in real life. Any volatile information is saved if possible before the software and hardware is re-initialized. This is the same and has the same options as the Printer MIB reset. No companion operation is needed to undo this. We don't want to go back to the "confused" state :-).

    Usage: The output device has gotten confused or need resetting to some initial conditions.

    Suggested name and operation: **Reset-Device**

18. Need a Printer operation to put the IPP Printer object out of business with no way in the protocol to bring that instantiation back to life. (but see Startup-Printer which brings up exactly one new instantiation to life with the same URL).

    Usage: The Printer is being moved or the building's power is being shut off.

    Suggested name and operation: **Shutdown-Printer**

19. Need a Printer operation to bring an IPP Printer to life when there is an already running host. Note: This operation is unlikely to be supported for the embedded Printer configuration.

    Usage: After the host is started (by means outside the IPP protocol), the operator is able to ask the host to bring up any number of Printer objects (that the host has been configured in some way) each with distinct URLs.

    Suggested name and operation: **Startup-Printer**

20. Need a Device operation to power off the output device after writing out any software state. It is assumed that other operations have more gracefully prepared the output device for this drastic and immediate. There is no companion Device operation to bring the power back on.

    Usage: The output device is going to be moved, the power in the building is going to be shutoff, the repair man has arrived and needs to take the output device apart.

    Suggested name and operation: **Power-Off-Device**

3.1 List of the Printer and Device operations

The list of Printer and the corresponding Device operations is shown in Table 1:

Table 1 - List of Printer operations and corresponding Device operations

Printer operation (see [ipp-set2])	Corresponding Device operation equivalent
Get-Printer-Attribute	no
Set-Printer-Attributes	no
Disable-Printer	Disable-Device
Enable-Printer	Enable-Device
Pause-Printer-After-Current-Job~~no~~	Pause-Device-Now
Pause-Printer-After-Current-Job~~no~~	Pause-Device-After-Current-Copy
Pause-Printer-After-Current-Job (= IPP/1.1 Pause-Job??)	Pause-Device-After-Current-Job
Pause-Printer-After-All-Current-Jobs	no
Resume-Printer (IPP/1.1 - [ipp-mod])	Resume-Device
Deactivate-Printer	Deactivate-Device
Activate-Printer	Activate-Device
Purge-Jobs (IPP/1.1 - [ipp-mod])	Purge-Device
Restart-Printer	Reset-Device
Shutdown-Printer	Power-Off-Device
Startup-Printer	no

When a Printer object receives a Device operation, it performs the corresponding Printer operation as shown in Table 1 and simultaneously controls the output device, so that the effect of the Device operation also happens to the IPP Jobs and the IPP Printer object, thereby keeping the IPP semantics correctly representing the state of the output device.

ISSUE 01 - Ok that every Device operation REQUIRES the IPP Printer to perform the corresponding Printer operation, if implemented?

ISSUE 02 - Which corresponding Printer operations MUST an implementation support, if it supports a particular Device operation?

4. Relationship between Printer objects and the output device

From [ipp-mod] section 2.1, we have:

> The term "IPP Printer" is a network entity that accepts IPP operation requests and returns IPP operation responses. As such, an IPP object MAY be:
>
> 1. an (embedded) device component that accepts IPP requests and controls the device or

2. a component of a print server that accepts IPP requests (where the print server controls one or more networked devices using IPP or other protocols).

The [ipp-set2] specification generalizes the IPP Printer object to be a parent and/or a subordinate Printer object to represent both IPP Printer fan-out and IPP Printer fan-in. IPP Printer fan-out is where a parent IPP Printer object has one or more subordinate Printer objects. IPP Printer fan-in is where a subordinate Printer object has two or more parent Printer objects.

4.1 The relationship between the Printer object and the output device

This Set3 document adds the following constraints to the definition of Printer objects relationships to output devices (see section 2.2 entitled "Other terminology (copied from Set2)":

A Leaf Printer object MUST "directly control" one or more (output device fan-out) output devices.

A Non-Leaf Printer object MUST NOT "directly control" any output devices. However, Non-Leaf Printer objects MAY "represent" output devices that its Subordinate Printer objects control.

An output device MUST have one or more (output device fan-in) Printer objects that "control" it.

Several Leaf Printer objects MAY "control" a single output device (output device fan-in).

Several Leaf Printer objects MAY "control" several output device objects (combination of output device fan-in and output device fan-out)

4.1.1 The output device fan-out configuration

IPP/1.0 [rfc2566] and IPP/1.1 [ipp-mod] define the output device fan-out as a configuration in which one Printer object represents more than one output device. Such a configuration is useful in order to provide load balancing between several output devices. Uses submit jobs to the IPP Printer and that Printer selects the least busy output device. If the output devices have differing capabilities, then the selection of which output device is more complicated. The Printer's "xxx-supported" attributes reflect the union of the output devices. If one or more identical output devices are a super set of the capabilities of the remaining output devices, then the Printer's "xxx-supported" attributes represent that superset. However, if the some output devices possess capabilities that others don't and those others possess capabilities that the first don't, the Printer's "xxx-capabilities" will include capabilities that cannot be utilized by a single job. It is RECOMMENDED to avoid such a configuration.

4.1.2 The output device fan-in configuration

The Set2 document [ipp-set2] introduces the Printer fan-in configuration. This document introduces the analogous configuration for the output device, namely, output device fan-in. While not explicitly provided in IPP/1.1 [ipp-mod], output device fan-in is not precluded by the IPP/1.1 semantics. Output device fan-in is where an output device is represented by more than one Printer object. Such a configuration can be used to offer different classes of service for a single output device, including differing capabilities and/or defaults for each Printer object with possibly differing access control that represent the single output device.

When an output device is represented by more than one Printer object, the Printer objects MAY be all hosted within a single server (see Figure 7 - single-server output device fan in) or each Printer object MAY be hosted in separate servers (see Figure 8 - multiple-server output device fan in). In either case, there are duplicate Printer objects attempting to represent the same output device.

