Internet Printing Protocol WG
INTERNET-DRAFT~~ — all issues have been resolved~~
<draft-ietf-ipp-ops-set2-01.txt>

Carl Kugler
H. Lewis
IBM Corporation
T. Hastings (editor)
Xerox Corporation

July 19~~6~~, 2000

Internet Printing Protocol (IPP): Job and Printer Administrative Operations

Status of this Memo

Abstract

This document is a submission to the Internet Printing Protocol Working Group of the Internet Engineering Task Force (IETF). After approval, it is intended to be on the IETF standards track. Comments should be submitted to the ipp@pwg.org mailing list.

This document specifies the following 16 additional OPTIONAL operations for use with the Internet Printing Protocol/1.0 (IPP) [RFC2565, RFC2566] and IPP/1.1 [ipp-mod, ipp-pro]~~.~~:

Printer operations:
Enable-Printer and Disable-Printer
Pause-Printer-After-Current-Job
Hold-New-Jobs and Release-Held-New-Jobs
Deactivate-Printer and Activate-Printer
Restart-Printer
Shutdown-Printer and Startup-Printer

Job operations:
Reprocess-Job
Cancel-Current-Job
Suspend-Current-Job and Resume-Job
Promote-Job
~~Redirect-Job~~Schedule-Job-After

New Job Description attributes: "original-requesting-user-name"
New Printer Description attributes: "subordinate-printers-supported"~~,~~ and "parent-printers-supported"~~, and "redirection-printers-supported"~~.
New "printer-state-reasons" values: 'hold-new-jobs' and 'deactivated'.
New "job-state-reasons" attribute values: 'job-suspended'.
New event keyword: 'forwarded-operation-failed'~~ event code~~.
New status code: 'server-error-printer-is-deactivated'.

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.

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 [IPP-MOD]
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]
Internet Printing Protocol (IPP): IPP Event Notification Specification [ipp-ntfy]

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: Model and Semantics”, describes a simplified model with abstract objects, their attributes, and their operations that are independent of encoding and transport. It introduces a Printer object and a Job object. The Job object optionally supports multiple documents per Job. It also addresses security, internationalization, and directory issues.

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.

The “Internet Printing Protocol (IPP): IPP Event Notification Specification” document defines the semantics for Subscription Creation Operations and the requirements for other Delivery Method documents to define a Delivery Method to carry an Event Notifications to a Notification Recipient.

Table of Contents

1 Introduction... 7

2 Terminology... 7
2.1 Conformance Terminology...7
2.2 Other terminology..7

3 Definition of the Printer Operations 12
3.1 The Disable and Enable Printer Operations 13
3.1.1 Disable-Printer Operation 14
3.1.2 Enable-Printer Operation 14
3.2 The Pause and Resume Printer Operations..14
3.2.1 Pause-Printer-After-Current-Job operation 15
3.3 Hold and Release New Jobs operations...17
3.3.1 Hold-New-Jobs operation 17
3.3.2 Release-Held-New-Jobs operation 18
3.4 Deactivate and Activate Printer Operations 18
3.4.1 Deactivate-Printer operation 19
3.4.2 Activate-Printer operation 19
3.5 Restart-Printer, Shutdown-Printer, and Startup-Printer operations 19
3.5.1 Restart-Printer operation 20
3.5.2 Shutdown-Printer Operation 20
3.5.3 Startup-Printer operation 21

4 Definition of the Job Operations... 21
4.1 Reprocess-Job Operation..22
4.2 Cancel-Current-Job Operation..22
4.3 Suspend and Resume Job operations...23
4.3.1 Suspend-Current-Job operation 23
4.3.2 Resume-Job operation 24
4.4 Job Scheduling Operations 25
4.4.1 Promote-Job operation 25
4.4.2 Schedule-Job-After operation 27

