INTERNET-DRAFT - 10 ISSUES are numbered and highlighted like this
<draft-ietf-ipp-ops-set2-01.txt>

Carl Kugler
IBM Corporation
T. Hastings
Xerox Corporation
H. Lewis
IBM Corporation
~~November 16~~December 8, 1999

Internet Printing Protocol/1.1: Set2 Operations

Status of this Memo

Abstract

This document specifies 16~~4~~ additional OPTIONAL operations for use with the Internet Printing Protocol/1.0 (IPP) [RFC2565, RFC2566] and IPP/1.1 [ipp-mod, ipp-pro]. These operations are 9~~7~~ Printer object operations that operators/administrators may perform on a Printer object:

- Set-Printer-Attributes
- Enable-Printer
- Disable-Printer
- Pause-Printer-After-Current-Job (IPP/1.1 Pause-Printer clarified)
- Pause-Printer-After-All-Current-Jobs
- Deactivate-Printer
- Activate-Printer
- Restart-Printer
- Shutdown-Printer
- Startup-Printer

and 7 Job object operations that end-users may perform on their jobs and operators/administrators may perform on any job, depending on circumstances:

- Set-Job-Attributes

Reprocess-Job
Cancel-Current-Job (though the target is the Printer object)
~~Pause~~Suspend-Current-Job (though the target is the Printer object)
Resume-Job
Promote-Job

The Interpreter object is added for those implementations that make a distinction on the values of some Printer attributes depending on the document-format, such as "resolution-supported".

In addition, two operation attributes are defined: "printer-message-from-operator" and "job-message-from-operator" are included to set the corresponding Printer and Job Description attributes with the same names. And the "when" operation attribute is added to the IPP/1.1 Pause-Printer operation as well as some of the Set2 Printer operations.

New Printer Description attributes are added, along with additional values for the "printer-state-reasons" and "job-state-reasons" attributes.

New status codes: 'client-error-attributes-not-settable' and 'server-error-printer-is-in-standby-mode' are added.

Finally, the 'not-settable' out-of-band value is added for use with the Set-Printer-Attributes and Set-Job-Attributes operations.

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]. This version includes the comments discussed at the IPP telecon, on 6/23/1999, 6/30/1999, at the IETF IPP WG meeting, 7/7/99-7/8/99, in Copenhagen, and the IPP telecon, 7/17/1999, the August, 1999 IPP meeting in Alaska and subsequent phone conferences and discussions. Specifically, the 9/16 update refers to this set of extensions simply as "Set2" rather than using the term "Administrative" which was misleading, controversial and incorrect as an overall description. Also, two new attributes have been proposed to clarify the intent of each operation in terms of its target, the Printer vs. the Print Job. Unresolved ISSUES are highlighted like this in the .doc and .pdf files.

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

List of Tables

List of Figures

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 end user, 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.

~~1.1~~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.

~~1.2~~2.2 Other terminology

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].

Output-Device - the physical imaging mechanism that an IPP Printer controls.

Output-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.

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 MAY have one or more Subordinate Printers.

Leaf Printer - a Subordinate Printer that has no Subordinate Printers.

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

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

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

3 Requirements and Use Cases

The following requirements and usage cover both the Set2 and Set3 [ipp-set3] 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. 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.

6. 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.

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

7. 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**

8. Need a Printer operation to stop the processing after the current IPP job completes and not start processing any additional IPP jobs, but continue to accept new IPP jobs. Need a companion operation to start processing IPP jobs again.

   Usage: Operator wants to gracefully stop the IPP Printer as the next job boundary. This operation is also invoked implicitly by the Deactivate-Printer and the Shutdown-Printer.

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

9. 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**

10. 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**

11. 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

12. 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**

13. 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

14. Need a Device operation that combines Disable-Device, Pause-Device-After-Current-Copy (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**

15. 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**

16. 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**

17. 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 a 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**

18. 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**

19. 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 Device operations are:

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

4 New objects and new relationships of existing objects

This section defines the new Interpreter object and the use of the IPP Printer object to represent IPP Printer fan-out and IPP Printer fan-in.

~~1.1~~4.1 Interpreter object

The Interpreter object models the document format interpreters that are contained in the Printer object. The purpose of the Interpreter object is to model those Printer attributes whose value depends on which interpreter is being used to process the document. Depending on implementation, the Printer object

attributes whose values ("xxx-supported" and "xxx-default") depend on the interpreter, i.e., on the "document-format" of the document being processed, are considered to be Interpreter object attributes instead. A Get-Printer-Attributes operation returns Printer and Interpreter attributes as specified in the "requested-attributes" operation attribute supplied by the client. Depending on the value of the "document-format" attribute supplied by the client in the Get-Printer-Attributes request (or the "document-format-default", if the client omits the "document-format" attribute), selects the corresponding Interpreter object.

If an implementation does not make a distinction on the values of Printer attributes by document format, say, for purposes of job validation (see [ipp-mod] Get-Printer-Attributes), there is no need to implement or support the Interpreter object. The Interpreter object is introduced to provide a means to model an implementation in which some attributes do depend on the document format. Those attributes are then sub-classed to be Interpreter object attributes.

Note: the addition of the Interpreter object is completely compatible with IPP/1.0 and IPP/1.1 (see the description of the "document-format" operation attribute in [ipp-mod] section 3.2.5.1 Get-Printer-Attributes request). The protocol and semantics are the same whether or not the Interpreter object is used to distinguish attributes that depend on the "document-format". Consequently, there is no "interpreters-supported" Printer Description attribute. In order to determine which Interpreter objects there MAY be, the client can request the values of the Printer's "document-format-supported" attribute.

Note: The Interpreter object is really a sub-class of the Printer object, rather than being a full fledged object in the sense that the Job and Printer objects are. There are no operations defined solely for the Interpreter object. The Get-Printer-Attributes and Set-Printer-Attributes operations operate on the Interpreter object if the implementation has the concept of an Interpreter object. However, if the IPP Printer implementation does not contain Interpreter objects, the same Interpreter object attributes are considered Printer object attributes, instead.

~~1.2~~4.2 Use of the Printer object to represent IPP Printer fan-out and IPP Printer fan-in

This section defines how the Printer object MAY be used to represent IPP Printer fan-out and IPP Printer fan-in. Fan-out is where an IPP Printer is used to represent other IPP Printer objects. Fan-in is where several IPP Printer objects are used to represent another IPP Printer object.

4.2.1 IPP Printer fan-out

The IPP/1.1 Model and Semantics introduces the semantic concept of an IPP Printer object that represents more than one output device (see [ipp-mod] section 2.1). This concept is called "output device fan-out". However, there was no way to represent the individual states of the output devices or to perform operations on a specific output device when there was fan-out. This specification generalizes the semantics of the Printer object to represent such "subordinate" fan-out output devices as IPP Printer objects. This concept is called "Printer object fan-out". A Printer object that has a subordinate Printer object is called a "non-leaf" Printer object. Thus a "non-leaf" Printer object MAY support one or more subordinate Printer objects in order to represent Printer object fan-out. A Printer object that does not have any subordinate Printer objects is called a "leaf" Printer object.

~~1.1.2~~4.2.2 IPP Printer fan-in

The IPP/1.1 Model and Semantics did not preclude the semantic concept of multiple IPP Printer objects that represent a single output device (see [ipp-mod] section 2.1). However, there was no way for the client to determine that there was a fan-in configuration, nor was there a way to perform operations on the subordinate device. This specification generalizes the semantics of the Printer object to allow several "non-leaf" IPP Printer objects to represent a single "subordinate" Printer object. Thus a "non-leaf" Printer object MAY share a subordinate Printer object with one or more other non-leaf Printer objects in order to represent IPP Printer fan-in. As with fan-out (see section 4.2), when a Printer object is a non-leaf, it MUST NOT have an associated output device. As with fan-out, a leaf Printer object has an associated output device(s). As with fan-out, the non-leaf Printer objects submit jobs to their subordinate Printer objects and otherwise control the subordinate Printer. As with fan-out, whether pending jobs are kept in the non-leaf Printers until the subordinate Printer can accept them or are kept in the subordinate Printer depends on implementation and/or configuration policy.

~~1.1.3~~4.2.3 Printer object attributes used to represent Printer fan-out and Printer fan-in

Each non-leaf Printer object submits jobs to its immediate subordinate Printers and otherwise controls the subordinate Printers using IPP or other protocols. Whether pending jobs are kept in the non-leaf Printer until a subordinate Printer can accept them or are kept in the subordinate Printers depends on implementation and/or configuration policy. Furthermore, a subordinate Printer object MAY, in turn, have subordinate Printer objects. Thus a Printer object can be both a non-leaf Printer and a subordinate Printer.

A subordinate Printer object MUST be a conforming Printer object, so it MUST support all of the REQUIRED operations and attributes. However, with access control, the subordinate Printer MAY be configured so that end-user clients are not permitted to perform any operations (or just Get-Printer-Attributes) while one or more non-leaf Printer object(s) are permitted to perform any operation.

The following Printer Description attributes are defined to represent the relationship between Printer object(s) and their subordinate Printer object(s):

1. "subordinate-printers-supported" (1setOf uri) - contains the URI of the immediate subordinate Printer object(s). Each non-leaf Printer object MUST support this Printer Description attribute. A leaf Printer object either does not support the "subordinate-printers-supported" attribute or does so with the 'no-value' out-of-band value (see [ipp-mod] section 4.1), depending on implementation.

2. "parent-printers-supported (1setOf uri) - contains the URI of the non-leaf printer object(s) for which this Printer object is the immediate subordinate, i.e., this Printer's immediate "parent" or "parents". Each subordinate Printer object MUST support this Printer Description attribute. A Printer that has no parents, either does not support the "parent-printers-supported" attribute or does so with the 'no-value' out-of-band value (see [ipp-mod] section 4.1), depending on implementation.

4.2.4 Subordinate Printer URI

Each subordinate Printer object has a URI which is used as the target of each operation on the subordinate Printer. The means for configuring URIs for subordinate Printer objects is implementation-dependent as are all URIs. However, there are two distinct approaches:

a. When the implementation wants to make sure that no operation on a subordinate Printer object as a target "sneaks by" the parent Printer object (or the subordinate Printer is fronting for a device that is not networked), the host part of the URI specifies the host of the parent Printer. Then the parent Printer object can easily reflect the state of the subordinate Printer objects in the parent's Printer object state and state reasons as the operation passes "through" the parent Printer object.

b. When the subordinate Printer is networked and the implementation allows operations to go directly to the subordinate Printer (with proper access control) without knowledge of the parent Printer object, the host part of the URI is different than the host part of the parent Printer object. In such a case, the parent Printer object MUST keep its "printer-state" and "printer-state-reasons" up to date, either by polling the subordinate Printer object or by subscribing to events with the subordinate Printer object (see [ipp-not-spec] for means to ~~do this~~subscribe to event notification when the subordinate Printer object supports IPP notification).

~~1.1.5~~4.2.5 Printer object attributes used to represent output device fan-out

When a Printer object is a non-leaf, it MUST NOT have an associated output device. Only leaf IPP Printer objects are allowed to have one or more associated output devices. Each leaf Printer object MAY support the "output-devices-supported" (1setOf name(127)) to indicate the user-friendly name(s) of the output device(s) that the leaf Printer object represents. It is RECOMMENDED that each leaf Printer object have only one associated output device, so that the individual output devices can be represented completely and controlled completely by clients. In other words, the Printer's "output-devices-supported" attribute SHOULD have only one value.