4.1.3 Figures to show all possible configurations

Figure 1, Figure 2, and Figure 3 are taken from [ipp-mod] to show the configurations possible with IPP/1.0 and IPP/1.1 where all Printer objects are Leaf Printer objects. The remaining figures show additional configurations that this document defines using non-Leaf and Leaf Printer objects. Legend for all figures:

```
----> indicates a network protocol with the direction of its requests

##### indicates a Printer object which is either:
        - embedded in an output device or
        - hosted in a server.  The Printer object
      might or might not be capable of queuing/spooling.

any   indicates any network protocol or direct
      connect, including IPP
```

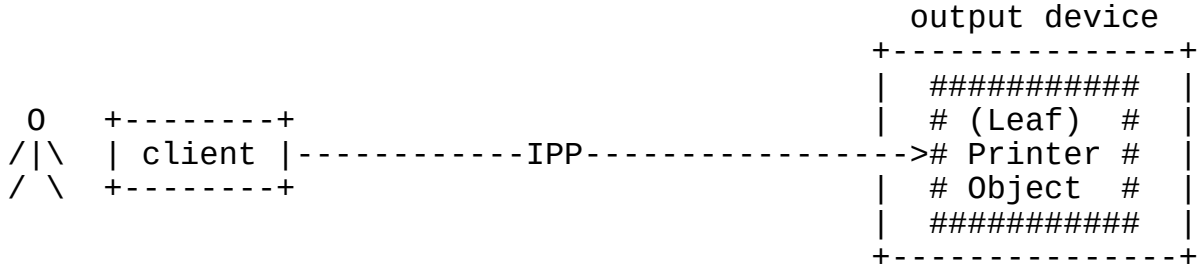


Figure 1 - embedded Printer object

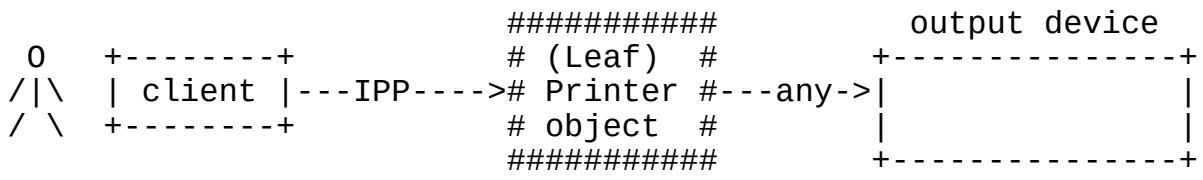


Figure 2 - hosted Printer object

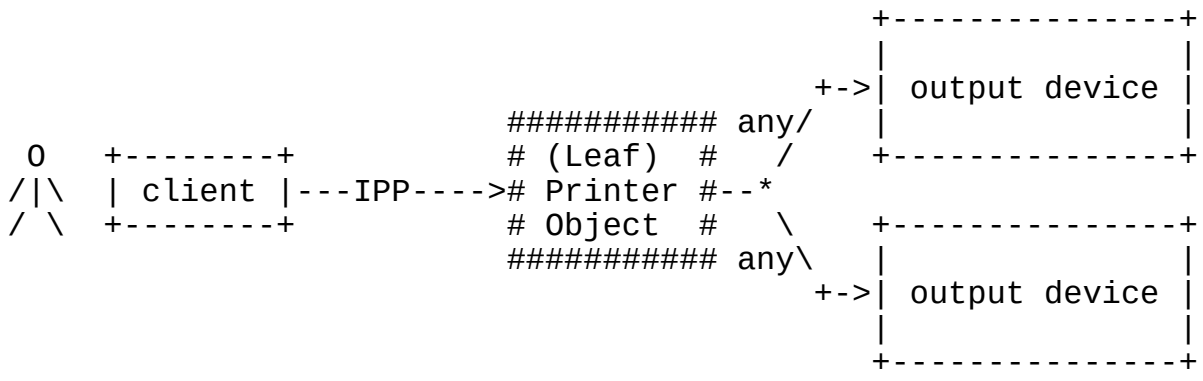
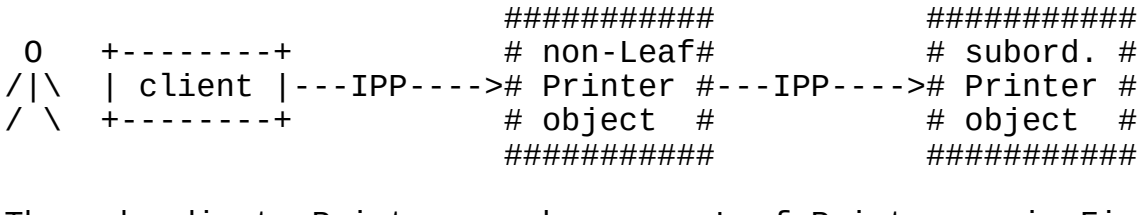
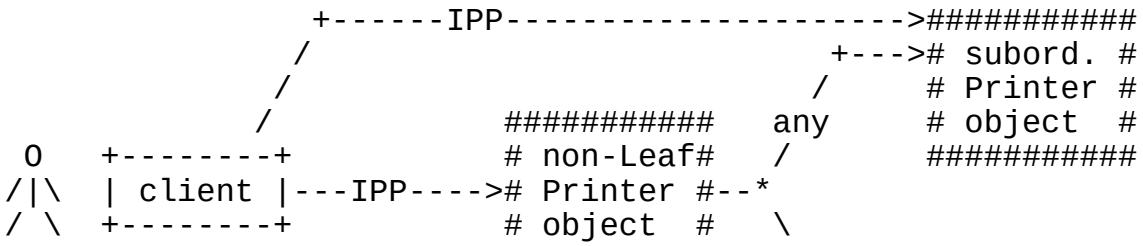