5 Additional status codes 28
5.1 'server-error-printer-is-deactivated' (0x050A)...28

6 Use of Operation Attributes that are Messages from the Operator.......................... 29

7 New Printer Description Attributes 30
7.1 subordinate-printers-supported (1setOf uri) 30
7.2 parent-printers-supported (1setOf uri)..30

8 Additional Values for "printer-state-reasons"... 31

8.1 'hold-new-jobs' 31
8.2 'deactivated' 31

9 Additional Values for "job-state-reasons" 31
9.1 'job-suspended' 31

10 Additional events 32

11 Use of the Printer object to represent IPP Printer Fan-Out and IPP Printer Fan-In 32
11.1 IPP Printer Fan-Out 32
11.2 IPP Printer Fan-In 32
11.3 Printer object attributes used to represent Printer Fan-Out and Printer Fan-In 33
11.4 Subordinate Printer URI 33
11.5 Printer object attributes used to represent Output Device Fan-Out 34
11.6 Figures to show all possible configurations 35
11.7 Forwarding requests 37
11.7.1 Forwarding requests that affect Printer objects 37
11.7.2 Forwarding requests that affect Jobs 37

12 Conformance Requirements 39

13 IANA Considerations 41

14 Internationalization Considerations 41

15 Security Considerations 41

16 Author's Addresses 41

17 References 42

18 Appendix A: Full Copyright Statement 49

List of Tables

Table 2 - Printer Operation Operation-Id assignments 13
Table 3 - Pause and Resume Printer and Device Operations 15
Table 4 - Job operation Operation-Id assignments 22
Table 5 - Operation attribute support for Printer Operations 29
Table 6 - Operation attribute support for Job operations 29
Table 7 - Forwarding operations that affect Printer objects 37
Table 8 - Forwarding operations that affect Jobs objects 38
Table 9 - Conformance Requirement Dependencies for Operations 39
Table 10- Conformance Requirement Dependencies for "printer-state-reasons" Values 40
Table 11- Conformance Requirement Dependencies for "job-state-reasons" Values 40

List of Figures

Figure 1 - Embedded Printer object 35
Figure 2 - Hosted Printer object 35
Figure 3 - Output Device Fan-Out 35
Figure 4 - Chained IPP Printer Objects 36
Figure 5 - IPP Printer Object Fan-Out 36
Figure 6 - IPP Printer Object Fan-In 36

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-mod, ipp-pro]) 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. In addition, this document extends the semantic model of the Printer object by allowing them to be configured into trees and/or inverted trees that represent Printer object Fan-Out and Printer object Fan-In, respectively. The special case of a tree with only a single Subordinate node represents Chained 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.

The requirements and use cases for this document are defined in [ipp-ops-admin-req]. That document also includes requirements and use cases for operations on the Device object which is the subject of a third document [ipp-device-ops]. That [ipp-device-ops] document is not needed in order to implement the operations defined in this document.

2 Terminology

This section defines terminology used throughout this document.

2.1 Conformance Terminology

Capitalized terms, such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED NOT, and OPTIONAL, have special meaning relating to conformance as specified in RFC 2119 [RFC2119] and ~~. These terms are defined in~~ [ipp-mod] section 12.1~~ on conformance terminology, most of which is taken from RFC 2119 [RFC2119]~~. These terms refer to conformance to this document or a particular operation, if this document or operation is implemented.

The following specialization of these terms apply to this document:

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

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

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.

In addition, the following capitalized terms are defined:~~.~~

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

Printer Operation - an operation whose target is an IPP Printer object and whose effect is on the Printer object.

Output Device - the physical imaging mechanism that an IPP Printer controls. Note: while this term is capitalized in this specification (but not in [ipp-mod]), there is no formal object called an Output Device defined in this document (or [ipp-mod]).

~~**Device Operation - an operation whose target is an IPP Printer object and whose defined effect is on an Output Device.**~~

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.