~~1.1.6~~4.2.6 Figures to show all possible configurations

Figure 1~~Figure 1~~, Figure 2~~Figure 2~~, and Figure 3~~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
```

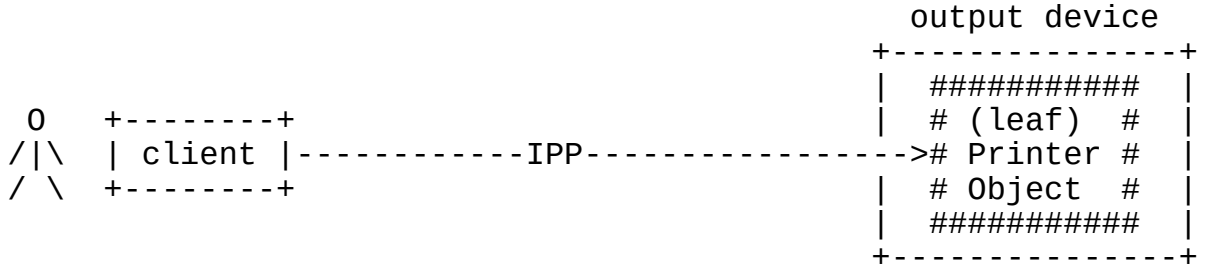


Figure 1 - embedded Printer object

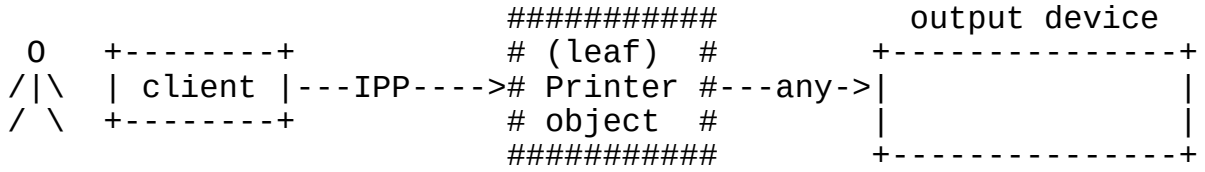


Figure 2~~2~~ - hosted Printer object

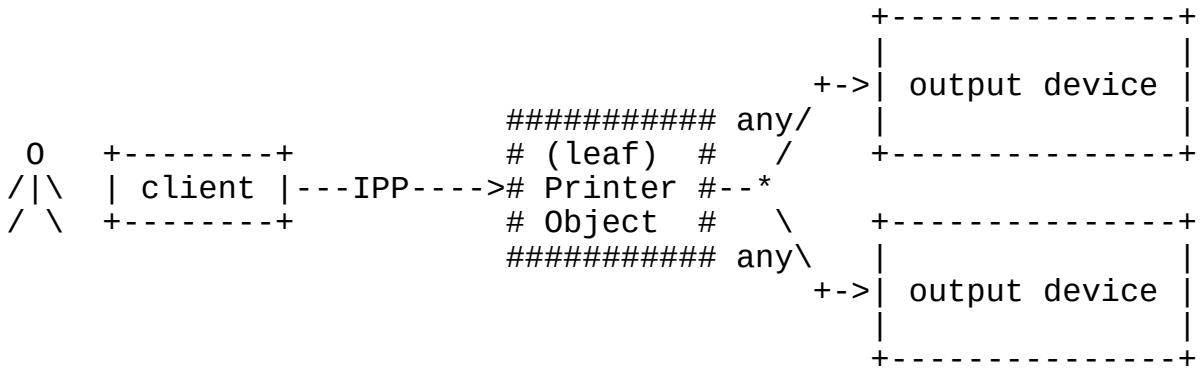
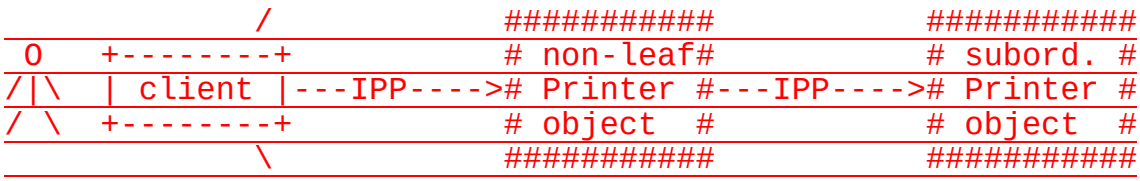
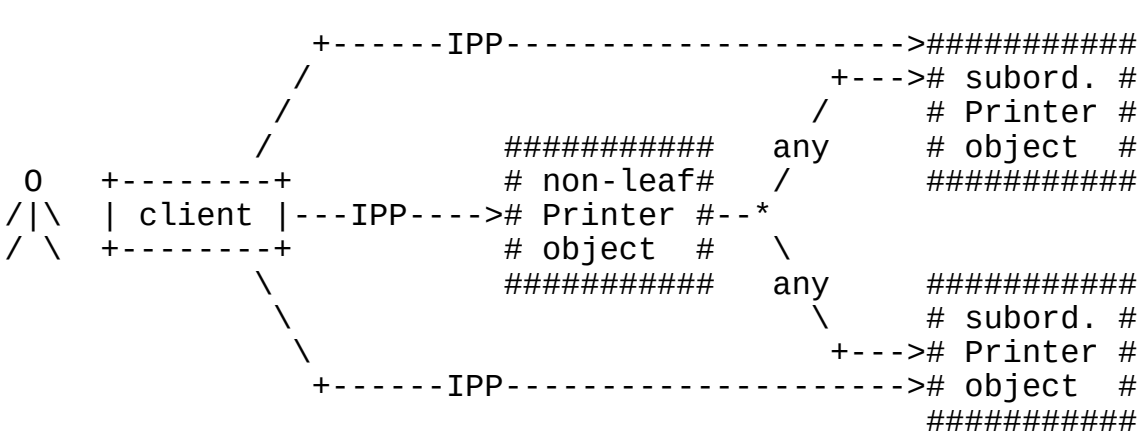


Figure 33 - output device fan out



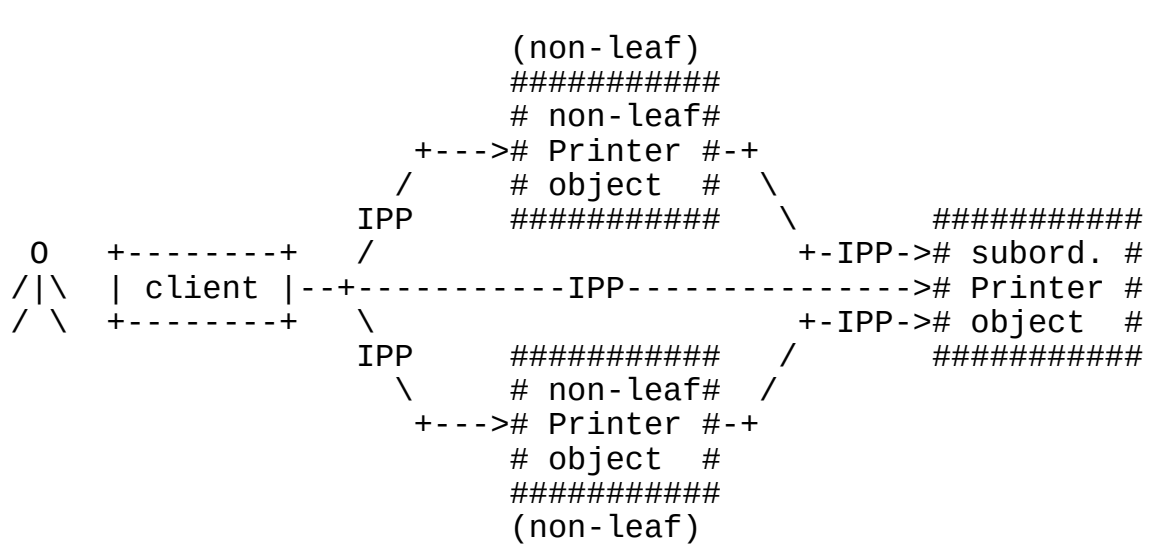
The subordinate Printer can be a non-leaf Printer as in Figure 44, Figure 5Figure 5, or Figure 6Figure 6, or can be a leaf Printer as in Figure 1, Figure 2Figure 2, or Figure 3Figure 3.

Figure 44 - Chained IPP Printer



The subordinate Printer can be a non-leaf Printer as in Figure 44, Figure 5Figure 5, or Figure 6Figure 6, or can be a leaf Printer as in Figure 1, Figure 2Figure 2, or Figure 3Figure 3.

Figure 55 - IPP Printer fan out



The subordinate Printer can be a non-leaf Printer as in Figure 44, Figure 5Figure 5, or Figure 6Figure 6, or can be a leaf Printer as in Figure 1, Figure 2Figure 2, or Figure 3Figure 3.

Figure 66 - IPP Printer fan in

1.34.3 Forwarding requests

This section describes the forwarding of Job and Printer requests to subordinate Printer objects.

4.3.1 Forwarding requests that affect Printer objects

In both Printer fan-out, and 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 section 6.6).

Table 1Table 1 lists the operations that affect Printer objects and the forwarding behavior that a non-leaf Printer MUST exhibit to its immediate subordinate Printers. Printer operations that affect jobs have a different forwarding rule (see section 4.3.2 and Table 2Table 2):

Table 1 - Forwarding operations that affect Printer objects

Printer operation	Non-leaf Printer action
Set2 Printer operations:	
Set-Printer-Attributes	MUST NOT forward to any of its subordinate Printers
Enable-Printer	MUST NOT forward to any of its subordinate Printers
Disable-Printer	MUST NOT forward to any of its subordinate Printers
Pause-Printer-After-All-	MUST NOT forward to any of its subordinate Printers

Current-Jobs	
Deactivate-Printer	MUST NOT forward to any of its subordinate Printers
Activate-Printer	MUST NOT forward to any of its subordinate Printers
Restart-Printer	MUST NOT forward to any of its subordinate Printers
Shutdown-Printer	MUST NOT forward to any of its subordinate Printers
Startup-Printer	MUST NOT forward to any of its subordinate Printers
IPP/1.1 Printer operations:	
Get-Printer-Attributes	MUST NOT forward to any of its subordinate Printers
Pause-Printer	MUST NOT forward to any of its subordinate Printers
Resume-Printer	MUST NOT forward to any of its subordinate Printers

~~ISSUE 01 - Should the purpose of the corresponding device operations, such as Pause-Device, Disable-Device, and Shutdown-Device (see [ipp-set3], to be forwarded by Printer objects to their subordinate Printers down to the leaf Printers objects in order to affect their corresponding output device? Using device operations in this way would relieve the operator client from having to work its way down the Printer object chain in order to affect an output device of a leaf Printer object.~~

~~ISSUE 02 - If a Printer object MUST forward a device operation down to the leaf Printer objects, would fanning out such forwarded device operations be too drastic? If yes, then maybe the device operations would need an operation attribute to say whether or not to allow fan-out of device operations, with the default value being 'false'.~~

~~ISSUE 03 - Does the administrator need a Printer object attribute that says whether or not this Printer object is to forward device operations when it has more than one subordinate Printer object? The default would be 'false'.~~

~~1.1.2~~4.3.2 Forwarding requests that affect Jobs

Unlike Printer operations that only affect Printer objects (see section 4.3.1), a non-leaf Printer object MUST forward operations that directly affect jobs to the appropriate Job object(s) in one or more of its immediate subordinate Printer objects. Forwarding is REQUIRED since the purpose of such a Job operation is to affect the indicated job which itself may have been forwarded. Such forwarding MAY be immediate or queued, depending on the operation and the implementation. For example, a non-leaf Printer object MAY queue/spool jobs, feeding a job at a time to its subordinate Printer(s), or MAY forward jobs immediately to one of its subordinate Printers. In either case, the non-leaf Printer object is forwarding Job Creation operations to one of its subordinate Printers. Only the time of forwarding of the Job Creation operations depends on whether the policy is to queue/spool jobs in the non-leaf Printer or the subordinate Printer.

When a non-leaf Printer object creates a Job object in its subordinate Printer, whether that non-leaf Printer object keeps a fully formed Job object or just keeps a mapping from the "job-ids" that it assigned to those assigned by its subordinate Printer object is IMPLEMENTATION-DEPENDENT. In either case, the non-leaf Printer MUST be able to accept and carry out future Job operations that specify the "job-id" that the non-leaf Printer assigned and returned to the job submitting client.

570 Table 2~~Table 2~~ lists the operations that directly affect jobs and the forwarding behavior that a non-leaf
571 Printer MUST exhibit to its subordinate Printers:

572 **Table 2~~2~~ - Forwarding operations that affect Jobs objects**

Job operation	Non-leaf Printer action
Set2 Job operations:	
Set-Job-Attributes	MUST forward to the appropriate Job in one of its subordinate Printers
Reprocess-Job	MUST forward to the appropriate Job in one of its subordinate Printers
Cancel-Current-Job	MUST forward to the appropriate Job in one of its subordinate Printers
~~Pause~~Suspend-Current-Job	MUST forward to the appropriate Job in one of its subordinate Printers
Resume-Job	MUST forward to the appropriate Job in one of its subordinate Printers
Promote-Job	MUST forward to the appropriate Job in one of its subordinate Printers
IPP/1.1 Printer operations:	
Print-Job	MUST forward immediately or queue to the appropriate subordinate Printer
Print-URI	MUST forward immediately or queue to the appropriate subordinate Printer
Validate-Job	MUST forward to the appropriate subordinate Printer
Create-Job	MUST forward immediately or queue to the appropriate subordinate Printer
Get-Jobs	MUST forward to *all* its subordinate Printers
Purge-Jobs	MUST forward to *all* its subordinate Printers
IPP/1.1 Job operations:	
Send-Document	MUST forward immediately or queue to the appropriate Job in one of its subordinate Printers
Send-URI	MUST forward immediately or queue to the appropriate Job in one of its subordinate Printers
Cancel-Job	MUST forward to the appropriate Job in one of its subordinate Printers
Get-Job-Attributes	MUST forward to the appropriate Job in one of its subordinate Printers, if the non-leaf Printer doesn't know the complete status of the Job object
Hold-Job	MUST forward to the appropriate Job in one of its subordinate Printers
Release-Job	MUST forward to the appropriate Job in one of its subordinate Printers
Restart-Job	MUST forward to the appropriate Job in one of its subordinate Printers

573 ISSUE 01: Do we want to define whether the response to the client for Job operations can happen before
574 the non-leaf Printer gets the response from its subordinate Printer or MUST the non-leaf Printer wait until
575 its gets the response from its subordinate Printer?

576 The following Job Description attributes are defined to help represent Job relationships for fan-out and
577 forwarding of jobs:

578 1. "output-device-assigned" (name(127)) - from [ipp-mod]: This attribute identifies the output device to
579 which the Printer object has assigned this job. If an output device implements an embedded Printer

object, the Printer object NEED NOT set this attribute. If a print server implements a Printer object, the value MAY be empty (zero-length string) or not returned until the Printer object assigns an output device to the job. This attribute is particularly useful when a single Printer object supports multiple devices (so called "fan-out").

2. "original-requesting-user-name" (name(MAX)) - operation attribute containing the user name of the original user, i.e., corresponds to the "requesting-user-name" operation attribute that the original client supplied to the first Printer object.

ISSUE 02 - The "requesting-user-name" operation attribute is the parent Printer's host, not the original requesting user, correct?)

ISSUE 03 - Presumably the "job-originating-user-name" remains as the authenticated original user, not the parent Printer's authenticated host, correct?

5 New Operation attributes

This section defines the new "printer-message-from-operation" and "job-message-from-operator" operation attributes that set the corresponding Printer and Job Description attributes.

~~1.1~~5.1 "printer-message-from-operator" (text(127))

~~Type of registration: attribute~~
~~Proposed keyword name of this attribute: "printer-message-from-operator"~~
~~Types of attribute (Operation, Job Template, Job Description, Printer Description): Operation~~
~~Operations to be used with if the attribute is an operation attribute: See below~~
~~Object (Job, Printer, etc. if bound to an object): Printer (already in IPP/1.0 and IPP/1.1)~~
~~Attribute syntax(es) (include 1setOf and range as in Section 4.2): text(127)~~

~~Specification of this attribute (follow the style of IPP Model Section 4.2):~~

~~"printer-message-from-operator" (text(127))~~

The client OPTIONALLY supplies this attribute. The Printer object SHOULD supports this operation attribute if it supports the corresponding Printer Description attribute. The value of this attribute is a message from the operator about the Printer object on which the operator is performing the operation. If supported, the Printer copies the value to the Printer's "printer-message-from-operator" Printer Description attribute (see [ipp-mod] section 4.4.25), automatically sets the value of the Printer's "printer-message-time" with the current value of the Printer's "printer-up-time" attribute, the value of the "printer-message-date-time" with the current value of the Printer's printer-current-date-time" attribute, and the value of the Printer's "printer-message-operation" with the operation-id value of this operation (see [ipp-mod] section 4.4.15).

If the client omits this attribute, the Printer does not change the value of its "printer-message-from-operator", "printer-message-time", "printer-message-date-time", and "printer-message-operation" Printer

Description attributes.

If the client supplies this attribute with a zero-length text value or with a value consisting solely of white space, the Printer copies that value as any other value to the Printer's "printer-message-from-operator" and sets the "printer-message-time", "printer-message-date-time", and "printer-message-operation" attributes. Supplying such a value is the way that the operator indicates that there is no longer a printer message from the operator (rather than using the "out-of-band" 'no-value' value).

This operation attribute is defined for use with the following operator operations on the Printer object:

Pause-Printer - see [ipp-mod] section 3.2.7
Resume-Printer - see [ipp-mod] section 3.2.8
Purge-Jobs - see [ipp-mod] section 3.2.9
Disable-Printer - see section 11.2
Enable-Printer - see section 11.2.2
Pause-Printer-After-Current-Job (IPP/1.1 Pause-Printer clarified) - see section 11.3.2
Pause-Printer-After-All-Current-Jobs - see section
Deactivate-Printer - see section 11.4.1
Activate-Printer - see section 11.4.2
Restart-Printer - see section 11.5.1
Shutdown-Printer - see section 11.5.2
Startup-Printer - see section 11.5.3

The "printer-message-from-operator" operation attribute MUST NOT be supported as an operation attribute for the Set-Printer-Attributes operation. If the operator wants to set the Printer's "printer-message-from-operator" Printer Description attribute when issuing the Set-Printer-Attributes operation, the client supplies the "printer-message-from-operator" with its new value as one of the Printer Description attributes in Group 2 in the request. The Printer also updates the Printer's "printer-message-date-time" and "printer-message-operation" Printer Description attributes. If the client does not explicitly supply the "printer-message-from-operator" with its new value, the Printer leaves the value of the Printer's "printer-message-from-operator" Printer Description attribute unchanged.

~~1.2~~5.2 "job-message-from-operator" (text(127))

~~Type of registration: attribute~~
~~Proposed keyword name of this attribute: "job-message-from-operator"~~
~~Types of attribute (Operation, Job Template, Job Description, Printer Description): Operation~~
~~Operations to be used with if the attribute is an operation attribute: See below~~
~~Object (Job, Printer, etc. if bound to an object): Job (already in IPP/1.0 and IPP/1.1)~~
~~Attribute syntax(es) (include 1setOf and range as in Section 4.2): text(127)~~

~~Specification of this attribute (follow the style of IPP Model Section 4.2):~~

~~"job-message-from-operator" (text(127))~~

The client OPTIONALLY supplies this attribute. The Printer object SHOULD supports this operation attribute if it supports the corresponding Job Description attribute. The value of this attribute is a message from the operator about the Job object on which the operator has just performed an operation. If supported, the Printer copies the value to the Job's "job-message-from-operator" Job Description attribute (see [ipp-mod] section 4.3.16).

If the client omits this attribute, the Printer does not change the value of its "printer-message-from-operator" Job Description attribute.

If the client supplies this attribute with a zero-length text value or with a value consisting solely of white space, the Printer copies that value as any other value to the job's "job-message-from-operator". Supplying such a value is the way that the operator indicates that there is no longer a job message from the operator (rather than using the "out-of-band" 'no-value' value).

Note: There are no corresponding 'job-message-time", "job-message-date-time", and "job-message-operation" Job Description attributes, since the usual lifetime of a job is limited.

This operation attribute is defined for use with the following operator operations on the Job object:

Cancel-Job - see [ipp-mod] section 3.2.4
Hold-Job - see [ipp-mod] section 3.3.5
Release-Job - see [ipp-mod] section 3.3.6
Restart-Job - see [ipp-mod] section 3.3.7
Reprocess-Job - see section 12.2
Cancel-Current-Job - see section 12.3
~~Pause~~Suspend-Current-Job - see section 12.4
Resume-Job - see section 12.4.2
Promote-Job - see section 12.5

The "job-message-from-operator" operation attribute MUST NOT be supported as an operation attribute for the Set-Job-Attributes operation. If the operator wants to set the Job's "job-message-from-operator" Job Description attribute when issuing the Set-Job-Attributes operation, the client supplies the "job-message-from-operator" with its new value as one of the Job Description attributes in Group 2 in the request. Otherwise, the Printer leaves the value of the Job's "job-message-from-operator" Job Description attribute unchanged by not explicitly setting the attribute.

~~1.3~~5.3 "factory-settings" (boolean) - new operation attribute for Get-Printer-Attributes:

~~Type of registration: attribute~~
~~Proposed keyword name of this attribute: "factory-settings"~~
~~Types of attribute (Operation, Job Template, Job Description, Printer Description): Operation~~
~~Operations to be used with if the attribute is an operation attribute: Get-Printer-Attributes~~
~~Object (Job, Printer, etc. if bound to an object): N/A~~
~~Attribute syntax(es) (include 1setOf and range as in Section 4.2): boolean~~

~~Specification of this attribute (follow the style of IPP Model Section 4.2):~~
~~"factory-settings" (boolean)~~