Figure 3 - output device fan out



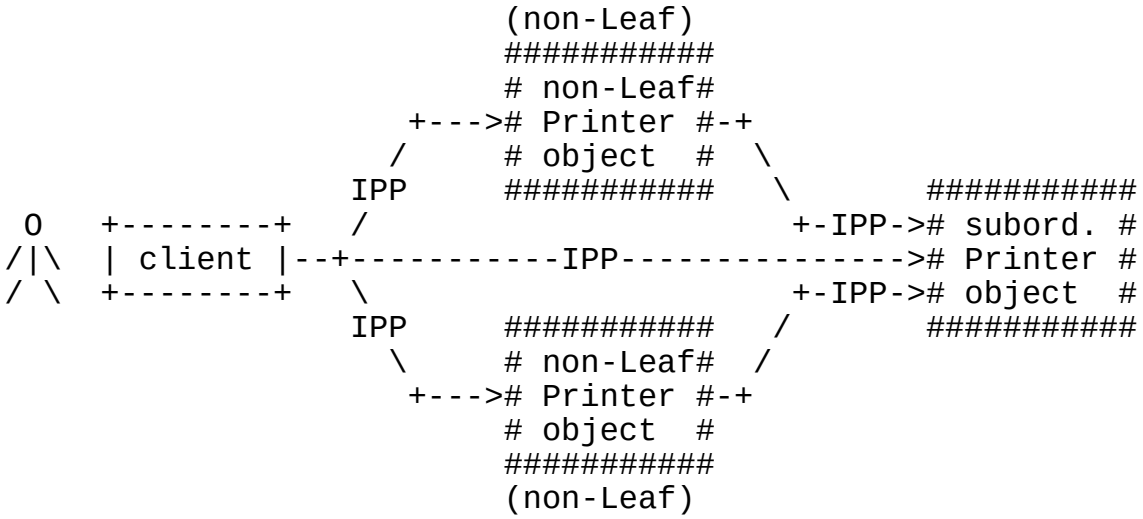
The subordinate Printer can be a non-Leaf Printer as in Figure 4 to Figure 6, or can be a Leaf Printer as in Figure 1 to Figure 3.

Figure 4 - Chained IPP Printer



The subordinate Printer can be a non-Leaf Printer as in Figure 4 to Figure 6, or can be a Leaf Printer as in Figure 1 to Figure 3.

Figure 5 - IPP Printer fan out



The subordinate Printer can be a non-Leaf Printer as in Figure 4 to~~,~~ ~~Figure 5, or~~ Figure 6, or can be a Leaf Printer as in Figure 1 to~~,~~ ~~Figure 2, or~~ Figure 3.