~~3Requirements and Use Cases~~

~~The following requirements and usage cover both the “Job and Printer Administrative Operations” (this document)and the “Device Administrative Operations” (see [ipp-device-ops]). The requirements are presented here together to show the parallelism.~~

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

~~2.Support fan-out of Printer objects.~~

~~3.Support fan-out of Output Devices.~~

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

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

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

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

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

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

~~8.Need a Device Operation to prevent an Output Device from accepting any new jobs from any job submission protocol and a companion one to restore the Output Device to accepting any jobs.~~

~~Usage: Operator is preparing to take the Output Device out of service.~~

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

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

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

~~Suggested name and operations: **Pause-Printer-After-Current-Job**, **(IPP/1.1) Resume-Printer**~~

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

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

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

~~11.Need a Printer Operation to stop the processing of IPP jobs after all of the currently accepted jobs 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 another operation after reconfiguring. He repeats the two operations to restore the Output Device to its normal media.~~

~~Suggested name and operations: **Hold-New-Jobs, Release-Held-New-Jobs**~~

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

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

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

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

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

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

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

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

~~Suggested name and operation: **Deactivate-Printer, Activate-Printer**~~

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

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

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

~~16.Need a Printer Operation to recover from the IPP Printer software that has gotten confused (run out of heap memory or gotten into a state that it doesn't seem to be able to get out of). This is a condition that shouldn't happen, but does in real life. Any volatile information is saved if possible before the software~~

~~is re-initialized. No companion operation is needed to undo this. We don't want to go back to the "confused" state :).~~

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

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

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

~~Usage: The Output Device has gotten confused or need resetting to some initial conditions.~~

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

~~18.Need a Printer Operation to put the IPP Printer object out of business with no way in the protocol to bring that instantiation back to life (but see Startup-Printer which brings up exactly one new instantiation to life with the same URL). Any volatile information is saved if possible.~~

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

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

~~19.Need a Printer Operation to bring an IPP Printer to life when there is an already running host.~~

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

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

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

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

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

~~21.Need a Device Operation to startup a powered-off device.~~

~~Usage: After a Power-Off-Device, if the device can be powered back up (possibly by an intervening host that supports the Device Operation).~~

~~Suggest name and operation: Power-On-Device~~

~~3.1List of the Printer and Device Operations~~

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

~~Table 1 – List of Printer Operations and corresponding Device Operations~~

~~Printer Operation~~	~~Corresponding Device Operation equivalent (see [ipp-device-ops])~~
~~Disable-Printer~~	~~Disable-Device~~
~~Enable-Printer~~	~~Enable-Device~~
~~Pause-Printer (IPP/1.1 - [ipp-mod] – one interpretation)~~	~~Pause-Device-Now~~
~~no~~	~~Pause-Device-After-Current-Copy~~
~~Pause-Printer-After-Current-Job~~	~~Pause-Device-After-Current-Job~~
~~Resume-Printer (IPP/1.1 - [ipp-mod])~~	~~Resume-Device~~
~~Hold-New-Jobs~~	~~no~~
~~Release-Held-New-Jobs~~	~~no~~
~~Deactivate-Printer~~	~~Deactivate-Device~~
~~Activate-Printer~~	~~Activate-Device~~
~~Purge-Jobs (IPP/1.1 – [ipp-mod])~~	~~Purge-Device~~
~~Restart-Printer~~	~~Reset-Device~~
~~Shutdown-Printer~~	~~Power-Off-Device~~
~~Startup-Printer~~	~~Power-On-Device~~

~~There are no conformance dependencies between Printer Operations and Device Operations. Either MAY be supported without supporting the corresponding operations.~~

3 Definition of the 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 defined in this document are summarized in Table 2:

Table 2 - Printer Operation Operation-Id assignments

Operation Name	Operation-Id	Brief description
Enable-Printer	0x22~~0x??~~	Allows the target Printer to accept Job Creation operations
Disable-Printer	0x23~~0x??~~	Prevents the target Printer from accepting Job Creation operations
Pause-Printer-After-Current-Job	0x24~~0x??~~	Pause the Printer after the current job has been sent to the Output Device.
Hold-New-Jobs	0x25~~0x??~~	Finishes processing all currently pending jobs. Any new jobs are placed in the 'pending-held' state.
Release-Held-New-Jobs	0x26~~0x??~~	Release all jobs to the 'pending' state that had been held by the effect of a previous Hold-New-Jobs operation and condition the Printer to no longer hold new jobs.
Deactivate-Printer	0x27~~0x??~~	Puts the Printer into a read-only deactivated state.
Activate-Printer	0x28~~0x??~~	Restores the Printer to normal activity
Restart-Printer	0x29~~0x??~~	Restarts the target Printer and re-initializes the software
Shutdown-Printer	0x2A~~0x??~~	Shuts down the target Printer so that it cannot be restarted or queried
Startup-Printer	0x2B~~0x??~~	Starts up the instance of the Printer object

All of the operations in this document are OPTIONAL for an IPP object to support. Unless the specification of an OPTIONAL operation requires support of another OPTIONAL operation, conforming implementations may support any combination of these operations. Many of the operations come in pairs and so both are REQUIRED if either one is implemented.

3.1 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 3.2) independently control the "output of new jobs" from the IPP Printer to the Output Device.

The Disable-Printer 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-device-ops]) are intended to stop the acceptance of all jobs by the associated Output Device(s).

3.1.1 Disable-Printer Operation

This OPTIONAL operation allows a client to stop the Printer object from accepting new jobs, i.e., cause the Printer to reject subsequent Job Creation operations 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.

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

The Disable-Printer Request and Disable-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 6).

3.1.2 Enable-Printer Operation

This OPTIONAL operation allows a client to start the Printer object accepting jobs, i.e., cause the Printer to accept subsequent Job Creation operations. 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.

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

The Enable-Printer Request and Enable-Printer Response have the same attribute groups and attributes as the 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 6).

3.2 The Pause and Resume Printer Operations

This section leaves the OPTIONAL IPP/1.1 Pause-Printer (see [ipp-mod] sections 3.2.7) to be ambiguous as to whether or not it stops the Printer immediately or after the current job and defines the OPTIONAL Pause-Printer-After-All-Current-Jobs operation to be after the current job. These operations affect the scheduling of IPP jobs. If either of these Pause Printer 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 3.1) 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-device-ops]) are intended to stop the acceptance of all jobs by the associated Output Device(s).

This document and [ipp-device-ops] define distinct operations in order to disambiguate the Pause-Printer operation as shown in Table 3. The Printer Operations affect only Jobs submitted using IPP, while the Device Operations affect all jobs no matter what job submission protocol was used to submit them to the Output Device.

Table 3 - Pause and Resume Printer and Device Operations

Pause and Resume Printer and Device Operations	Description
IPP/1.1 Pause Printer	Stops the IPP Printer from sending new IPP Jobs to the Output Device(s) either immediately or after the current job completes, depending on implementation, as defined in [ipp-mod].
Pause-Printer-After-Current-Job	Stops the IPP Printer from sending new IPP Jobs to the Output Device(s) after the current jobs finish
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.~~

3.2.1 Pause-Printer-After-Current-Job 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. 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 3.1.1).

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”).