The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this attribute, if it supports the Set-Printer-Attributes operation. If the client omits this attribute or supplies the 'false' value, the Printer returns the current values of the requested attributes that are settable, i.e., the values that have been set by previous Set-Printer-Attributes. If the client supplies the 'true' value, the Printer returns the factory settings, i.e., the inherent values supported by the implementation as shipped from the manufacturer or established at install time. This operation attribute allows an operator to determine which values are supported in an implementation after having modified a settable attribute. Attributes that are not settable are not affected by this operation attribute, so that the Printer returns the same values for non-settable attribute when either the 'true' or 'false' value has been supplied. If this operation attribute is supported, both 'true' and 'false' values MUST be supported.

~~5.4New operation attribute for Pause-Printer: "when" (type2 keyword)~~

~~Type of registration: attribute~~
~~Proposed keyword name of this attribute: "when"~~
~~Types of attribute (Operation, Job Template, Job Description, Printer Description): Operation~~
~~Operations to be used with if the attribute is an operation attribute: Pause-Printer, Shutdown-Printer, and Pause-Current-Job~~
~~Object (Job, Printer, etc. if bound to an object): N/A~~
~~Attribute syntax(es) (include 1setOf and range as in Section 4.2): type2 keyword~~

~~Specification of this attribute (follow the style of IPP Model Section 4.2):~~
~~"when" (type2 keyword)~~
~~The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this attribute, if it supports this operation. The value of this attribute indicates when to pause the printer. If the client omits this attribute, the Printer assumes the 'now' value. The 'now' value is REQUIRED to be supported if the "when" attribute is supported; the remaining values are OPTIONAL.~~
~~Note: There is no way to query to see which values of the "when" attribute are supported for this or any other operation. Since this operation is intended for Operators, rather than End-Users, it did not seem necessary to provide the means to query the supported values.~~

~~Standard keyword values are:~~
~~'now' - pause the Printer immediately, while the current job is processing. See [ipp-mod] section 3.2.7) for the effect on the Printer's "printer-state" and "printer-state-reasons" attributes. Jobs in the 'held' and 'pending' state remain in those states.~~
~~'after-current-copy' - pause the Printer after the current job finishes printing its current copy. Jobs in the 'held' and 'pending' state remain in those states.~~
~~'after-current-job' - pause the Printer after the current job finishes printing (all its copies). Jobs in the 'held' and 'pending' state remain in those states.~~
~~'after-all' - pause the Printer after all 'pending' jobs finish printing. Jobs in the 'held' state remain in the 'held' state.~~

~~Note: The "when" operation attribute is also defined for use with the Shutdown-Printer (see section 10.1.5) and the Pause-Current-Job (see section 10.2.4) operations, but with a subset of these values.~~

~~1.4~~5.4 Summary of the operation attributes for the Printer operations

The following table shows the operation attributes for the Printer operations that a client MAY supply in a request. Operation parameters and attributes that a client MUST supply are not shown. Also "requesting-user-name" is not shown, though it is RECOMMENDED that a client supply it in every request.

Legend:

R - REQUIRED for a Printer to support

O - OPTIONAL for a Printer to support; the Printer ignores the attribute if not supported

Operation Attribute	Pause-Printer	Resume-Printer	Purge-Jobs	Get-Printer-Attributes	Set-Printer-Attributes	Enable-Print	Disable-Printer	Restart-Printer	Shut down-Printer
document-format				R	R				
factory-settings				O					
printer-message-from-operator	O	O	O			O	O	O	O
job-type						O	O		
~~when~~	~~O~~								~~O~~
reset-function									
~~synchronize~~									~~O~~
~~shutdown-function~~									~~R~~

~~5.6~~5.5 Summary of the operation attributes for the Job operations

The following table shows the operation attributes for the Job operations that a client MAY supply in a request. Operation attributes that specify the Printer or Job object as the target are shown in the first two rows, respectively. Other operation attributes and parameters that a client MUST supply are not shown. Also "requesting-user-name" is not shown, though it is RECOMMENDED that a client supply it in every request.

Legend:

R - REQUIRED for a Printer to support

O - OPTIONAL for a Printer to support; the Printer ignores the attribute if not supported

Operation Attribute	Cancel-Job	Cancel-Current-Job	Hold-Job	Release-Job	~~Pause~~ Suspend-Current-Job	Resume-Job	Get-Job-Attributes	Set-Job-Attributes	Restart-Job	Reprocess-Job	Promote-Job
printer-uri		R			R						
printer-uri/job-id or job-uri	R		R	R		R	R	R	R	R	R
job-id		R			R						
requested-attributes							R				
~~back-space~~											

Operation Attribute	Cancel-Job	Cancel-Current-Job	Hold-Job	Release-Job	~~Pause~~ Suspend-Current-Job	Resume-Job	Get-Job-Attributes	Set-Job-Attributes	Restart-Job	Reprocess-Job	Promote-Job
~~forward-space~~											
~~synchronize~~					~~O~~						
~~when~~					~~O~~						
job-message-from-operator	O	O	O	O	O	O			O	O	O
message [to-operator]	O		O	O	O	O			O	O	O
job-hold-until			O*	O*						O**	

* The Printer MUST support the "job-hold-until" operation attribute if it supports the "job-hold-until" Job Template attribute.

** The Printer MUST support the "job-hold-until" operation attribute if it supports the Set-Job-Attributes operation, so that the client can hold the job with the Reprocess-Job operation and the modify the job before releasing it to be processed.

6 New Printer Description Attributes

The following new Printer Description attributes are needed to support the new operations defined in this document.

~~1.1~~6.1 printer-settable-attributes (1setOf type2 keyword)

~~Type of registration: attribute~~
~~Proposed keyword name of this attribute: printer-settable-attributes~~
~~Types of attribute (Operation, Job Template, Job Description, Printer Description): Printer Description~~
~~Operations to be used with if the attribute is an operation attribute: N/A~~
~~Object (Job, Printer, etc. if bound to an object): Printer~~
~~Attribute syntax(es) (include 1setOf and range as in Section 4.2): 1setOf type2 keyword~~
~~If attribute syntax is 'keyword' or 'enum', is it type2 or type3: type2~~
~~If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2): Yes~~
~~If this is a Job Template attribute, how does its specification depend on the value of the "multiple-document-handling" attribute: N/A~~
~~Specification of this attribute (follow the style of IPP Model Section 4.2):~~

~~4.4.? printer-settable-attributes (1setOf type2 keyword)~~

This READ-ONLY Printer attribute identifies the Printer object attributes that are settable in this implementation, i.e., that are settable using the Set-Printer-Attributes operations (see section 11.1). This attribute MUST be supported if the Set-Printer-Attributes operations is supported. The Printer MUST reject attempts to set any Printer attributes that are not one of the values of this attribute~~in this list~~, returning the 'client-error-attributes-not-settable' status code (see section 9.1). The value of this attribute MAY

depend on the value of the "document-format" operation attribute supplied in the Get-Printer-Attributes operation (see [ipp-mod] section 3.2.5.1).

Standard keyword values are:

'none': There are no settable Printer attributes.
'xxx': Where 'xxx' is any of the keyword attribute names in the list defined in section 11.1.1

~~1.2~~6.2 job-settable-attributes (1setOf type2 keyword)

~~Type of registration: attribute~~
~~Proposed keyword name of this attribute: job-settable-attributes~~
~~Types of attribute (Operation, Job Template, Job Description, Printer Description): Printer Description~~
~~Operations to be used with if the attribute is an operation attribute: N/A~~
~~Object (Job, Printer, etc. if bound to an object): Printer~~
~~Attribute syntax(es) (include 1setOf and range as in Section 4.2): 1setOf type2 keyword~~
~~If attribute syntax is 'keyword' or 'enum', is it type2 or type3: type2~~
~~If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2): Yes~~
~~If this is a Job Template attribute, how does its specification depend on the value of the "multiple-document-handling" attribute: N/A~~
~~Specification of this attribute (follow the style of IPP Model Section 4.2):~~

~~4.4.? job-settable-attributes (1setOf type2 keyword)~~

This READ-ONLY Printer attribute identifies the Job object attributes that are settable in this implementation, i.e., that are settable using the Set-Job-Attributes operation~~s~~ (see section 12.1). This attribute MUST be supported if the Set-Job-Attributes operations is supported. The Printer MUST reject attempts to set any Job attributes that are not one of the values of this attribute~~in the list~~, returning the 'client-error-attributes-not-settable' status code (see section 9.1).

Standard keyword values are:

'none': There are no settable Job attributes.
'xxx': Where 'xxx' is any of the keyword attribute names in the list defined in section 12.1.1

~~6.3printer-controls-other-protocols (boolean)~~

~~Type of registration: attribute~~
~~Proposed keyword name of this attribute: printer-controls-other-protocols~~
~~Types of attribute (Operation, Job Template, Job Description, Printer Description): Printer Description~~
~~Operations to be used with if the attribute is an operation attribute: N/A~~
~~Object (Job, Printer, etc. if bound to an object): Printer~~
~~Attribute syntax(es) (include 1setOf and range as in Section 4.2): boolean~~
~~If attribute syntax is 'keyword' or 'enum', is it type2 or type3: N/A~~
~~If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2): No~~
~~If this is a Job Template attribute, how does its specification depend on the value of the "multiple-document-handling" attribute: N/A~~
~~Specification of this attribute (follow the style of IPP Model Section 4.2):~~

~~4.4.? printer-controls-other-protocols (boolean)~~

~~This Printer attribute indicates whether all IPP printer and job operations, such as Disable-Printer, Pause-Printer, Cancel-Job, etc., affect non-IPP protocols that may be supported. It is RECOMMENDED that IPP control other non-IPP protocols. However, this attribute permits an implementation to indicate explicitly whether it does affect other protocols or not.~~

~~A 'false' value indicates that IPP operations only affect jobs submitted by the IPP Protocol. For example, a 'true' value indicates that a Disable-Printer operation prevents all protocols from submitting jobs, not just the IPP protocol. A 'true' value indicates that the Pause-Printer operation would pause the current job, no matter what job submission protocol had submitted it. A 'true' value indicates that the Cancel-Job operation will cancel the specified job even if it was submitted using a different job submission protocol, such as LPR.~~

~~As with any Printer Description attribute that this specification does not list as READ-ONLY, an implementation MAY allow a client to change the value of this attribute using Set-Printer-Attributes, thereby changing the way that the IPP operations affect other non-IPP protocols. An implementation MAY also support other means to change the value of this attribute, such as via the console or at installation time.~~

~~6.4~~6.3 printer-message-time (integer(MIN:MAX))

~~Type of registration: attribute~~
~~Proposed keyword name of this attribute: printer-message-time~~
~~Types of attribute (Operation, Job Template, Job Description, Printer Description): Printer Description~~
~~Operations to be used with if the attribute is an operation attribute: N/A~~
~~Object (Job, Printer, etc. if bound to an object): Printer~~
~~Attribute syntax(es) (include 1setOf and range as in Section 4.2): integer(MIN:MAX)~~
~~If attribute syntax is 'keyword' or 'enum', is it type2 or type3: N/A~~
~~If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2): No~~
~~If this is a Job Template attribute, how does its specification depend on the value of the "multiple-document-handling" attribute: N/A~~
~~Specification of this attribute (follow the style of IPP Model Section 4.2):~~

~~4.4.? printer-message-time (integer(MIN:MAX))~~

This READ-ONLY Printer Description attribute contains the time that the Printer's "printer-message-from-operator" was changed by the operator using any operation where the client supplied the "printer-message-from-operator" operation attribute (see section 5.1) or was explicitly set using the Set-Printer-Attributes operation (see section 11.1). This attribute allows the users to know when the "printer-message-from-operator" attribute was last set.

The Printer sets the value of this attribute by copying the value of the Printer's "printer-up-time" attribute (see [ipp-mod] section 4.3.14). If the Printer resets its "printer-up-time" attribute to 1 on power-up, then it MUST change the value of the "printer-message-time" to 0 or a negative number as specified in [ipp-mod] section 4.3.14.

Note: This attribute helps users better understand the context for the "printer-message-from-operator" message.

~~6.5~~6.4 printer-message-date-time (dateTime)

~~Type of registration: attribute~~
~~Proposed keyword name of this attribute: printer-message-date-time~~
~~Types of attribute (Operation, Job Template, Job Description, Printer Description): Printer Description~~
~~Operations to be used with if the attribute is an operation attribute: N/A~~
~~Object (Job, Printer, etc. if bound to an object): Printer~~
~~Attribute syntax(es) (include 1setOf and range as in Section 4.2): dateTime~~
~~If attribute syntax is 'keyword' or 'enum', is it type2 or type3: N/A~~
~~If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2): No~~
~~If this is a Job Template attribute, how does its specification depend on the value of the "multiple-document-handling" attribute: N/A~~
~~Specification of this attribute (follow the style of IPP Model Section 4.2):~~

~~4.4.? printer-message-date-time (dateTime)~~

This READ-ONLY Printer Description attribute contains the date and time that the Printer's "printer-message-from-operator" was changed by the operator using any operation where the client supplied the "printer-message-from-operator" operation attribute (see section 5.1) or was explicitly set using the Set-Printer-Attributes operation (see section 11.1). This attribute allows the users to know when the "printer-message-from-operator" attribute was last set. This attribute MUST be supported if the Printer supports both the "printer-message-from-operator" and the "printer-current-date-time" attributes.

Note: This attribute helps users better understand the context for the "printer-message-from-operator" message.

~~6.6~~6.5 printer-message-operation (type2 keyword)

~~Type of registration: attribute~~
~~Proposed keyword name of this attribute: printer-message-operation~~
~~Types of attribute (Operation, Job Template, Job Description, Printer Description): Printer Description~~
~~Operations to be used with if the attribute is an operation attribute: N/A~~
~~Object (Job, Printer, etc. if bound to an object): Printer~~
~~Attribute syntax(es) (include 1setOf and range as in Section 4.2): type2 enum~~
~~If attribute syntax is 'keyword' or 'enum', is it type2 or type3: type2~~
~~If this is a Printer attribute, MAY the value returned depend on "document-format" (See Section 6.2): No~~
~~If this is a Job Template attribute, how does its specification depend on the value of the "multiple-document-handling" attribute: N/A~~
~~Specification of this attribute (follow the style of IPP Model Section 4.2):~~

~~4.4.? printer-message-operation (type2 enum)~~

This READ-ONLY Printer Description attribute contains the operation that was used to changed the Printer's "printer-message-from-operator" by the operator using any operation where the client supplied the "printer-message-from-operator" operation attribute (see section 5.1) or explicitly set using the Set-Printer-Attributes operation (see section 11.1). This attribute allows the users to know which operation was used to change the "printer-message-from-operator" attribute when it was last set.