Figure 6 - IPP Printer fan in

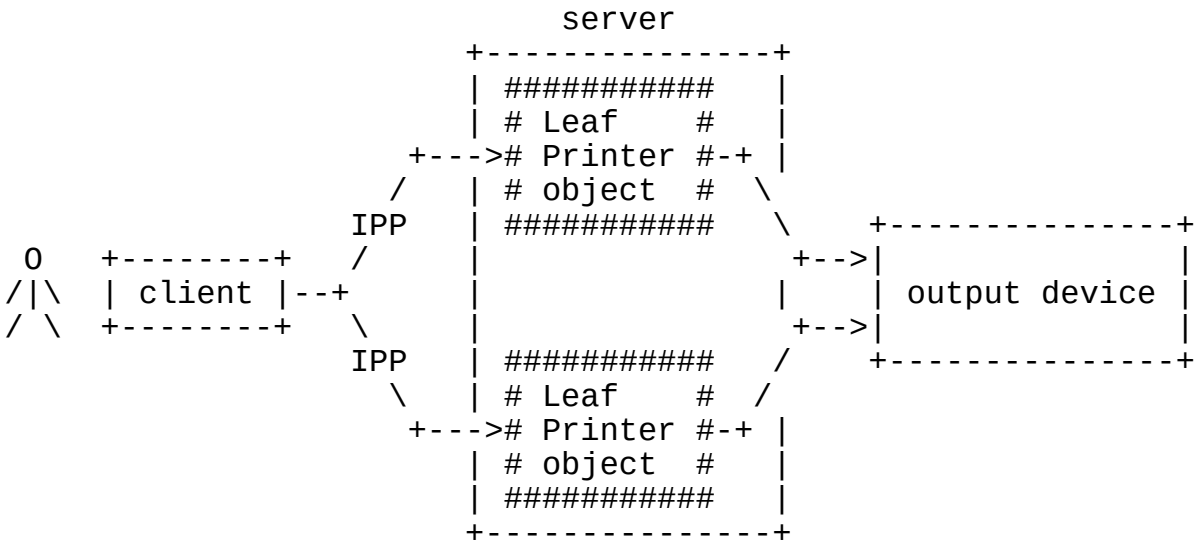


Figure 7 - single-server output device fan in

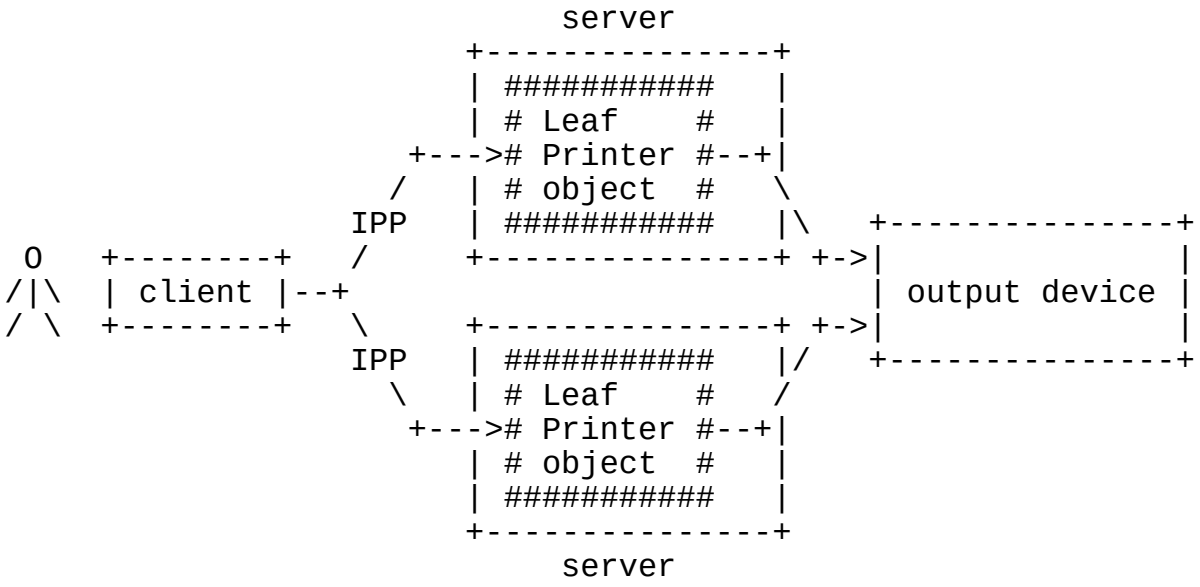


Figure 8 - multiple-server output device fan in

Note: the addition of subordinate Printer objects and output device fan-in is completely compatible with IPP/1.0 and IPP/1.1. The protocol and semantics are the same between a client and a (parent) Printer object for all configurations.

4.2 Summary of the relationship between the Printer object and the output device

This section summarizes the relationships between the Printer object and the output device:

An output device "is represented by" one (Figure 1, Figure 2, and Figure 3) or more (Figure 7 and Figure 8) Leaf Printer objects.

A Printer object is either a Leaf Printer or a non-Leaf Printer, but not both:

A Leaf Printer object "represents" one (Figure 1, Figure 2, Figure 7, and Figure 8) or more (Figure 3) physical output devices.

A non-Leaf Printer object "supports" one (Figure 4 and Figure 6) or more (Figure 5) subordinate Printer objects.

4.3 Forwarding requests

This section describes the forwarding of Device operations.

4.3.1 Forwarding requests that affect output devices

The [ipp-set2] contains the following restrictions about forwarding Printer operations to subordinate Printer objects:

When there is Printer fan-out, Printer fan-in, and Chained Printers, the non-Leaf IPP Printer object MUST NOT forward the Printer operations that affect Printer objects to its subordinate Printer objects. If a client wants to explicitly target a subordinate Printer, the client MUST specify the URI of the subordinate Printer. The client can determine the URI of any subordinate Printers by querying the Printer's "subordinate-printers-supported (1setOf uri) attribute (see [ipp-set2] section 6.6).

There are similar, though not identical, conformance requirements and restrictions about forwarding Device operations:

1. If a Printer object supports a Device operation and is controlling a single output device (Figure 1, Figure 2, Figure 7, and Figure 8) or a single subordinate Printer object (Figure 4 and Figure 6), the Printer object MUST forward the Device operation to that single output device or Printer object, respectively. Note: This rule differs from the rule in [ipp-set2] for Printer operations, since Printer operations MUST NOT be forwarded to subordinate Printer objects for all configurations. This exception is made for Device operations so that there is no difference in the Device operation semantics as seen by an operator or administrator client whether the first Printer object is (1) using IPP (to control the single immediate downstream Printer object) versus (2) using some other protocol (to control the single immediate downstream output device).

2. A Printer object MUST NOT support (and MUST NOT forward) a Device operation when the (Leaf) Printer object is controlling more than one output device (device fan-out - Figure 3) or the (non-Leaf) Printer object is controlling more than one immediate subordinate IPP Printer object (Printer object fan-out - Figure 5). Otherwise, the "printer-state" and "printer-state-reasons" become too complicated to represent the collective states of several output printers. Also if some of the

forwarded Device operations were to succeed and others fail, the resulting state is too hard to represent. If a client wants to explicitly target a subordinate Printer, the client MUST specify the URI of the subordinate Printer. The client can determine the URI of any subordinate Printers by querying the Printer's "subordinate-printers-supported (1setOf uri) attribute (see [ipp-set2 section 6.6).

Table 2 lists the Device operations and the forwarding behavior that a Leaf Printer to its output device(s) and a non-Leaf Printer MUST exhibit to its immediate subordinate Printer object(s).

Table 2 - Forwarding Device operations

Device operation	1 embedded (no fan-out) (Leaf)	2 hosted (no fan-out) (Leaf)	3 output device fan-out (Leaf)	4 Chained Printer (non-Leaf)	5 Printer fan-out (non-Leaf)	6 Printer fan-in (non-Leaf)	7&8 output device-fan-in (Leaf)
Disable-Device	forward	forward	no	forward	no	forward	forward
Enable-Device	forward	forward	no	forward	no	forward	forward
Pause-Device-Now	forward	forward	no	forward	no	forward	forward
Pause-Device-After-Current-Copy	forward	forward	no	forward	no	forward	forward
Pause-Device-After-Current-Job	forward	forward	no	forward	no	forward	forward
Resume-Device	forward	forward	no	forward	no	forward	forward
Deactivate-Device	forward	forward	no	forward	no	forward	forward
Activate-Device	forward	forward	no	forward	no	forward	forward
Purge-Device	forward	forward	forward*	forward	forward*	forward	forward
Reset-Device	forward	forward	no	forward	no	forward	forward
Power-Off-Device	forward	forward	no	forward	no	forward	forward

* An exception is made for Purge-Device, since its purpose is to affect jobs, not the output device itself. Therefore, Purge-Jobs is always forwarded, just like all operations that directly affect jobs (see [ipp-set2]).

5. New Printer Description Attributes

The following new Printer Description attributes are defined for use in this extension.