The IPP Printer MUST accept the request in any state and transition the Printer to the indicated new “printer-state” and MUST add the indicated value to “printer-state-reasons” attribute before returning as follows:

Current “printer-state”	New “printer-state”	“printer-state-reasons”	IPP Printer’s response status code and action: REQUIRED/OPTIONAL state transition for a Printer to support
‘idle’	‘stopped’	‘paused’	REQUIRED: ‘successful-ok’
‘processing’	‘processing’	‘moving-to-paused’	OPTIONAL: ‘successful-ok’; Later, when the IPP Printer has finished sending IPP jobs to an Output Device, the “printer-state” becomes ‘stopped’, and the ‘paused’ value replaces the ‘moving-to-paused’ value in the “printer-state-reasons” attribute
‘processing’	‘stopped’	‘paused’	REQUIRED: ‘successful-ok’; the IPP Printer wasn’t in the middle of sending an IPP job to an Output Device
‘stopped’	‘stopped’	‘paused’	REQUIRED: ‘successful-ok’

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-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 6).

3.3 Hold and Release New Jobs operations

This section defines operations to condition the Printer to hold any new jobs and to release them.

3.3.1 Hold-New-Jobs operation

This OPTIONAL operation allows a client to condition the Printer to complete the current ‘pending’ and ‘processing’ IPP Jobs but not start processing any subsequently created 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 with the ‘job-held-on-create’ value being added to the job’s “job-state-reasons” attribute. Thus all newly accepted jobs will be automatically held by the Printer.

When the Printer completes all of the ‘pending’ and ‘processing’ jobs, it enters the ‘idle’ state as usual. An operator that is monitoring Printer state changes will know when the Printer has completed all current jobs because the Printer enters the ‘idle’ state.

This operation MUST NOT affect the acceptance of Job Creation requests (see Disable-Printer section 3.1.1), except to put the Jobs into the ‘pending-held’ state, instead of the ‘pending’ or ‘processing’ state.

The IPP Printer MUST accept the request in any state, MUST NOT transition the Printer to any other “printer-state”, and MUST add the ‘hold-new-jobs’ value to the Printer’s “printer-state-reasons” attribute (whether the value was present 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 Hold-New-Jobs Request and Hold-New-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 6).

3.3.2 Release-Held-New-Jobs operation

This OPTIONAL operation allows a client to undo the effect of a previous Hold-New-Jobs operation. In particular, the Printer releases all of the jobs that it had held as a consequence of a Hold-New-Jobs operations, i.e., while the ‘hold-new-jobs’ value was present in the Printer’s “printer-state-reasons” attribute. In addition, the Printer MUST accept this request in any state, MUST NOT transition the Printer to any other “printer-state”, and MUST remove the ‘hold-new-jobs’ value from its “printer-state-reasons” attribute (whether the value was present or not) so that the Printer no longer holds newly created jobs.

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 Release-Held-New-Jobs Request and Release-Held-New-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 6).

3.4 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-device-ops]) are intended to stop the associated Output Device(s) from performing work and accepting operations, except query operations.

3.4.1 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 3.1.1) 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 ‘deactivated’ value in its “printer-state-reasons” attribute. Note: neither the Disable-Printer nor the Pause-Printer-After-Current-Job set the ‘deactivated’ value.

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 6).

3.4.2 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 ‘deactivated’ value from its “printer-state-reasons” attribute (whether present 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 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 6).

3.5 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-device-ops]) are intended to initialize or power off the associated Output Device(s).

3.5.1 Restart-Printer 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'.

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 Restart-Printer Request and Restart-Printer Response have the same attribute groups and attributes as the 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 6).

3.5.2 Shutdown-Printer 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 without losing any jobs. There is no way to bring the instance of the Printer object back to being used, except for the Startup-Printer (see section 3.5.3) 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 3.5.1), Startup-Printer (section), and Reset-Device [ipp-device-ops] for the way to initialize the software or reset the Output Device(s). See the Disable-Printer operation (section 3.1) for a way for the client to stop the Printer from accepting Job Creation requests without stopping processing or shutting down.

The Printer MUST add 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 3.4.1) which performs a Disable-Printer and Pause-Printer-After-Current-Job operation).

Note: In order to shutdown the Printer after all the currently submitted jobs have completed, the operator issues a Disable-Printer operation (see section 3.1.1) and then waits until all the jobs have completed and the Printer goes into the 'idle' state before issuing the Shutdown-Printer operation.