Note: This attribute helps users better understand the context for the "printer-message-from-operator".

~~6.7~~6.6 subordinate-printers-supported (1setOf uri)

This Printer attribute is REQUIRED if an implementation supports subordinate Printers (see section 4.2) and contains the URIs of the immediate subordinate Printer object(s) associated with this Printer object.

The precise format of the subordinate Printer URIs is implementation dependent (see section 4.2.4).

If the Printer object does not have an associated output device ~~Device object (see [ipp-set3])~~, the value of the subordinate Printer object's "printer-name" MAY be used to populate the Job object's "output-device-assigned" attribute (see [ipp-mod] section 4.3.13). The "output-device-assigned" Job attribute identifies the output device to which the Printer object has assigned a job, for example, when a single Printer object is supporting Device fan-out or Printer fan-out.

7 Additional Values for "printer-state-reasons"

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

~~1.1~~7.1 '~~deactivated~~~~standby~~'

~~Type of registration: type2 keyword attribute value~~
~~Name of attribute to which this keyword specification is to be added: printer-state-reasons~~
~~Proposed keyword name of this keyword value: standby~~
~~Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):~~

'~~deactivated~~~~standby~~': The Printer has been ~~deactivated~~~~shutdown in standby mode~~. Only the ~~Restart~~Activate-Printer, Get-Job-Attributes, Get-Jobs, ~~and~~ Get-Printer-Attributes, and any other Get-Xxx operations that may be defined are accepted in this state; all other operations are rejected with the 'server-error-printer-is-~~deactivated~~~~in standby mode~~'.

ISSUE 04 - What new 'moving-to-xxx' and 'xxx' values do we need to support the new operations defined in this document?

~~1.2~~7.2 'moving-to-paused-all'

'moving-to-paused-all': Someone has paused the Printer object using the Pause-Printer-After-All-Current-Jobs operation (see section 11.3.3) or other means, but the output-device(s) are taking an appreciable time to stop. Later, when all output has stopped, the "printer-state" becomes 'stopped', and the 'paused' value replaces the 'moving-to-paused' value in the "printer-state-reasons" attribute.

This value MUST be supported, if the Pause-Printer-After-All-Current-Jobs operation is supported and the implementation takes significant time to pause a device in certain circumstances.

~~1.3~~7.3 'printer-deactivated'

'printer-deactivated': Someone has issued a Deactivate-Printer operation for the Printer object (see section 11.4.1) and the Printer is in the process of becoming deactivated or has become deactivated. The Printer MUST reject all requests except Activate-Printer, 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 11.2.1) and return the 'server-error-service-unavailable' status code.

8 Additional Values for "job-state-reasons"

This section defines additional values for the "job-state-reasons" Job Description attribute.

~~1.1~~8.1 'job-suspended~~paused~~'

~~Type of registration: type2 keyword attribute value~~
~~Name of attribute to which this keyword specification is to be added: job-state-reasons~~
~~Proposed keyword name of this keyword value: job-paused~~
~~Specification of this keyword value (follow the style of IPP Model Section 4.1.2.3):~~

'job-suspended~~paused~~': The job has been suspended~~paused~~ while processing using the ~~Pause~~Suspend-Current-Job operation~~s~~ and other jobs can be processed on the Printer. The Job can be resumed using the Resume-Job operation which removes this value.

9 Additional status codes

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

~~1.1~~9.1 'client-error-attributes-not-settable' (0x0413)

~~Type of registration: status code~~
~~Keyword symbolic name of this status code value: 'client-error-attributes-not-settable'~~
~~Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):~~
~~Operations that this status code may be used with: Set-Printer-Attributes, Set-Job-Attributes~~
~~Specification of this status code (follow the style of IPP Model Section 13 APPENDIX B: Status Codes and Suggested Status Code Messages):~~

~~13.1.4.20 client-error-attributes-not-settable (0x0413)~~

In a Set-Printer-Attributes or Set-Job-Attributes request, if the Printer object does not support one or more attributes as settable, the Printer object MUST return this status code. The Printer object MUST also return in the Unsupported Attributes Group all the attributes and/or values supplied by the client that are not settable. See [ipp-mod] section 3.1.7. For example, if the request indicates 'job-state', all implementations MUST reject the request. As another example, if the request indicates an attribute that is supported, but not settable by this implementation, such as, say, "printer-name", the implementation rejects the request.

~~1.2~~9.2 'server-error-printer-is-deactivated~~in-standby-mode~~' (0x????)

~~Type of registration: status code~~
~~Keyword symbolic name of this status code value: 'server-error-printer-is-in-standby-mode'~~
~~Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):~~
~~Operations that this status code may be used with: Shutdown-Printer~~
~~Specification of this status code (follow the style of IPP Model Section 13 APPENDIX B: Status Codes and Suggested Status Code Messages):~~

~~13.1.5.8 server-error-printer-is-in-standby-mode~~

The Printer has been deactivated using the Deactivate-Printer operation~~shutdown~~ and is only accepting the ~~Restart~~Activate-Printer (see section 11.5.1), Get-Job-Attributes, Get-Jobs, ~~and~~ Get-Printer-Attributes, and any other Get-Xxxx operations. An operator can perform the Activate~~Restart~~-Printer operation to allow the Printer to accept other operations.

10 Additional 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.

~~1.1~~10.1 'not-settable' out of band value

The 'not-settable' out of band value is used by the IPP Printer in a Set-Job-Attributes or Set-Printer-Attributes response to indicate an attribute that was supplied, is supported but is defined to be a READ-ONLY attribute, is not settable in that implementation or as defined by the policy for that system, because it is not in the "job-settable-attributes"~~ on the one hand, or the "interpreter-settable-attributes"~~ or "printer-settable-attributes", respectively~~on the other hand~~.

11 Definition of the Set 2 Printer operations

All Printer 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 2 Printer operations are summarized in ~~the following table~~Table 3~~3~~:

Table 3~~3~~ - Printer operation Operation-Id assignments

Operation Name	Operation-Id	Brief description
Set-Printer-Attributes	0x??	Sets attribute values of the target Printer object
Enable-Printer	0x??	Allows the target Printer to accept ~~create job~~ Job Creation operations
Disable-Printer	0x??	Prevents the target Printer from accepting ~~create job~~Job Creation operations
Pause-Printer-After-All-Current-Jobs	0x??	Finishes processing all currently pending jobs. Any new jobs are placed in the 'pending-held' state.
Deactivate-Printer	0x??	Puts the Printer into a read-only deactivated state.
Activate-Printer	0x??	Restores the Printer to normal activity
Restart-Printer	0x??	Restarts the target Printer and re-initializes the software~~from a standby shutdown mode~~
Shutdown-Printer	0x??	Shuts down the target Printer ~~in several ways~~so that it cannot be restarted or queried
Startup-Printer	0x??	Starts up the instance of the Printer object

All of the operations in this document~~registration proposal specification~~ 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.

993

994 ~~1.1~~11.1 Set-Printer-Attributes Operation

995 ~~Type of registration: operation~~
996 ~~Proposed name of this operation: Set-Printer-Attributes~~
997 ~~Object Target: Printer~~
998 ~~Specification of this operation:~~

999 This OPTIONAL operation allows a client to set the values of the attributes of a Printer object. In the
1000 request, the client supplies the set of Printer attribute names and values that are to be set. In the response,
1001 the Printer object returns success or rejects the request with indications of which attribute or attributes
1002 could not be set.

1003 How the Printer object validates the client-supplied attributes in the Set-Printer-Attributes request is
1004 implementation-dependent, since there are no corresponding Printer attributes that specify the allowed
1005 values that may be set on the Printer object.

1006 The Printer MUST accept this operation in any state, i.e., for any of the values of the Printer object's
1007 "printer-state" attribute.

1008 ~~1.1.1~~11.1.1 Settable and READ-ONLY Printer Description attributes

1009 If the Printer supports the "printer-message-from-operator" Printer Description attribute (see [ipp-mod]
1010 section 4.4.25) and the client explicitly supplies a new value for the "printer-message-from-operator" in the
1011 request, then the Printer MUST set the "printer-message-from-operator" Printer attribute to this new value
1012 and MUST also set the "printer-message-time", "printer-message-date-time", and "printer-message-
1013 operation" attributes, if supported (see sections 6.3, 6.4, and 6.5).

1014 If the Printer supports the Set-Printer-Attributes operation, then it SHOULD support setting of:

1015 all Job Template Default ("xxx-default") attributes
1016 all Job Template Supported ("xxx-supported") attributes
1017 all Job Template Ready ("xxx-ready") attributes

1018 that the implementation supports (see [ipp-mod] section 4.2 and extensions).

1019 The following Printer Description attributes (see [ipp-mod] section 4.4) MUST NOT be settable, i.e., they
1020 are READ-ONLY:

printer-state
printer-state-reasons
printer-state-message
printer-is-accepting-jobs - see Enable-Printer/Disable-Printer
queued-job-count
~~printer-message-time - see Set-Printer-Attributes when setting "printer-message-from-operator"~~
~~printer-message-date-time - see Set-Printer-Attributes when setting "printer-message-from-operator"~~
~~printer-message-operation - see Set-Printer-Attributes when setting "printer-message-from-operator"~~
printer-up-time

The following Set2 Printer Description attributes are READ-ONLY:

printer-message-time - see Set-Printer-Attributes when setting "printer-message-from-operator"
printer-message-date-time - see Set-Printer-Attributes when setting "printer-message-from-operator"
printer-message-operation - see Set-Printer-Attributes when setting "printer-message-from-operator"

Note: From now on, all extensions that define new object attributes will indicate whether or not they are READ-ONLY, by including the "READ-ONLY" adjective in their descriptions. If "READ-ONLY" is omitted, it is assumed that the attribute MAY be settable by the appropriate Set-Xxx operation.

The remaining Printer Description attributes MAY be settable using the Set-Printer-Attributes operation, depending on implementation. If "xxx-supported" Printer Description attribute are settable, then they MUST affect the behavior of the implementation. If they are READ-ONLY then they reflect the implementation and cannot be changed using the Set-Printer-Attributes operation. Consider the following examples:

For example, if the "operations-supported" Printer Description attribute (see [ipp-mod] section 4.4.15) is settable in a particular implementation, then changing its value with a Set-Printer-Attributes operation MUST affect the operations that the implementation accepts or rejects. Such an implementation will need to be able to reject values for operations that it contains no code support for. If the "operations-supported" Printer Description attribute is not settable in a particular implementation, then that implementation MUST reject an attempt to set it with a Set-Printer-Attributes operation and return the 'client-error-attributes-not-settable' status code (see section 9.1).

As another example, if the "ipp-versions-supported" Printer Description attribute (see [ipp-mod] section 4.4.14) is settable in a particular implementation, then changing its value with a Set-Printer-Attributes operation MUST affect the protocol versions that are accepted or rejected. Such an implementation will need to be able to reject values for versions~~operations~~ that it contains no code support for.

See the "factory-settings" operation attribute (see section 5.3) for a way to query the implementation supported values using the Get-Printer-Attributes operation. ~~See the "reset-function" operation attribute of the Reset-Device operation (see [ipp-set3]) for a way to restore the values of the Device object to the factory settings.~~

~~ISSUE 07 - Ok not to have a way to reset the values of the Printer object to the factory settings, but only have a way to reset the factory default settings for the Device object?~~

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

Most Printer attributes will require administrator privileges to set, such as "xxx-supported", while some will require operator privileges only, such as "media-ready" and "printer-message-from-operator". Which attributes require which privileges depends on implementation and MAY depend on site policy.

~~1.1.2~~11.1.2 Set-Printer-Attributes Request

The following sets of attributes are part of the Set-Printer-Attributes Request:

Group 1: Operation Attributes

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod] section 3.1.4.1.

Target:
The "printer-uri" (uri) operation attribute which is the target for this operation as described in [ipp-mod] section 3.1.5.

Requesting User Name:
The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as described in [ipp-mod] section 8.3.

"document-format" (mimeMediaType):
The client SHOULD supply this attribute. The Printer object MUST support this attribute. This attribute is useful for a client to select the Interpreter object to which the attribute modification should be applied. Each Printer object is modeled to contain one or more Interpreter objects. Those Printer attributes whose values vary from Interpreter to Interpreter, are modeled as Interpreter objects, while those that do not are Printer object attributes. Thus the target of a Get-Printer-Attributes or Set-Printer-Attributes operation is either the Printer object or the Interpreter object identified by the "document-format" operation attribute supplied by the client. Except for Get-Printer-Attributes and Set-Printer-Attributes, there are no other operations with the Interpreter object as a target. See [ipp-mod] section 3.2.5.1 "Get-Printer-Attributes Request".

If a client wants to set an attribute for~~of~~ all of the Interpreter objects to the same value, it can query the Printer's "document-format-supported" Printer Description attribute and perform separate Set-Printer-Attributes for each document format supported.

If the Printer object does not distinguish between different sets of supported values for each different document format when validating jobs in the create and Validate-Job operations, it MUST NOT distinguish between different document formats in the Set-Printer-Attributes operation. If the Printer object does distinguish between different sets of supported values for each different

document format specified by the client, this specialization applies only to the same Printer attributes as the Get-Printer-Attributes operation (see [ipp-mod] section 3.2.5.1).

If the client omits this "document-format" operation attribute, the Printer object MUST respond as if the attribute had been supplied with the value of the Printer object's "document-format-default" attribute. It is recommended that the client always supply a value for "document-format", since the Printer object's "document-format-default" may be 'application/octet-stream', in which case the set attributes and values are for the union of the document formats that the Printer can automatically sense. For more details, see the description of the 'mimeMediaType' attribute syntax in [ipp-mod] section 4.1.9.