5.1 output-devices-supported (1setOf name(127))

~~ISSUE 10 - For consistency with [ipp-mod], shouldn't this be singular even though it is multi-valued, i.e., device-name-supported (1setOf name(127))?~~

This OPTIONAL Printer attribute contains the user-friendly device name or device names which this Printer object is "representing". If this Printer object is a Leaf Printer object, then the Printer object MUST control the output device(s) so named. If this Printer object is a non-Leaf Printer, then the values in this attribute MUST be the union of the values of the "output-devices-supported" attributes of its immediate subordinate Printer objects. Therefore an end user client querying this attribute of this Printer object will discover all possible (down steam) output devices on which a job could be assigned if submitted to this Printer object.

ISSUE 03 - Ok to REQUIRE roll-up of the "output-devices-supported" Printer Description attribute.

An Administrator determines device names and configures this attribute to contain those device names via IPP Set-Printer-Attributes operation (see [ipp-set2]) or by some means outside the scope of this document. The precise format of these device names is implementation dependent and MAY depend on the protocol stack and the directory namespace.

Note: This attribute enhances the usefulness of the IPP/1.1 Job object attribute "output-device-assigned" (see [ipp-mod] section 4.3.13). The "output-device-assigned" Job attribute identifies the user-friendly output device to which the Printer object has assigned a job, for example, when a single Printer object is supporting multiple devices.

6. Additional values for the "printer-state-reasons" Printer Description attribute

This section defines additional values for the "printer-state-reasons" Printer Description attribute.

6.1 'device-deactivated'

'device-deactivated': Someone has issued a Deactivate-Device operation for the Printer object (see section 9.3.1) and the output device is in the process of becoming deactivated or has become deactivated. The Printer MUST reject all requests except: Activate-Device, queries (Get-Printer-Attributes, Get-Job-Attributes, Get-Jobs, etc.), Send-Document, and Send-URI (so that partial job submission can be completed - see section 9.3.1) and return the 'server-error-service-unavailable' status code.

ISSUE 04 - What additional 'device-moving-to-xxx' are needed as "printer-state-reasons" values? What target 'device-xxx' delayed states are needed as "printer-state-reasons" values?

7. New status codes

This section defines new status codes used by the operations defined in this document.

ISSUE 05 - What new status codes are needed, if any?

8. New out-of-band values

This section defines additional out-of-band values that can be used with any attribute in principle. See the beginning of [ipp-mod] section 4.1.

ISSUE 06 - What new out-of-band values are needed, if any?

9. Definition of the Set 3 Device operations

All Device operations are directed at Printer objects. A client MUST always supply the "printer-uri" operation attribute in order to identify the correct target of the operation. These descriptions assume all of the common semantics of IPP/1.1 Model and Semantics document [ipp-mod] section 3.1.

The Set 3 Device operations are summarized in the following table:

Table 3 - Device operation Operation-Id assignments

Operation Name	Operation-Id	Brief description
Disable-Device	0x??	Prevents the output device from accepting jobs with any job submission protocol.
Enable-Device	0x??	Allows the output device to accept jobs from any job submission protocol.
Pause-Device-Now	0x??	Stops the output device from marking media as soon as possible on the page or sheet.
Pause-Device-After-Current-Copy	0x??	Stops the output device from marking media after the current copy has been stacked.
Pause-Device-After-Current-Job	0x??	Stops the output device from marking media after the current job has been stacked.
Resume-Device	0x??	Continues the output device from the last Pause Device operation.
Deactivate-Device	0x??	Puts the output device into a read-only deactivated state.
Activate-Device	0x??	Restores the output device to normal activity.
Purge-Device	0x??	Removes all traces of jobs in the output device.
Reset-Device	0x??	Resets the hardware state of the output device and re-initializes the output device software.
Power-Off-Device	0x??	Powers off the output device

All of the operations in this document are OPTIONAL for an IPP object to support. Unless the specification of an OPTIONAL operation requires support of another OPTIONAL operation, conforming

implementations may support any combination of these operations. Many of the operations come in pairs and so both are REQUIRED if either one is implemented.

9.1 The Disable and Enable Device Operations

This section defines the OPTIONAL Disable-Device and Enable-Device operations that stop and start the output device from accepting new jobs and, therefore, the IPP Printer from accepting IPP Jobs. If either of these operations are supported, both MUST be supported.

These operations allow the operator to control whether or not the output device (and the IPP Printer object) will accept new jobs using any of its supported job submission protocols. These operations have no other effect on any of the other operations of the output device, so that the output device continues to accept all other operations and continues to schedule and process jobs normally that it has already received. In other words, these operation control the "input of new jobs" to the output while the Pause and Resume Device operations (see section 9.2) independently control the "output of new jobs" from the output device to the output media.

Note: Contrast the Disable Device operations which affect all job submission protocols that the output device supports and the Disable Printer operations (see [ipp-set2]) which affect only the IPP Job Creation operations to the Printer object. In other words, the Disable Device operations have the same effect on all job submission protocols that the Disable Printer operations have on the IPP job submission protocol.

9.1.1 Disable-Device Operation

This OPTIONAL operation allows a client to stop the output device from accepting new jobs, i.e., cause the output device to reject subsequent operations to create new jobs using any job submission protocol. The Printer object performs a Disable-Printer operation (see [ipp-set2]) (which sets the Printer's "printer-is-accepting-jobs" READ-ONLY Printer Description attribute to 'false') plus controls the output device to stop accepting new jobs with any of the output device's job submission protocols. The output device still accepts all other operations. All previously created or submitted jobs and currently processing jobs continue unaffected on the output device.

The IPP Printer MUST accept the request in any state of the IPP Printer or the output device. This operation has no immediate or direct effect on the Printer's "printer-state" and "printer-state-reasons" attributes.

Access Rights: The authenticated user (see [ipp-mod] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5).

The Disable-Device request and response have the same attribute groups and attributes as the Pause-Device operation (see [ipp-mod] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see [ipp-set2] section 5.1).