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 Pause-Printer-After-Current-Job operation until the activity is completed and the Printer object disappears.

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 6).

3.5.3 Startup-Printer operation

This OPTIONAL operation allows a client to startup an instance of a Printer object, provided that there isn't one already instantiated. The purpose of Startup-Printer is to allow a hosted implementation of the IPP Printer object (i.e., a Server that implements an IPP Printer on behalf of a networked or local Output Device) to be started after the host is available (by means outside this document). See Restart-Printer (section 3.5.1) and Reset-Device [ipp-device-ops] 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 'false'. Then the operator can reconfigure the Printer before performing an Enable-Printer operation. However, when a Printer is first powered up, it is RECOMMENDED that its "printer-is-accepting-jobs" attribute be set to 'true' in order to achieve easy "out of the box" operation.

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 6).

4 Definition of the 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 Job Operations defined in this document are summarized in Table 4:

Table 4 - Job operation Operation-Id assignments

Operation Name	Operation-Id	Brief description
Reprocess-Job	0x2C~~0x??~~	Creates a copy of a completed target job with a new Job ID and processes it
Cancel-Current-Job	0x2D~~0x??~~	Cancels the current job on the target Printer or the specified job if it is the current job
Suspend-Current-Job	0x2E~~0x??~~	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	0x2F~~0x??~~	Resume the suspended target job
Promote-Job	0x30~~0x??~~	Promote the pending target job to be next after the current job(s) complete
~~Redirect-Job~~	~~0x??~~	~~Redirect the target job to the specified Printer on the same server.~~
Schedule-Job-After	0x31~~0x??~~	Schedule the target job immediately after the specified job, all other scheduling factors being equal.

4.1 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.

4.2 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.

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).

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 6), 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.

4.3 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.

4.3.1 Suspend-Current-Job operation

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-suspended’ value (see section 9.1) 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 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 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 suspend the new job, the wrong job is 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 suspends the specified job and processed other jobs.

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

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).

The Suspend-Current-Job Request and Suspend-Current-Job Response have the same attribute groups and attributes as the Pause-Printer operation (see [ipp-mod] section 3.2.8), including the new "job-message-from-operator" operation attribute (see section 6), 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.

4.3.2 Resume-Job operation

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

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

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).

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 6).

4.4 Job Scheduling Operations

This section defines jobs that allow an operator to control the scheduling of jobs.

4.4.1 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 MUST reject~~s~~ the request and return~~s~~ the ‘client-error-not-possible’ status code.

If the Printer implements the “job-priority” Job Template attribute (see [ipp-mod] section 4.2.1), the Printer sets the job’s “job-priority” to the highest value supported (so that the job will print before any of the other pending jobs). 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, suspended (see section 4.3.1), 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~~ processed before the previous job that was to be processed next.

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).

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 6).

~~12.5Redirect-Job operation~~

~~This OPTIONAL operation allows a client to redirect a not-completed job to another Printer on the same server. Redirect-Job is defined to be a Job Creation operation, along with the Print-Job, Print-URI, and Create-Job operations. Thus all semantics that apply to Job Creation operations also apply to this operation. For example, the new Printer validates the job using all of its "xxx-supported" attributes and either accepts or rejects the job. If the job is rejected, it remains in its original state before the Redirect-Job operation was attempted. As an other example, the Job inherits the defaults for the new Printer (since the defaults aren't copied onto the Job object when it is created, but are applied when the job is processed - see [ipp-mod]). Finally, this operation generates a 'job-created' event as does any Job Creation Operation.~~

~~In order to preserver the "ipp-attribute-fidelity" semantics that the original client supplied when the job was first created, each Job Creation Operation copies the "ipp-attributes-fidelity" (boolean) operation attribute o the job as a Job Description attribute, if the Redirect-Job operation is supported. Then the "ipp-attribute-fidelity" attribute is re-used by the new Printer during its job validation, unless the client performing the Redirect-Job operation supplies the "ipp-attribute-fidelity" operation attribute.~~