If the client supplies a value for the "document-format" Operation attribute that is not supported by the Printer, i.e., is not among the values of the Printer object's "document-format-supported" attribute, the Printer object MUST reject the operation and return the 'client-error-document-format-not-supported' status code.

Group 2: Printer Attributes

The client MUST supply a set of Printer attributes as defined in [ipp-mod] section 4.2 Job Template Attributes ("xxx-default", "xxx-supported", and "xxx-ready" attributes), ~~and~~ section 4.4 Printer Description Attributes, and any attribute extensions supported by the Printer. Each Printer attribute supplied in Group 2 replaces the value(s) of the corresponding Printer attribute on the target Printer object. If a Printer object attribute had not been configured yet and so had the 'no-value' out-of-band value (see [ipp-mod] 4.1), the supplied value(s) replace the 'no-value' value. For attributes that can have multiple values (1setOf), all values supplied by the client replace all values of the corresponding Printer object attribute.

~~1.1.3~~11.1.3 Set-Printer-Attributes Response

The Printer object returns the following sets of attributes as part of the Get-Printer-Attributes Response:

Group 1: Operation Attributes

Status Message:
In addition to the REQUIRED status code returned in every response, the response OPTIONALLY includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation attribute as described in [ipp-mod] sections 13 and 3.1.6.

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod] section 3.1.4.2.

Group 2: Unsupported Attributes

See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes.

In the case of attributes that are supported, but are not settable by the implementation, i.e., are not among the values of the Printer's "printer-settable-attributes" Printer Description attribute (see section 6.1), the Printer object returns the client-supplied attribute(s) with a substituted out-of-band value of 'not-settable~~supported~~' (see section 10.1)~~(same out-of-band value as for attributes that are not supported)~~. This value's syntax type is "out-of-band" and its encoding is defined by special rules for "out-of-band" values in the "Encoding and Transport" document [IPP-PRO]. Its value indicates that the attribute is supported, but is ~~either~~ not settable ~~or is not supported at all~~.

~~1.2~~11.2 The Disable and Enable Printer Operations

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

These operations allow the operator to control whether or not the Printer will accept new Job Creation (Print-Job, Print-URI, and Create-Job) operations. These operations have no other effect on the Printer, so that the Printer continues to accept all other operations and continues to schedule and process jobs normally. In other words, these operation control the "input of new jobs" to the IPP Printer while the Pause and Resume operations (see section 11.3) independently control the "output of new jobs" from the IPP Printer to the output device.

The Disable and Enable Printer operations MUST NOT affect the submission of jobs using other job submission protocols to the associated output device; the Disable and Enable Device operations (see [ipp-set3]) are intended to stop the acceptance of all jobs by the associated output device(s).

~~1.1.1~~11.2.1 Disable-Printer Operation

~~Type of registration: operation~~
~~Proposed name of this operation: Disable-Printer~~
~~Object Target: Printer~~
~~Specification of this operation:~~

This OPTIONAL operation allows a client to stop the Printer object from accepting new jobs, i.e., cause the Printer to reject subsequent ~~create job~~Job Creation operations ~~(Print-Job, Print-URI, and Create-Job operation)~~ and return the 'server-error-not-accepting-jobs' status code. The Printer still accepts all other operations, including Validate-Job, Send-Document and Send-URI operations. Thus a Disable-Printer operation allows a client to continue submitting multiple documents of a multiple document job if the Create-Job operation had already been accepted. All previously created or submitted Jobs and currently processing Jobs continue unaffected.

The IPP Printer MUST accept the request in any state. The Printer sets the value of its "printer-is-accepting-jobs" READ-ONLY Printer Description attribute to 'false' (see [ipp-mod] section 4.4.20), no matter what the previous value was. This operation has no immediate or direct effect on the Printer's "printer-state" and "printer-state-reasons" attributes.

~~Note: Use the Enable-Printer operation (section 10.1.3) to enable a Printer to accept Jobs again.~~

~~If the Disable-Printer operation is supported, then the Enable-Printer operation MUST be supported, and vice-versa.~~

~~Note: Use the Enable-Printer and Disable-Printer operations to allow or prevent input to a Printer. Use the Pause-Printer and Resume-Printer operations to prevent or allow output from the Printer.~~

~~Whether or not the Disable-Printer operation stops jobs that are submitted using job submission protocols other than 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. See "printer-controls-other-protocols" (section 6.3).~~

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

The Disable-Printer Request and Disable-Printer Response have the same attribute groups and attributes as the ~~Resume~~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 section 5.1)~~., and with the addition of the following Group 1 operation attribute in the request:~~

~~"job-type" (type2 keyword):~~

~~The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this attribute. The value of this attribute indicates the type of job to be disabled. If the client omits this attribute, the Printer assumes the 'network-jobs' value.~~

~~Standard keyword values are:~~

~~'network-jobs' - disable jobs submitted using the create job operations.~~
~~'walk-up-jobs' - disable jobs submitted locally, such as walkup copy jobs~~
~~'all-jobs' - disable all type of jobs~~

~~1.1.2~~11.2.2 Enable-Printer Operation

~~Type of registration: operation~~
~~Proposed name of this operation: Enable-Printer~~
~~Object Target: Printer~~
~~Specification of this operation:~~

This OPTIONAL operation allows a client to start the Printer object accepting jobs, i.e., cause the Printer to accept subsequent ~~create job~~Job Creation operations ~~(Print-Job, Print-URI, and Create-Job operation)~~. The Printer still accepts all other operations. All previously submitted Jobs and currently processing Jobs continue unaffected.

The IPP Printer MUST accept the request in any state. The Printer sets the value of its "printer-is-accepting-jobs" READ-ONLY Printer Description attribute to 'true' (see [ipp-mod] section 4.4.20), no matter what the previous value was. This operation has no immediate or direction effect on the Printer's "printer-state" and "printer-state-reasons" attributes.

~~Note: Use the Disable-Printer operation (section 10.1.2) to stop a Printer from accepting Jobs.~~

~~If the Enable-Printer operation is supported, then the Disable-Printer operation MUST be supported, and vice-versa.~~

~~Note: Use the Enable-Printer and Disable-Printer operations to allow or prevent input to a Printer. Use the Pause-Printer and Resume-Printer operations to prevent or allow output from the Printer.~~

~~Whether or not the Enable-Printer operation allows acceptance of jobs that are submitted using job submission protocols other than 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. See "printer-controls-other-protocols" (section 6.3).~~

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

The Enable-Printer Request and Enable-Printer Response have the same attribute groups and attributes as the Resume-Printer operation (see [ipp-mod] sections 3.2.8.1 and 3.2.8.2), including the new "printer-message-from-operator" operation attribute (see section 5.1)~~, and with the addition of the following Group 1 operation attribute in the request:~~

~~"job-type" (type2 keyword):~~
~~The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this attribute. The value of this attribute indicates the type of job to be enabled. If the client omits this attribute, the Printer assumes the 'network-jobs' value.~~

~~Standard keyword values are:~~
~~'network-jobs' - enable jobs submitted using the create-job operations.~~

~~'walk-up-jobs' - enable jobs submitted locally, such as walkup copy jobs~~
~~'all-jobs' - enable all types of jobs~~

~~1.3~~11.3 The Pause and Resume Printer operations

This section clarifies the OPTIONAL IPP/1.1 Pause-Printer (to be Pause-Printer-After-Current-Job) and Resume-Printer (see [ipp-mod] sections 3.2.7 and 3.2.8) and defines the OPTIONAL Pause-Printer-After-All-Current-Jobs operations. These operations affect the scheduling of IPP jobs. If either of the Pause operations are supported, then the Resume-Printer operation MUST be supported.

These operations allow the operator to control whether or not the Printer will send new IPP jobs to the associated output device(s) that the IPP Printer object represents. These operations have no other effect on the Printer, so that the Printer continues to accept all operations. In other words, these operation control the "output of new jobs" to the output device(s) while the Disable and Enable Printer operations (see section 11.2) independently control the "input of new jobs" to the IPP Printer.

The Pause and Resume Printer operations MUST NOT affect jobs that were submitted using other job submission protocols to the associated output device; the Pause and Resume Device operations (see [ipp-set3]) are intended to stop the acceptance of all jobs by the associated output device(s).

~~1.1.1~~11.3.1 IPP/1.1 Pause-Printer operation clarified to be Pause-Printer-After-Current-Job

IPP/1.1 defines the Pause-Printer operation (see [ipp-mod] section 3.2.7) with a number of implementation options:

> 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.
>
> If the Pause-Printer operation is supported, then the Resume-Printer operation MUST be supported, and vice-versa.
>
> The IPP Printer stops the current job(s) on its 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 [ipp-mod] 4.4.12). 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 (see [ipp-mod] 4.4.12) to the Printer object's "printer-state-reasons" attribute.

The Set2 and Set3 documents define distinct operations in order to disambiguate the Pause-Printer operation as shown in Table ~~4~~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 44 - 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 05 - 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?

11.3.2 Pause-Printer-After-Current-Job

This OPTIONAL operation allows a client to stop the Printer object from starting to send IPP jobs to any of its output devices or subordinate Printers. If the IPP Printer is in the middle of sending an IPP job to an output device or subordinate Printer, the IPP Printer MUST complete sending that Job. However, after receiving this operation, the IPP Printer MUST NOT start to send any additional IPP jobs to any of its output devices or subordinate Printers. In addition, after having received this operation, the IPP Printer MUST NOT start processing any more jobs, so additional jobs MUST NOT enter the 'processing' state.

If the IPP Printer is not sending an IPP Job to the output device or subordinate Printer (whether or not the output device or subordinate Printer is busy processing any jobs), the IPP 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 implementation will take appreciable time to complete sending an IPP job that it has started sending to an output device or subordinate Printer, the IPP Printer adds the 'moving-to-paused' value to the Printer object's "printer-state-reasons" attribute (see section [ipp-mod] 4.4.12). When the IPP Printer has completed sending IPP jobs that it was in the process of sending, 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 Job Creation requests (see Disable-Printer section 11.2.1).

For any jobs that are 'pending' or 'pending-held', the 'printer-stopped' value of the jobs' "job-state-reasons" attribute also applies. However, the IPP Printer NEED NOT update those jobs' "job-state-reasons" attributes and only need return the 'printer-stopped' value when those jobs are queried using the Get-Job-Attributes or Get-Jobs operations (so-called "lazy evaluation").

~~Whether the Pause-Printer operation affects jobs that were submitted to the device from other sources than the IPP Printer object in the same way that the Pause-Printer 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 IPP Printer MUST accept the request in any state and transition the Printer 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'	~~OPTION 1:~~ 'successful-ok'; Later, when ~~all output has stopped~~the IPP Printer has finished sending IPP jobs, the "printer-state" becomes 'stopped', and the 'paused' value replaces the 'moving-to-paused' value in the "printer-state-reasons" attribute
'processing'	'stopped'	'paused'	~~OPTION 2:~~ 'successful-ok'; the IPP Printer wasn't in the middle of sending an IPP job to the output device ~~all device output stopped immediately~~
'stopped'	'stopped'	'paused'	'successful-ok'

ISSUE 06 - Or should the Printer's "printer-state" attribute be independent of the Pause Printer operations so that the 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 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). Otherwise, the IPP Printer MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.~~

The Pause-Printer-After-Current-Job Request and Pause-Printer-After-Current-Job 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 section 5.1).

~~1.1.3~~11.3.3 Pause-Printer-After-All-Current-Jobs

ISSUE 07: Would a better name for Pause-Printer-After-All-Current-Jobs be Hold-Future-Jobs? Unfortunately, unlike Pause-Printer-After-All-Current-Jobs which gets to 'paused', the state transition would just be to 'idle' when all of the current jobs have completed? But what operation would undo this condition? Do-Not-Hold-Future-Jobs, Release-All-Jobs? Or how about having a single Schedule-Jobs operation that has a parameter that says whether to hold all future jobs or not?

This OPTIONAL operation allows a client to complete the current 'pending' IPP Jobs but not start processing any subsequently received IPP Jobs. If the IPP Printer is in the middle of sending an IPP job to an output device or subordinate Printer, the IPP Printer MUST complete sending that Job. Furthermore, the IPP Printer MUST send all of the current 'pending' IPP Jobs to the output device(s) or subordinate IPP Printer object(s). Any subsequently received Job Creation operations will cause the IPP Printer to put the Job into the 'pending-held' state until the Printer is resumed using the Resume-Printer operation.

If the IPP Printer has no 'pending' IPP Jobs and is not sending an IPP Job to an output device or subordinate Printer (whether or not the output device or subordinate Printer is busy processing any jobs), the IPP Printer object transitions immediately to the 'stopped' state by setting its "printer-state" attribute to 'stopped', removing the 'moving-to-paused-all' value, if present, from its "printer-state-reasons" attribute, and adding the 'paused' value to its "printer-state-reasons" attribute.

ISSUE 08: Any better name than 'moving-to-paused-all' Printer state reason to distinguish Pause-Printer-After-All-Current-Jobs from Pause-Printer-After-Current-Job which uses 'moving-to-paused'?

If the IPP Printer has 'pending' jobs or the implementation will take appreciable time to complete sending an IPP job that it has started sending to an output device or subordinate Printer, the IPP Printer adds the 'moving-to-paused-all' value to the Printer object's "printer-state-reasons" attribute (see section [ipp-mod] 4.4.12). When the IPP Printer has completed sending IPP jobs that it was in the process of sending and all its 'pending' jobs, the Printer object transitions to the 'stopped' state by setting its "printer-state" attribute to 'stopped', removing the 'moving-to-paused-all' 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 Job Creation requests (see Disable-Printer section 11.2.1), except to put the Jobs into the 'pending-held' state, instead of the 'pending' or 'processing' state.