9.1.2 Enable-Device Operation

This OPTIONAL operation allows a client to start the output device accepting new jobs, i.e., cause the output device to accept subsequent operations to create new jobs using any job submission protocol. The Printer object performs an Enable-Printer operation (see [ipp-set2]) (which sets the Printer's "printer-is-accepting-jobs" READ-ONLY Printer Description attribute to 'true') plus controls the output device to start accepting new jobs with any of the output device's job submission protocols.

The IPP Printer MUST accept the request in any state. This operation has no immediate or direction effect on the Printer's "printer-state" and "printer-state-reasons" attributes.

Access Rights: The authenticated user (see [ipp-mod] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5).

The Enable-Device request and response have the same attribute groups and attributes as the Pause-Device operation (see [ipp-mod] sections 3.2.8.1 and 3.2.8.2), including the new "printer-message-from-operator" operation attribute (see [ipp-set2] section 5.1).

9.2 The Pause and Resume Device operations

This section defines the OPTIONAL Pause-Device-Now, Pause-Device-After-Current-Copy, Pause-Device-After-Current-Job, and Resume-Device operations. These operations affect the scheduling of jobs from any job submission protocol on the output device. The Pause-Device-Now and Pause-Device-After-Current-Job operation are possible implementation options of the OPTIONAL IPP/1.1 Pause-Printer (see [ipp-mod] sections 3.2.7 and Table 4 below). If any of the Pause Device operations are supported, then the Resume-Device operation MUST be supported.

These operations allow the operator to control the current job's marking of media by the output device. These operations have no other effect on the output device, so that the output device continues to accept all operations. In other words, these operation control the "output of" the output device(s) while the Disable and Enable Printer operations (see section 9.1) independently control the "input of new jobs" to the IPP Printer.

Note: Contrast the Pause Device operations which affect all job submission protocols that the output device supports and the Pause Printer operations (see [ipp-set2]) which affect only the IPP Job Creation operations to the Printer object. In other words, the Disable Device operations have the same effect on all job submission protocols that the Disable Printer operations have on the IPP job submission protocol.

The Set2 and Set3 documents define distinct operations in order to disambiguate the IPP/1.1 Pause-Printer operation (see [ipp-mod] section 4.4.12 and [ipp-set2]) as shown in Table 4. Set2 Printer operations affect only Jobs submitted using IPP, while Set3 Device operations affect all jobs no matter what job submission protocol was used to submit them to the output device.

Table 4 - Set2 and Set3 Pause and Resume operations

Set2 and Set3 Pause and Resume Printer and Device operations	Description
Pause-Printer-After-Current-Job	Stops the IPP Printer from sending new IPP Jobs to the output device(s) after the current jobs finish
Pause-Printer-After-All-Current-Jobs	Stops the IPP Printer from sending IPP Jobs that are accepted subsequently to the output device(s). All currently pending jobs are scheduled and printed.
Resume-Printer	Starts the IPP Printer sending IPP Jobs to the output device again.
Pause-Device-Now	Stops the output device immediately from producing marked media (current page, sheet, depending on implementation) for any job. Like the Pause button on the output device.
Pause-Device-After-Current-Copy	Stops the output device from producing marked media after the current copy of the current job.
Pause-Device-After-Current-Job	Stops the output device from producing marked media after the current job.
Resume-Device	Starts the output device processing any jobs again.

ISSUE 07 - Should Pause-Printer-After-Current-Job be a new operation with a new operation-id code or be a clarification of the existing IPP/1.1 Pause-Printer operation and use its operation-id? Or should the Pause-Device-Now operation be a new operation-id code or be the clarification of the existing IPP/1.1 Pause-Printer operation and use its operation-id? Or should both Pause-Printer-After-Current-Job and Pause-Device-Now be new operation-id codes and leave the IPP/1.1 Pause-Printer with its current ambiguous (implementer free-for-all) semantics?

9.2.1 Pause-Device-Now, Pause-Device-After-Current-Copy, Pause-Device-After-Current-Job operations

These OPTIONAL operations allows a client to stop the output device from marking the current job. If the output device is in the middle of marking on output media, the IPP Printer MUST stop marking with the immediacy defined for the operation (see Table 4). The Printer object performs a Pause-Printer-After-Current-Job operation (see [ipp-set2]) (which eventually sets the Printer's "printer-state" to 'stopped' and "printer-state-reasons" to 'moving-to-paused' and 'paused') plus controls the output device to stop marking the output media for the current job submitted with any of the output device's job submission protocols. After the IPP Printer receives this operation, the output device MUST NOT start processing or marking any additional jobs. However, the output device MUST continue to accept other operations, including additional jobs, if it would have accepted them before the Printer object received the Pause Device operation.

If the output device is not processing any jobs and/or is not marking output media, the Printer object transitions immediately to the 'stopped' state by setting its "printer-state" attribute to 'stopped', removing the 'moving-to-paused' value, if present, from its "printer-state-reasons" attribute, and adding the 'paused' value to its "printer-state-reasons" attribute.

If the output device will take appreciable time to stop marking the current job that it is marking , the IPP Printer object adds the 'moving-to-paused' value to the Printer object's "printer-state-reasons" attribute (see section [ipp-mod] 4.4.12). When the output device has marking the current job, the Printer object transitions to the 'stopped' state by setting its "printer-state" attribute to 'stopped', removing the 'moving-to-paused' value, if present, from its "printer-state-reasons" attribute, and adding the 'paused' value to its "printer-state-reasons" attribute.

This operation MUST NOT affect the acceptance of other requests (see Disable-Device section 9.1.1).