For any jobs that are 'pending' or 'pending-held', the 'printer-stopped' value of the jobs' "job-state-reasons" attribute also applies. However, the IPP Printer NEED NOT update those jobs' "job-state-reasons" attributes and only need return the 'printer-stopped' value when those jobs are queried using the Get-Job-Attributes or Get-Jobs operations (so-called "lazy evaluation").

The IPP Printer MUST accept the request in any state and transition the Printer 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-all'	'successful-ok'; Later, when the IPP Printer has finished sending IPP jobs, the "printer-state" becomes 'stopped', and the 'paused' value replaces the 'moving-to-paused-all' value in the "printer-state-reasons" attribute
'processing'	'stopped'	'paused'	'successful-ok'; the IPP Printer didn't have any 'pending' jobs and wasn't in the middle of sending an IPP job to the output device
'stopped'	'stopped'	'paused'	'successful-ok'

ISSUE 09: Or should the Printer's "printer-state" attribute be independent of the Pause Printer operations so that the 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-Printer-After-All-Current-Jobs Request and Pause-Printer-After-All-Current-Jobs 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 section 5.1).

~~1.4~~11.4 Deactivate and Activate Printer operations

This section defines the OPTIONAL Deactivate-Printer and Activate-Printer operations that stop and start the IPP Printer object from accepting all requests except queries and performing work. If either of these operations are supported, both MUST be supported.

These operations allow the operator to put the Printer into a dormant read-only condition and to take it out of such a condition. These operations are a combination of the Deactivate and Pause operations, plus preventing the acceptance of any other requests, except queries.

The Deactivate and Activate Printer operations MUST NOT affect the submission of jobs using other job submission protocols to the associated output device; the Deactivate and Activate Device operations (see [ipp-set3]) are intended to stop the associated output device(s) from performing work and accepting operations, except query operations.

11.4.1 Deactivate-Printer operation

This OPTIONAL operation allows a client to stop the Printer object from starting to send IPP jobs to any of its output devices or subordinate Printers (Pause-Printer-After-Current-Job) and stop the Printer object from accepting any, but query requests. The Printer performs a Disable-Printer and a Pause-Printer-After-Current-Job operation immediately, including use of all of the "printer-state-reasons" if these two operations cannot be completed immediately. In addition, the Printer MUST immediately reject all requests, except Activate-Printer, 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 11.2.1) and return the 'server-error-service-unavailable' status code.

The IPP Printer MUST accept the request in any state. Immediately, the Printer MUST set the 'printer-deactivated' value 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-Printer Request and Deactivate-Printer 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 section 5.1).

~~1.1.2~~11.4.2 Activate-Printer operation

This OPTIONAL operation allows a client to undo the effects of the Deactivate-Printer, i.e., allow the Printer object to start sending IPP jobs to any of its output devices or subordinate Printers (Pause-Printer-After-Current-Job) and start the Printer object from accepting any requests. The Printer performs an Enable-Printer and a Resume-Printer operation immediately. In addition, the Printer MUST immediately start accepting all requests.

The IPP Printer MUST accept the request in any state. Immediately, the Printer MUST immediately remove the 'printer-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-Printer Request and Activate-Printer 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 section 5.1).

~~1.5~~11.5 Restart-Printer, Shutdown-Printer, and Startup-Printer operations

This section defines the OPTIONAL Restart-Printer, Shutdown-Printer, and Startup-Printer operations that initialize, shutdown, and startup the Printer object, respectively. Each of these operations is OPTIONAL and any combination MAY be supported.

The Restart-Printer, Shutdown-Printer, and Startup-Printer operations MUST NOT affect the submission of jobs using other job submission protocols to the associated output device; the Reset-Device and Power-Off-Device operations (see [ipp-set3]) are intended to initialize or power off the associated output device(s).

~~11.2.2.3~~11.5.1 Restart-Printer operation

~~Type of registration: operation~~
~~Proposed name of this operation: Restart-Printer~~
~~Object Target: Printer~~
~~Specification of this operation:~~

This OPTIONAL operation allows a client to restart a Printer object whose operation is in need of initialization because of incorrect or erratic behavior, i.e., perform the effect of a software re-boot. The implementation MUST attempt to save any information about Jobs and the Printer object before re-initializing. However, this operation MAY have drastic consequences on the running system, so the operator should first try the Deactivate-Printer to minimize the effect on the current state of the system. The effects of previous Disable-Printer, Pause Printer, and Deactivate-Printer operations are lost.

The IPP Printer MUST accept the request in any state. The Printer object MUST initialize its Printer's "printer-state" to 'idle', remove the state reasons from its "printer-state-reasons" attribute, and its "printer-is-accepting-jobs" attribute to 'true'.~~that has previously been shutdown in standby mode (see section 11.5.2)~~.
~~Standby mode is indicated by the Printer's "printer-state" being 'stopped' and its "printer-state-reasons" including the 'standby' and 'paused' values. If the Printer is not in standby mode, the Printer MUST reject this operation and return the 'client-error-not-possible' status code.~~

~~ISSUE 08: 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'.~~

~~DENVER MEETING> There was significant confusion as to the meaning of shutdown. If a shutdown operation halts communications to the printer a restart is not possible. It was agreed to include restart attributes 'restart-to-standby' (requires a two step process), 'restart-fully', and 'restart-sync'.~~

~~ACTION ITEM (Harry Lewis): make a complete proposal and submit for approval.~~

~~If the Restart-Printer operation is supported, then the Shutdown-Printer operation MUST be supported, since the Restart-Printer operation is meaningful only after a Shutdown-Printer operation has been performed. However, if the Shutdown-Printer operation is supported, the Restart-Printer NEED NOT be supported.~~

~~Issue 09: Why? This is backward, if you ask me (HRL).~~

~~Denver meeting> Consider "shutdown-attributes" similar to Windows "shutdown". Like:~~

- ~~Completely power off (walk to printer to bring it back up~~
- ~~Power off but listen for "restart" - shutdown to standby~~

~~Denver meeting> Also consider restart attributes.~~

- ~~Restart to Standby (disabled and paused) (two step bring up)~~
- ~~Restart (fully).~~
- ~~Restart sych~~

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

The Restart-Printer Request and Restart-Printer Response have the same attribute groups and attributes as the ~~Resume~~Pause-Printer operation (see [ipp-mod] sections 3.2.8.1 and 3.2.8.2), including the new "printer-message-from-operator" operation attribute (see section 5.1). ~~and the following Group 1 operation attribute:~~

~~11.2.2.4~~11.5.2 Shutdown-Printer Operation

~~Type of registration: operation~~
~~Proposed name of this operation: Shutdown-Printer~~
~~Object Target: Printer~~
~~Specification of this operation:~~

This OPTIONAL operation allows a client to shutdown a Printer, i.e., stop processing jobs and make the Printer object no longer available for any operations using the IPP protocol~~power off in some implementation-dependent manner~~. There is no way to bring the instance of the Printer object back to being used, except for the Startup-Printer (see section 11.5.3) which starts up a new instance of the Printer object for hosted implementations. The purpose of Shutdown-Printer is to shutdown the Printer for an extended period, not to reset the device(s) or modify a Printer attribute. See Restart-Printer (section 11.5.1), Startup-Printer (section), and Reset-Device [ipp-set3] for the way to initialize the software or reset the output device(s). See the Disable-Printer operation (section 11.2) for a way for the client to stop the Printer from accepting Job Creation requests without stopping processing or shutting down.

The Printer ~~is disabled immediately (see the Disable-Printer operation in section 11.2). The Printer~~ adds the 'shutdown' value (see [ipp-mod] section 4.4.11) immediately to its "printer-state-reasons" Printer Description attribute and performs a Deactivate-Printer operation (see section 11.4.1). ~~The "when" operation attribute specifies how much processing occurs before the Printer is paused (see [ipp-mod] section 3.2.7) and the shutdown is complete. All other requests continue to be accepted until the printer is powered down.~~

~~ISSUE 10 - 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.~~

~~HRL> Is this granularity really achievable enough of a wide enough variety of environments to be reliable or, in reality, will this be implementation dependent?~~

~~Whether or not the Shutdown-Printer operation affects jobs that were submitted to the device using job submission protocols other than 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. See "printer-controls-other-protocols" (section 1.1).~~

The Printer object MUST accept this operation in any state and transition the Printer object through the "printer-states" and "printer-state-reasons" defined for the Deactivate-Printer operation until the activity is completed and the Printer object disappears. ~~to the 'idle' state~~.

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

The Shutdown-Printer Request and Shutdown-Printer 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 section 5.1).~~, and with the addition of the following Group 1 operation attributes in the request:~~

~~"when" (type2 keyword)~~

~~The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this attribute, if it supports this operation. The value of this attribute indicates when to pause the printer. If the client omits this attribute, the Printer assumes the 'after-current-job' value. The 'after-current-job' value is REQUIRED to be supported if the "when" attribute is supported; the remaining values are OPTIONAL.~~

~~Note: There is no way to query to see which values of the "when" attribute are supported for this or any other operation. Since this operation is intended for Operators, rather than End-Users, it did not seem necessary to provide the means to query the supported values.~~

~~Standard keyword values are:~~

~~'now' - cancel the currently printing job(s) and shutdown the Printer. Jobs in the 'held' and 'pending' state remain in those states.~~

~~'after-current-job' - shutdown the Printer after the current job finishes printing (all its copies). Jobs in the 'held' and 'pending' state remain in those states.~~

~~'after-all' - shutdown the Printer after all 'pending' jobs finish printing. Jobs in the 'held' state remain in the 'held' state.~~

~~ISSUE 11 - Ok to remove the 'after-current-copy' value from Shutdown-Printer?~~

~~ISSUE 12 - Ok to have removed the "shutdown-function" operation attribute with values 'stand-by' vs. 'power-off' Shutdown-Printer on the grounds that for the Printer object they have no meaning. Power-off is for the Shutdown-Device operation.~~

~~"synchronize" (boolean)~~

~~The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this attribute. This attribute indicates whether or not the printer is to synchronize the checkpoint data for the current job ("when" = 'now') with the pages that have actually printed. If the value of the "when" attribute is not 'now' or the "when" attribute is not supplied, then the "synchronize" attribute has no meaning and the Printer MUST ignore it. If this attribute is supported, then a value of 'true' implies that the Printer will be able to resume the job at the point of synchronization when the Printer is restarted. If the implementation does not support resuming a job (either automatically or with the Resume-Job operation) after a Shutdown-Printer with "when" = 'now', then it does not implement this attribute. In this case, check-pointing implies that the job may be resumed in the future, exactly from the point and in the state and resource context from which it left off.~~

~~If the Printer supports this attribute but the client does not supply it, the Printer is assumed to perform synchronization ('true' behavior). If the Printer does not support this attribute, the Printer is assumed to not synchronize ('false' behavior).~~

~~ISSUE 13: Is the current job automatically restarted when the Printer is restarted? Or does some client have to issue a Restart-Job operation?~~

11.5.3 Startup-Printer operation

This OPTIONAL operation allows a client to startup an instance of a Printer object, provided that there isn't already one instantiated. The purpose of Startup-Printer is to allow a hosted implementation of the IPP Printer object to be started after the host is available (by means outside this document). See Restart-Printer (section 11.5.1) and Reset-Device [ipp-set3] for the way to initialize the software or reset the output device(s) when the IPP Printer object has already been instantiated.

The host MUST accept this operation only when the Printer object has not been instantiated. If the Printer object already exists, the host must return the 'client-error-not-possible' status code.

The result of this operation MUST be with the Printer object's "printer-state" set to 'idle', the state reasons removed from its "printer-state-reasons" attribute, and its "printer-is-accepting-jobs" attribute set to 'true'.

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 Shutdown-Printer Request and Shutdown-Printer 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 section 5.1).

12 Definition of the Set2 Job Operations

All Job operations are directed at Job objects. A client MUST always supply some means of identifying the Job object in order to identify the correct target of the operation. That job identification MAY either be a single Job URI or a combination of a Printer URI with a Job ID. The IPP object implementation MUST support both forms of identification for every job.

The Set2 Job Operations are summarized in Table 5~~Table 5~~:

Table 5~~5~~ - Job operation Operation-Id assignments

Operation Name	Operation-Id	Brief description
Set-Job-Attributes	0x??	Sets attribute values of the target Job object
Reprocess-Job	0x??	Creates a copy of a completed target job with a new Job ID and processes it
Cancel-Current-Job	0x??	Cancels the current job on the target Printer or the specified job if it is the current job
~~Pause~~Suspend-Current-Job	0x??	~~Pauses~~ Suspends the current processing job on the target Printer or the specified job if it is the current job, allowing other jobs to be processed instead
Resume-Job	0x??	Resume the paused target job
Promote-Job	0x??	Promote the pending target job to be next after the current job(s) complete

~~1.1~~12.1 Set-Job-Attributes

~~Type of registration: operation~~
~~Proposed name of this operation: Set-Job-Attributes~~
~~Object Target: Job~~
~~Specification of this operation:~~

This OPTIONAL operation allows a client to set the values of the attributes of a Job object. In the request, the client supplies the set of Job attribute names and values that are to be set. In the response, the IPP object returns success or rejects the request with indications of which attribute or attributes could not be set.

This operation is almost identical to the Set-Printer-Attributes operation (see section 11.1). The only differences are that the Set-Job-Attributes operation is directed at a Job object rather than a Printer object, there is no "document-format" operation attribute used when setting a Job object, and the validation is the same as the create job operations, i.e., depends on the "xxx-supported" Printer Description attributes.

The validation of the Set-Job-Attributes request is performed as if the job had been submitted originally with the new values and with "ipp-attribute-fidelity" set to 'true', i.e., all modified attributes MUST be supported along with the attributes not modified. If such a create job operation would have been accepted, then the Set-Job-Attributes MUST be accepted. If such a create job operation would have been rejected, then the Set-Job-Attributes MUST be rejected and the Job MUST be unchanged.

The IPP object MUST accept or reject the request based on the job's current state and transition the job to the indicated new state as follows:

Current "job-state"	New "job-state"	IPP object's response status code and action:
'pending'	'pending'	'successful-ok'
'pending'	'pending-held'	'successful-ok' - needed resources are not ready
'pending-held'	'pending-held'	'successful-ok'
'pending-held'	'pending'	'successful-ok' - needed resources are ready
'processing'	'processing'	'successful-ok' or 'client-error-not-possible' depending on the attributes being set, whether the job has started marking media, and/or implementation
'processing-stopped'	'processing-stopped'	'successful-ok' or 'client-error-not-possible' depending on the attributes being set, whether the job has started marking media, and/or implementation
'completed'	'completed'	'client-error-not-possible'
'canceled'	'canceled'	'client-error-not-possible'
'aborted'	'aborted'	'client-error-not-possible'

~~1.1.1~~12.1.1 Settable and READ-ONLY Job Description attributes

If the Printer supports the "job-message-from-operator" Job Description attribute (see [ipp-mod] section 4.3.16) and the client explicitly supplies a new value for the "job-message-from-operator" in the request, then the Printer MUST set the "job-message-from-operator" Job attribute to this new value.

If the Printer supports the Set-Job-Attributes operation, then it SHOULD support setting of:

all Job Template ("xxx") attributes (see [ipp-mod] section 4.2 and extensions)

that the implementation supports.

The following Job Description attributes (see [ipp-mod] section 4.3) MUST NOT be settable, i.e., they are READ-ONLY:

job-uri
job-id
job-printer-uri
job-more-info
job-originating-user-name - set in create operation
job-state
job-state-reasons
job-state-message
number-of-documents
output-device-assigned
time-at-creation
time-at-processing
time-at-completed
job-printer-up-time
date-time-at-creation
date-time-at-processing
date-time-at-completed
number-of-intervening-jobs
job-k-octets
job-k-octets-processed
job-impressions-completed
job-media-sheets-completed
attributes-charset - set in Job Creation operation
attributes-natural-language - set in Job Creation operation

The following Set2 Job Description attributes are READ-ONLY:

original-requesting-user-name - set in Job Creation operation

Note: From now on, all extensions that define new object attributes will indicate whether or not they are READ-ONLY, by including the "READ-ONLY" adjective in their descriptions. If "READ-ONLY" is omitted, it is assumed that the attribute MAY be settable by the appropriate Set-Xxx operation.

The remaining Job Description attributes MAY be settable using the Set-Job-Attributes operation, depending on implementation.

~~Whether or not the Set-Job-Attributes operation affects jobs that are submitted using job submission protocols other than 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. See "printer-controls-other-protocols" (section 1.1).~~

Access Rights: The authenticated user (see [ipp-mod] section 8.3) performing this operation must either be the job owner (as determined in the Job Creation operation) or an operator or administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5). ~~Authentication and access control (see [ipp-mod] sections 1, 8.3, and 8.5) apply to this operation.~~

~~1.1.2~~12.1.2 Set-Job-Attributes Request

The following sets of attributes are part of the Set-Job-Attributes Request:

Group 1: Operation Attributes

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod] section 3.1.4.1.

Target:
Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation attribute(s) which define the target for this operation as described in [ipp-mod] section 3.1.5.

Requesting User Name:
The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as described in [ipp-mod] section 8.3.

Group 2: Job Attributes

The client MUST supply a set of Job attributes as defined in [ipp-mod] section 4.2 Job Template Attributes ("xxx" attributes), ~~and~~ section 4.3 Job Description Attributes, and any attribute extensions supported by the Printer. Each Job attribute supplied in Group 2 replaces the value(s) of the corresponding Job attribute on the target Job object. For attributes that can have multiple values (1setOf), all values supplied by the client replace all values of the corresponding Job object attribute.

~~1.1.3~~12.1.3 Set-Job-Attributes Response

The Printer object returns the following sets of attributes as part of the Set-Job-Attributes Response:

Group 1: Operation Attributes

Status Message:
In addition to the REQUIRED status code returned in every response, the response OPTIONALLY includes a "status-message" (text(255)) and/or a "detailed-status-message" (text(MAX)) operation attribute as described in [ipp-mod] sections 13 and 3.1.6.

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as described in [ipp-mod] section 3.1.4.2.

Group 2: Unsupported Attributes

See [ipp-mod] section 3.1.7 for details on returning Unsupported Attributes.

~~The attributes returned are the same as for the create operation with the same new attribute values.~~
In the case of attributes that are supported, but are not settable by the implementation, i.e., are not among the values of the Printer's "job-settable-attributes" Printer Description attribute (see section

6.1), the Printer object returns the client-supplied attribute(s) with a substituted out-of-band value of 'not-settable~~supported~~' (see section 10.1)~~(same out-of-band value as for attributes that are not supported)~~. This value's syntax type is "out-of-band" and its encoding is defined by special rules for "out-of-band" values in the "Encoding and Transport" document [IPP-PRO]. Its value indicates that the attribute is supported, but is ~~either~~ not settable ~~or is not supported at all~~.

~~1.2~~12.2 Reprocess-Job Operation

This OPTIONAL operation is a create job operation that allows a client to re-process a copy of a job that had been retained in the queue after processing completed, was canceled, or was aborted (see [ipp-mod] section 4.3.7.2). This operation is the same as the Restart-Job operation (see [ipp-mod] section 3.3.7), except that the Printer creates a new job that is a copy of the target job and the target job is unchanged. The new job is assigned new values to the "job-uri" and "job-id" attributes and the new job's Job Description attributes that accumulate job progress, such as "job-impressions-completed", "job-media-sheets-completed", and "job-k-octets-processed", are initialized to 0 as with any create job operation. The target job moves to the Job History after a suitable period, independent of whether one or more Reprocess-Job operations have been performed on it.

If the Set-Job-Attributes operation is supported, then the "job-hold-until" operation attribute MUST be supported with at least the 'indefinite' value, so that a client can modify the new job before it is scheduled for processing using the Set-Job-Attributes operation. After modifying the job, the client can release the job for processing, by using the Release-Job operation specifying the newly assigned "job-uri" or "job-id" for the new job.

~~1.3~~12.3 Cancel-Current-Job Operation

This OPTIONAL operation allows a client to cancel the current job on the target Printer or the specified job if it is the current job on the Printer. See [ipp-mod] section 3.3.3 for the semantics of canceling a job. Since a Job might already be marking by the time a Cancel-Current-Job is received, some media sheet pages might be printed before the job is actually terminated.

ISSUE 10: Should Cancel-Current-Job really be moved to Set3 as a Device operation, i.e., Cancel-Current-Device-Job or do we need both a Printer and a Device operation that cancels the current job?

If the client does not supply a "job-id" operation attribute, the Printer MUST accept the request and cancel the current job if there is a current job in the 'processing' or 'processing-stopped' state; otherwise, it MUST reject the request and return the 'client-error-not-possible' status code. If more than one job is in the 'processing' or 'processing-stopped' states, the one that is marking is canceled and the others are unaffected.

Warning: On a shared printer, there is a race condition. Between the time that a user issues this operation and its acceptance, the current job might change to a different job. If the user or operator is authenticated to cancel the new job, the wrong job is canceled. To prevent this race from canceling the wrong job, the client MAY supply the "job-id" operation attribute which is checked against the current job's job-id. If the job identified by the "job-id" attribute is not the current job on the Printer, i.e., is not in the 'processing' or 'processing-stopped' states, the Printer MUST reject this operation and return the 'client-error-not-possible' status code. Otherwise, the Printer cancels the specified job.

Access Rights: The authenticated user (see [ipp-mod] section 8.3) performing this operation must either be the job owner (as determined in the Job Creation operation) or an operator or administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5). ~~Authentication and access control (see [ipp-mod] sections 1, 8.3, and 8.5) apply to this operation.~~

The Cancel-Current-Job Request and Cancel-Current-Job Response have the same attribute groups and attributes as the Resume-Printer operation (see [ipp-mod] section 3.2.8), including the new "job-message-from-operator" operation attribute (see section 5.2), with the addition of the following Group 1 Operation attributes in the request:

"job-id" (integer(1:MAX)):

The client OPTIONALLY supplies this Operation attribute in order to verify that the identified job is still the current job on the target Printer object. The IPP object MUST supports this operation attribute, if it supports this operation.

~~1.4~~12.4 Suspend and Resume Job operations

This section defines the Suspend-Current-Job and Resume-Job operations. These operations allow an operator or user to suspend a job while it is processing and allow other jobs to be processed and the resume the suspended job at a later point in time without losing any of the output.

If either of these operations is supported, they both MUST be supported.

The Hold-Job and Release-Job operations ([ipp-mod] section 3.3.5) are for holding and releasing held jobs, not suspending and resuming suspended jobs.

~~1.1.1~~12.4.1 Suspend~~Pause~~-Current-Job operation

~~ISSUE 14 - Since the semantics of Pause-Current-Job is different from Pause-Printer, in that other jobs are processed, while this job is stopped, should we have a different name for the verb, such as Suspend-Current-Job, instead of Pause-Current-Job. Otherwise, users may be mistakenly think that the printer is paused when the job is paused. If the name is changes, then the 'job-paused' job state reason would also be changed to 'job-suspended'.~~

This OPTIONAL operation allows a client to stop the current job on the target Printer or the specified job if it is the current job on the Printer, and allow other jobs to be processed instead. The Printer moves the current job or the target job to the 'processing-stopped' state and sets the 'job-~~paused'~~ suspended' value (see section 8.1) in the job's "job-state-reasons" attribute and processes other jobs.

If the client does not supply a "job-id" operation attribute, the Printer MUST accept the request and ~~pause~~ suspend the current job if there is a current job in the 'processing' or 'processing-stopped' state; otherwise, it MUST reject the request and return the 'client-error-not-possible' status code. If more than one job is in the 'processing' or 'processing-stopped' states, all of them are ~~paused~~suspended.

Warning: On a shared printer, there is a race condition. Between the time that a user issues this operation and its acceptance, the current job might change to a different job. If the user or operator is authenticated to ~~pause~~suspend the new job, the wrong job is ~~paused~~suspended. To prevent this race from pausing the wrong job, the client MAY supply the "job-id" operation attribute which is checked against the current job's job-id. If the job identified by the "job-id" attribute is not the current job on the Printer, i.e., is not in the 'processing' or 'processing-stopped' states, the Printer MUST reject this operation and return the 'client-error-not-possible' status code. Otherwise, the Printer ~~pause~~suspends the specified job and processed other jobs.

~~A paused job is resumed using the Resume-Job operation (see section 12.5). If the Pause-Current-Job operation is supported, then the Resume-Job operation MUST be supported, and vice-versa.~~

The Printer MUST reject a ~~Release~~Resume-Job request (and return the 'client-error-not-possible') for a job that has been ~~pause~~suspended , i.e., for a job in the 'processing-stopped' state, with the 'job-~~pause~~suspended'

value in its "job-state-reasons" attribute. ~~The Hold-Job and Release-Job operations are for holding and releasing held jobs, not pausing and resuming paused jobs.~~

~~ISSUE 15 - Ok for Pause-Job to reject a request for a Job that is already paused (like Cancel-Job), instead of accepting the request?~~

Access Rights: The authenticated user (see [ipp-mod] section 8.3) performing this operation must either be the job owner (as determined in the Job Creation operation) or an operator or administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5). ~~Authentication and access control (see [ipp-mod] sections 1, 8.3, and 8.5) apply to this operation.~~

The ~~Pause~~Suspend-Current-Job Request and ~~Pause~~Suspend-Current-Job Response have the same attribute groups and attributes as the ~~Resume~~Pause-Printer operation (see [ipp-mod] section 3.2.8), including the new "job-message-from-operator" operation attribute (see section 5.2), with the addition of the following Group 1 Operation attributes in the request:

"job-id" (integer(1:MAX)):

The client OPTIONALLY supplies this Operation attribute in order to verify that the identified job is still the current job on the target Printer object. The IPP object MUST supports this operation attribute, if it supports this operation.

~~"when" (type2 keyword)~~

~~The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this attribute, if it supports this operation. The value of this attribute indicates when to pause the printer. If the client omits this attribute, the Printer assumes the 'after-current-copy' value. The 'after-current-copy' value is REQUIRED to be supported if the "when" attribute is supported; the remaining values are OPTIONAL.~~

~~Note: There is no way to query to see which values of the "when" attribute are supported for this or any other operation. Since this operation is intended for Operators, rather than End-Users, it did not seem necessary to provide the means to query the supported values.~~

~~Standard keyword values are:~~

~~'now' - pause the current Job immediately, while the current Job is processing.~~

~~'after-current-copy' - pause the current Job after the current Job finishes printing its current copy.~~

~~ISSUE 16 - Ok that the only values for the "when" operation attribute for the Pause-Current-Job are 'now' and 'after-current-copy' with 'now' being the default and REQUIRED?~~