The IPP Printer MUST accept any of the Pause Device requests in any state and transition the Printer object to the indicated new "printer-state" before returning as follows:

Current "printer-state"	New "printer-state"	"printer-state-reasons"	IPP Printer's response status code and action:
'idle'	'stopped'	'paused'	'successful-ok'
'processing'	'processing'	'moving-to-paused'	'successful-ok'; Later, when the IPP Printer has stopped marking the current job, the "printer-state" becomes 'stopped', and the 'paused' value replaces the 'moving-to-paused' value in the "printer-state-reasons" attribute
'processing'	'stopped'	'paused'	'successful-ok'; the IPP Printer was able to stop the output device immediately
'stopped'	'stopped'	'paused'	'successful-ok'

ISSUE 08 - Or should the Printer's "printer-state" attribute be independent of the Pause Printer operations so that the Pause Device (and Pause Printer) operations don't set the "printer-state" to 'stopped', i.e., the "printer-state" tries to reflect 'idle', 'processing', or 'stopped' of the output device(s) as best it can independent of whether the IPP Printer object is paused or not?

Access Rights: The authenticated user (see [ipp-mod] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5).

The Pause-Device-Now, Pause-Device-After-Current-Copy, and Pause-Device-After-Current-Job requests and responses have the same attribute groups and attributes as the Pause-Printer operation (see [ipp-mod] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see [ipp-set2 section 5.1).

9.2.2 Resume-Device operations

This operation allows a client to resume the output device marking output media. The Printer object performs a Resume-Printer operation (see [ipp-mod] section 3.2.8) (which MUST remove the 'paused' and 'moving-to-paused' values from the Printer object's "printer-state-reasons" attribute, if present, and remove the 'printer-stopped' value from any job's "job-state-reasons" attributes contained in that Printer). If there are no other reasons to keep the output device paused (such as media-jam), the IPP Printer transitions itself

667 to the 'processing' or 'idle' states, depending on whether there are jobs to be processed or not, respectively,
668 and the output device resumes processing jobs.

669 The IPP Printer MUST accept the request in any state, transition the Printer object to the indicated new state
670 as follows:

Current "printer-state"	New "printer-state"	IPP Printer's response status code and action:
'idle'	'idle'	'successful-ok'
'processing'	'processing'	'successful-ok'
'stopped'	'processing'	'successful-ok'; when there are jobs to be processed
'stopped'	'idle'	'successful-ok'; when there are no jobs to be processed.

671 *Access Rights:* The authenticated user (see [ipp-mod] section 8.3) performing this operation must be an
672 operator or administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5).

673 The Disable-Device request and response have the same attribute groups and attributes as the Pause-Device
674 operation (see [ipp-mod] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator"
675 operation attribute (see [ipp-set2] section 5.1).

676 ## 9.3 The Deactivate and Activate Device operations

677 This section defines the OPTIONAL Deactivate-Device and Activate-Device operations that stop and start
678 the output device performing work and accepting all requests, except queries and, therefore, the IPP Printer
679 object performing work and accepting all IPP requests, except queries. If either of these operations are
680 supported, both MUST be supported.

681 These operations allow the operator to put the output device (and IPP Printer object) into a dormant read-
682 only condition and to take it out of such a condition. These operations are a combination of the Deactivate
683 and Pause Device operations, plus preventing the acceptance of any other requests, except queries.

684 The Deactivate and Activate Device operations MUST affect the submission of jobs using other job
685 submission protocols to the associated output device; the Deactivate and Activate Printer operations (see
686 [ipp-set2]) are intended to stop the IPP Printer object from performing IPP work and accepting IPP
687 operations, except IPP query operations.

688 ### 9.3.1 Deactivate-Device operation

689 This OPTIONAL operation allows a client to stop the output device from processing jobs and stop the
690 output device from accepting any, but query requests. The Printer object performs a Deactivate-Printer
691 operation immediately (which performs a Disable-Printer and a Pause-Printer-After-Current-Job including
692 use of all of the "printer-state-reasons" if the operation cannot be completed immediately and immediate
693 rejection all subsequent requests, except Activate-Printer, queries, Send-Document, and Send-URI - see
694 [ipp-set2]).

The IPP Printer MUST accept the request in any state. Immediately, the Printer MUST set the 'device-deactivated' value (see section 6.1) in its "printer-state-reasons" attribute.

Access Rights: The authenticated user (see [ipp-mod] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5).

The Deactivate-Device request and response have the same attribute groups and attributes as the Pause-Printer operation (see [ipp-mod] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see [ipp-set2] section 5.1).

9.3.2 Activate-Device operation

This OPTIONAL operation allows a client to undo the effects of the Deactivate-Device operation, i.e., allow the output device to start or continue marking output media and start the output device accepting any requests from any protocol. The Printer object performs an Enable-Device and a Resume-Device operation immediately. In addition, the output device (and Printer object) MUST immediately start accepting all requests.

The IPP Printer MUST accept the request in any state. Immediately, the Printer MUST immediately remove the device-deactivated' value from its "printer-state-reasons" attribute.

Access Rights: The authenticated user (see [ipp-mod] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5).

The Activate-Device request and response have the same attribute groups and attributes as the Pause-Printer operation (see [ipp-mod] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see [ipp-set2] section 5.1).

9.4 Purge-Device

This OPTIONAL operation allows a client to remove all jobs from the output device regardless of their job submission protocol and regardless of their job states. The IPP Printer object performs a Purge-Jobs operation (see [ipp-mod] section 3.2.9 (which removes all IPP jobs from the IPP Printer, including the Printer object's Job History (see [ipp-mod] section 4.3.7.2). After a Purge-Device operation has been performed, a Printer object MUST return no jobs in subsequent Get-Job-Attributes and Get-Jobs responses (until new jobs are submitted to the output device by any job submission protocol).

IPP/1.1 Purge-Jobs operation has the following implementation option:

> Whether the Purge-Jobs (and Get-Jobs) operation affects jobs that were submitted to the device from other sources than the IPP Printer object in the same way that the Purge-Jobs operation affects jobs that were submitted to the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP protocol is being used as a universal management protocol or just to manage IPP jobs, respectively.

The Purge-Device allows an implementation to support the Purge-Jobs operation to affect only IPP jobs and the Purge-Device to affect all jobs that the output device supports (including IPP jobs).