~~"synchronize" (boolean)~~

~~The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports this attribute. This attribute indicates whether or not the printer is to synchronize the checkpoint data for the current job being paused with the pages that have actually printed. If this attribute is supported, then a value of 'true' implies that the Printer will be able to resume the job at the point of synchronization when the job is resumed.~~

~~If the Printer supports this attribute but the client does not supply it, the Printer is assumed to perform synchronization ('true' behavior). If the Printer does not support this attribute, the Printer is assumed to not synchronize ('false' behavior).~~

~~1.1.2~~12.4.2 Resume-Job operation

This OPTIONAL operation allows a client to resume the target job at the point where it was ~~paused~~suspended. The Printer moves the target job to the 'pending' state and removes the 'job-~~paused~~ syspended' value from the job's "job-state-reasons" attribute.

If the target job is not in the 'processing-stopped' state with the 'job-~~paused'~~ suspended' value in the job's "job-state-reasons" attribute, the Printer rejects the request and returns the 'client-error-not-possible' status code, since the job was not ~~paused~~suspended.

~~ISSUE 17 - Ok to return an error (like Cancel-Job), rather than having Resume-Job accept the request even if the Job is already resumed (i.e., not paused)?~~

~~If the Resume-Job operation is supported, then the Pause-Current-Job operation MUST be supported, and vice-versa.~~

Access Rights: The authenticated user (see [ipp-mod] section 8.3) performing this operation must either be the job owner (as determined in the Job Creation operation) or an operator or administrator of the Printer object (see [ipp-mod] Sections 1 and 8.5). ~~Authentication and access control (see [ipp-mod] sections 1, 8.3, and 8.5) apply to this operation.~~

The Resume-Job Request and Resume-Job Response have the same attribute groups and attributes as the Release-Job operation (see [ipp-mod] section 3.3.6), including the new "job-message-from-operator" operation attribute (see section 5.2).

~~1.5~~12.5 Promote-Job operation

This OPTIONAL operation allows a client to make the pending target job be processed next after the current job completes. This operation is specially useful in a production printing environment where the operator is involved in job scheduling.

If the target job is in the 'pending' state, this operation does not change the job's state, but causes the job to be processed after the current job(s) complete. If the target job is not in the 'pending' state, the Printer rejects the request and returns the 'client-error-not-possible' status code. The Printer returns the target job immediately after the current job(s) in a Get-Jobs response (see [ipp-mod] section 3.2.6) for the 'not-completed' jobs.

When the current job completes, is canceled, ~~paused~~suspended, or aborted, the target of this operation is processed next.

If a client issues this request (again) before the target of the operation of the original request started processing, the target of this new request is scheduled before the previous job that was to be processed next.

~~Note:~~ IPP is specified not to require queues for job scheduling, since there are other implementation techniques for scheduling multiple jobs, such as re-evaluating a criteria function for each job on a scheduling cycle. However, if an implementation does implement queues for jobs, then the Promote-Job puts the specified job at the front of the queue. A subsequent Promote-Job before the first job starts processing puts that specified job at the front of the queue, so that it is "in front" of the previously promoted job.

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

The Promote-Job Request and Promote-Job Response have the same attribute groups and attributes as the Cancel-Job operation (see [ipp-mod] section 3.3.3), including the new "job-message-from-operator" operation attribute (see section 5.2).

13 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

14 Internationalization Considerations

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

15 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.

16 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

17 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.

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

18 Change History

This section summarizes the changes. Each sub-section is in reverse chronological order. Adding or removing ISSUES that don't change the document are not listed here.

~~1.1~~18.1 Changes to the November 16, 1999 version to make the December 8, 1999 version

The following changes to the November 16, 1999 version to make the December 8, 1999 version as a result of the IPP WG telecons and mailing list discussion:

1. Introduced the separation of Printer operation from Device operations. Removed the "printer-controls-other-protocols" (boolean) Printer Description attribute. Printer operations affect only IPP jobs and objects, while the Device operations affect the output device. Set2 has the Printer operations and Set3 has the Device operations. But do both sets of operations with only the Printer object and only the "printer-uri" target.

2. Remove the "when" operation attribute and added distinct Pause operations instead: Pause-Printer-After-Current-Job (IPP/1.1 Pause-Printer clarified), Pause-Printer-After-All-Current-Jobs

3. Added Deactivate-Printer and Activate-Printer which do Disable-Printer, Pause-Printer-After-Current-Job, and only allow query, Send-Document, Send-URI, and Activate-Printer operations. This is a clearer "shutdown" that can be brought back up using the protocol.

4. Clarified that Shutdown-Printer cannot be brought back via the protocol, though added Startup-Printer for hosted implementations to instantiate a fresh copy of the Printer object.

5. Changed the name of Pause-Current-Job to Suspend-Current-Job, since other jobs can be processed on the Printer (unlike Pause-Printer).

6. Added the Terminology section

7. Added the Requirements and Use Cases section

8. Added pictures of chained Printers, Printer fan-out, and Printer fan-in.

9. Added the concept of subordinate Printers and the "subordinate-printers-supported" (1setOf uri) Printer Description attribute to describe the configuration.

10. Added the forwarding rules: IPP Printer objects MUST NOT forward Printer operations to subordinate IPP Printer objects, except for the chained Printer configuration. IPP Printer objects MUST forward Job operations to the intended Job object.

11. Removed the "synchronize" operation attribute from all operations.

12. Renamed 'standby' to 'deactivated' Printer state reason.

13. Added 'moving-to-paused-all' Printer state reason for use with Pause-Printer-After-All-Current-Jobs

14. Added 'printer-deactivated' Printer state reason for use with Deactivate-Printer.

15. Renamed job-paused' to 'job-suspended' to go with the rename Suspend-Current-Job operation.

16. Renamed 'server-error-printer-is-in-standby-mode' status code to 'server-error-printer-is-deactivate'.

17. Grouped attributes that come in pairs.

18. Changed Shutdown-Printer so that there is no operation to come back to life, except Startup-Printer which starts a new instance (but there can only be one instance per Printer object).

18.2 Changes to the November 1, 1999 version to make the November 16, 1999 version

~~The following changes to the October 22, 1999 version to make the November 1, 1999 version as a result of the IPP WG meeting in Durham, 10/99:~~

1. Formally defined IPP Printer fan-out, IPP Printer fan-in, and output device fan-out. Added figures to show IPP Printer fan-out and IPP Printer fan-in.

2. Added "parent-printers-supported (1setOf uri) Printer Description attribute to point back up the Printer hierarchy.

3. Added the requirements for forwarding operations that affect Jobs and for not forwarding operations that affect Printers.

4. Added "original-requesting-user-name" (name(MAX)) to represent the original end user, not the parent Printer's host.

5. Changed the default for "when" for the Pause-Printer operation from 'after-current-job' to 'now', since that is the behavior in IPP/1.1 where the "when" operation attribute is not defined.

6. Allowed a non-leaf Printer to have only one subordinate Printer.

7. Changed most of the "parent" Printer terminology to "non-leaf" Printer to contrast more clearly with "leaf" Printer objects. The term "parent" is only used when talking about a subordinate's immediate parent Printer object.

8. Added "original-requesting-user-name" (name (MAX)) to the list of READ-ONLY Job Description attributes.

18.3 Changes to the October 22, 1999 version to make the November 1, 1999 version

The following changes to the October 22, 1999 version to make the November 1, 1999 version as a result of the IPP WG meeting in Durham, 10/99:

1. Removed the Reset-Printer, Non-Process-Run-Out, and Space-Current-Job operations from this Set2 spec and moved them to a new Set3 spec for use with the new Device object, renaming them appropriately, to Reset-Device, Non-Process-Run-Out-Device, and Space-Device.

2. Added the concept of parent and subordinate Printer objects to formally represent fan-out. Mentioned the Device object that is in a new [ipp-set3] spec.

3. Distributed the definition of the "when" operation attribute to the Pause-Printer (IPP/1.1), Shutdown-Printer, and Pause-Current-Job operations and listed the values that are appropriate to that operation only:
   Pause-Printer: 'now', 'after-current-copy', 'after-current-job' (default), and 'after-all'.
   Shutdown-Printer: 'now', 'after-current-job' (default), and 'after-all'
   Pause-Current-Job: 'now', 'after-current-copy' (default)

4. Deleted the "device-name" operation attribute and the "device-names-supported" (1setOf name(127)) Printer Description attribute. The latter will be part of the [ipp-set3] document.

5. Kept the "job-settable-attributes" (1setOf type2 keyword) and "printer-settable-attributes" (1setOf type2 keyword), but deleted the "interpreter-settable-attributes (1setOf type2 keyword), since the Interpreter object and its attributes are really a sub-class of the Printer object.

6. Deleted the "when-values-supported" (1setOf type2 keyword) Printer Description attribute.

7. Added the "subordinate-printers-supported" (1setOf uri) Printer Description attribute.

18.4 Changes to the September 19, 1999 version to make the October 22, 1999 version

Adding or removing ISSUES that don't change the document are not listed here. The following changes to the September 19, 1999 version to make the October 22, 1999 version as a result of the IPP WG meeting in Denver, 9/99:

1. Added the Interpreter object.
2. Added the "device-name" operation attribute to handle passing operations through the IPP Printer object to the device.
3. Added the out-of-band 'not-settable' to allow the Set-Job-Attributes and Set-Printer-Attributes response to indicate the difference between an unsupported attribute and a supported, but not settable, attribute in the Unsupported Attributes Group.
4. Removed "when-values-supported" and "job-settable-attributes" and "printer-settable-attributes" and "interpreter-settable-attributes" from the list of attributes that MUST be read-only. So an administrator could sub-set the policy on what when values are supported or which attributes can be set.

18.5 Changes to the July 19, 1999 version to make the September 19, 1999 version

The following changes to the July 19, 1999 version to make the September 19, 1999 version as a result of the IPP WG meeting in Alaska, 8/99:

1. Refer to proposal as "Set2" rather than "Administrative" operations.
2. Revise the emphasis on administrator throughout the document, although the word administrator remains wherever appropriate.
3. Convert non-process-run-out from an operations attribute to an operation.
4. Added Issue 21: For all these "access" caveats, why not just say... 'authentication and access control (see ipp-mod sections 1, 8.3 and 8.5) applies to this operation".?
5. Added Issue 22: Why? This is backward, if you ask me (HRL).
6. Per resolution of Issue 2, the "settable-attributes" Printer Description attribute, was replaced 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.

18.6 Changes to the June 30, 1999 version to make the July 19, 1999 version

The following changes to the June 30, 1999 version to make the July 19, 1999 version as a result of the IPP WG meeting in Copenhagen, 7/7/99-7/8/99, and the IPP telecon, 7/14/1999:

1. Sections 2.1 and 2.2: Clarified that the way to remove a message from the operator was for the client to supply a zero-length or all white space text string which is copied as usual to the "xxx-message-from-operator" attribute.

2. Section 2.3: Added "factory-settings" (boolean) operation attribute to the Get-Printer-Attributes operation.

3. Section 2.4: Added the "when" operation attribute to the Pause-Current-Job operation.

4. Section 2.4: Made the "when" operation attribute OPTIONAL for use in operations (Pause-Printer, Reset-Printer, Shutdown-Printer, and Pause-Current-Job operations).

5. Sections 2.5: Added table of operation attributes for the Printer operations to make it easy to compare.

6. Sections 2.6: Added table of operation attributes for the Job operations to make it easy to compare.

7. Section 3.1: Added "settable-attributes" (1setOf type2 keyword) READ-ONLY Printer Description attribute.

8. Section 3.2: Added "printer-controls-other-protocols" (boolean) Printer Description attribute

9. Section 3.3: Added the READ-ONLY "printer-message-time" (integer(MIN:MAX)) Printer Description attribute to keep time message updated in time ticks.

10. Section 4.2: Deleted the 'process-next' "job-state-reasons" value, so that repeated Promote-Job operations promote each job "to the front of the queue".

11. Sections 6.1.1.1 and 6.2.1.1: Replaced the table that listed all attributes with one that lists only the attributes that MUST be READ-ONLY.

12. Section 6.1.1.1: Indicated that attributes that are not specified as READ-ONLY in this document MAY be settable. If they control behavior, that changing their values MUST change the behavior.

13. Section 6.1.1.2 and 6.2.1.2: Deleted the "ipp-attribute-fidelity" operation attribute from the Set-Printer-Attributes and Set-Job-Attributes operations. All set operations are atomic.

14. Section 6.1.1.2: Add the concept of the Interpreter object to handle attributes whose values vary in the Set-Printer-Attributes and Get-Printer-Attributes, depending on the value of the "document-format" operation attribute.

15. Sections 6.1.1.3 and 6.2.1.2: Changed the "out-of-band" 'not-settable' value back to the existing 'not-supported' value.

16. Section 6.1.2 and 6.1.3: Added "job-type" operation attribute to Disable-Printer and Enable-Printer operations with values: 'network-jobs', 'walk-up-jobs', and 'all-jobs'.

17. Section 6.1.5: Clarified that Restart-Printer brings up the Printer disabled and paused, since that is the eventual state that Shutdown-Printer leaves the printer in.

18. Section 6.1.5: Indicated that if Restart-Printer is supported, then Shutdown-Printer MUST be supported.

19. Section 6.1.6: Deleted Space-Printer operation. Keep Space-Current-Job operation only which has a "job-id" operation attribute that a client MAY supply.

20. Section 6.1.6: Clarified that Shutdown-Printer is for a long period of time, not just to reset the device or change attribute values. Also that Shutdown performs an immediate Disable-Printer and an eventual Pause-Printer.

21. Sections 6.2.3, 6.2.4, and 6.2.7 : Added a "job-id" operation attribute to Cancel-Current-Job, Pause-Current-Job, and Space-Current-Job that a client MAY supply to check for race condition where current job changes

22. Section 6.2.4: Combined Pause-Job into Pause-Current-Job operation.

23. Sections 6.2.4 and 6.2.5: Pause-Current-Job puts job in 'processing-stopped' state, not 'pending-held' state.

24. Section 6.2.6: Simplified Promote-Job, so that it behaves as if the job were put at the front of the queue.

19 Appendix A: Full Copyright Statement