The effect of this operation on the currently processing job(s), if any, is not specified by this document. Note: If this operation does affect the current job(s), it is expected that the operator would issue this operation on a Printer in the 'idle' state after deactivating the output device (see section 9.3.1) in order to prevent a job from inadvertently being affected by this operation.

ISSUE 09 - Or should we define Purge-Device to cancel any current job rather than having the behavior undefined on output device?

Note: if an operator wants to cancel all jobs without clearing out the Job History, the operator uses the Cancel-Job operation on each job instead of using the Purge-Device or Purge-Jobs operation.

The Printer object MUST accept this operation in any state and transition the Printer object to the 'idle' state.

Access Rights: Authentication and access control (see [ipp-mod] sections 1, 8.3, and 8.5) apply to this operation.

The Purge-Device request and response have the same attribute groups and attributes as the Pause-Printer operation (see [ipp-mod] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see [ipp-set2] section 5.1).

9.5 Reset-Device operation

This OPTIONAL operation allows a client to reset the output device in a number of ways. The Printer object performs a Restart-Printer operation, if implemented, (see [ipp-set2]) (which has the effect of a software reboot which causes the Printer object to set its "printer-state" to 'idle', remove the state reasons from its "printer-state-reasons" attribute, and set its "printer-is-accepting-jobs" attribute to 'true') plus controls the output device to stop marking the output media for the current job submitted with any of the output device's job submission protocols. Then the IPP Printer performs a reset of the output device depending on the "reset-function" operation attribute. The keyword values of this attribute map one-to-one to the enum values that the SNMP Network Management Station (NMS), i.e., the SNMP client, writes into the prtGeneralReset object in the Printer MIB [RFC1759] to affect a reset operation. As in the Printer MIB, the 'reset-to-nvram' (soft reset) value MUST be supported, if this operation is supported. The other values are OPTIONAL.

As the Printer MIB specification [RFC1759] states, if a device does not have NVRAM (non-volatile RAM), the device MUST none-the-less respond to this operation for the 'reset-to-nvram' value with some sort of warm reset that resets the device to some implementation-defined state that is preferably under control of the system administrator by some means outside the scope of the Printer MIB and this document.

The effect of this operation on the currently processing job(s), if any, is not specified by this document. Note: If this operation does affect the current job(s), it is expected that the operator would issue this

operation on a Printer in the 'idle' state after deactivating the output device (see section 9.3.1) in order to prevent a job from inadvertently being affected by this operation.

ISSUE 10 - Or should we define Reset-Device to cancel any current job rather than having the behavior undefined on current jobs in the output device?

The Printer object MUST accept this operation in any state and transition the Printer object to the 'idle' state.

Access Rights: Authentication and access control (see [ipp-mod] sections 1, 8.3, and 8.5) apply to this operation.

The Reset-Printer request and response have the same attribute groups and attributes as the Pause-Printer operation (see [ipp-mod] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see [ipp-set2] section 5.1), with the addition of the following Group 1 operation attributes in the request:

"reset-function" (type3 keyword):

The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute, if it supports this operation. The value of this attribute indicates the reset function to be performed. If the client omits this attribute, the Printer assumes the 'reset-to-nvram' value.

Standard keyword values are:

'power-cycle-reset' - Cold start, i.e., to the state when the device is powered up.
'reset-to-nvram' - Warm start.
'reset-to-factory-defaults' - reset NVRAM to factory defaults, i.e. to factory settings and/or values established at install time.

ISSUE 11 - What happens to 'pending' jobs on a Reset-Device for various values of "reset-function"? If the output device implements persistent jobs, aren't they saved?

9.6 Power-Off-Device operation

This OPTIONAL operation allows a client to power off the output device. The Printer object performs a Shutdown-Printer operation, if implemented, (see [ipp-set2]) (which shuts down the IPP Printer object so that it cannot be access by any IPP protocol operations) plus turns the power off for the output device after the current job completes. There is no way to bring back the output device using the IPP protocol either.

The Printer object MUST accept this operation in any state and transition the Printer object to the 'idle' state.

Access Rights: Authentication and access control (see [ipp-mod] sections 1, 8.3, and 8.5) apply to this operation.

The Power-Off-Device request and response have the same attribute groups and attributes as the Pause-Printer operation (see [ipp-mod] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see [ipp-set2] section 5.1).

10. IANA Considerations

The operations and attributes in this registration proposal will be published by IANA according to the procedures in RFC 2566 [rfc2566] section 6.4 for operations with the following URL:

ftp.isi.edu/iana/assignments/ipp/operations/set2.txt

11. Internationalization Considerations

This document has the same localization considerations as the [ipp-mod].

12. Security Considerations

The IPP Model and Semantics document [ipp-mod] discusses high level security requirements (Client Authentication, Server Authentication and Operation Privacy). Client Authentication is the mechanism by which the client proves its identity to the server in a secure manner. Server Authentication is the mechanism by which the server proves its identity to the client in a secure manner. Operation Privacy is defined as a mechanism for protecting operations from eavesdropping.

13. Author's Addresses

Carl Kugler
IBM
Boulder CA

Phone: (303) 924-5060
FAX:
e-mail: kugler@us.ibm.com

Tom Hastings
Xerox Corporation
737 Hawaii St. ESAE 231
El Segundo, CA 90245

Phone: 310-333-6413
Fax: 310-333-5514
e-mail: hastings@cp10.es.xerox.com

Harry Lewis
IBM
Boulder CA

Phone: (303) 924-5337
FAX:
e-mail: harryl@us.ibm.com

14. References

[ipp-mod]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and Semantics", <draft-ietf-ipp-model-v11-03.txt>, June, 1999.

[RFC1759]
Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759, March 1995.

[RFC2566]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and Semantics", RFC 2566, April 1999.

15. Change History

This section summarizes the changes. Each sub-section is in reverse chronological order.

This is the first version of the Set3 document which separates the Device operations (Set3) from the Printer operations (Set2).

16. Appendix A: Full Copyright Statement