~~This operation is limited to redirecting a job to another Printer on the same server. Thus the same copy of the job MAY be used, depending on implementation. Also, depending on implementation, the new Printer MAY generate a new job-id and job-uri, or use the same one. In either case the response contains the "job-id" and "job-uri" for the redirected job as for any Job Creation operation. If the new Printer does assign a new "job-id" and "job-uri", then it MUST automatically update an Per-Job Subscription objects that are associated with the job.~~

~~The Printer MUST accept this operation whenever the job is in the 'pending' or 'pending-held' states. The Printer MUST reject this operation whenever the job is in the 'completed', 'aborted', or 'canceled' states and return the 'client-error-not-possible' status code. Whether the Printer accepts this operation when the job is in the 'processing' or 'processing-stopped' states depends on implementation.~~

~~*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).~~

~~The Redirect-Job Request have the same attribute groups and attributes as the Create-Job operation (see [ipp-mod] section 3.2.4), plus the new "job-message-from-operator" operation attribute (see section 5). In addition, the following operation attributes are defined:~~

~~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. The client MUST supply this attribute and the Printer MUST support it.~~

~~new-printer-uri (uri):~~

~~The URI of another Printer on the same server. The client MUST supply this attribute and the Printer MUST support it.~~

~~ipp-attribute-fidelity (boolean):~~

~~The client MAY supply this attribute, but the Printer MUST support it. It indicates whether or not the Job Template attributes on the Job object MUST be supported by the new Printer. If the client omits this attribute, the new Printer uses the value copied to the job as a Job Description attribute when the job was originally created. The Job Description attribute is not affected by the value supplied in this request, so that the original user's intent is preserved across multiple Redirect-Job operations.~~

~~The Redirect-Job Response has the same attribute groups, attributes, and status codes as the Create-Job operation (see [ipp-mod] section 3.2.4). The following status codes have particular meaning for this operation:~~

~~'client-error-not-possible' - the job was in the 'completed', 'aborted', or 'canceled' states or the implementation does not support the Redirect-Job operation on a job when it is in the 'processing' or 'processing-stopped' states.~~

~~'client-error-not-found' - the target job was not found.~~

~~'client-error-attributes-or-values-not-supported' - the specified Printer is not supported for redirection, i.e., the URI was not amongst the Printer's "redirection-printers-supported" (1setOf uri).~~

4.4.2 Schedule-Job-After operation

This OPTIONAL operation allows a client to request the Printer to schedule the target job so that it will be processed immediately after the specified predecessor job, all other scheduling factors being equal. 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 predecessor job completes. The predecessor job can be in the ‘pending’, ‘processing’, or ‘processing-stopped’ states. If the target job is not in the ‘pending’ state or the predecessor job is not in the ‘pending’, ‘processing’, or ‘processing-stopped’ states, the Printer MUST reject the request and returns the ‘client-error-not-possible’ status code, since the job cannot have its position changed.

If the Printer implements the “job-priority” Job Template attribute (see [ipp-mod] section 4.2.1), the Printer sets the job’s “job-priority” to that of the predecessor job (so that the job will print after the predecessor job). The Printer returns the target job immediately after the predecessor in a Get-Jobs response (see [ipp-mod] section 3.2.6) for the ‘not-completed’ jobs.

When the predecessor job completes processing or is canceled or aborted while processing, the target of this operation is processed next.

If the client does not supply a predecessor job, this operation has the same semantics as Promote-Job (see section 4.4).

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 Schedule-Job-After operation puts the specified job immediately after the specified job in the queue. A subsequent Schedule-Job-After operation specifying the same job will cause its target job to be placed after that job, even though it is between the first target job and the specified job. For example, suppose the job queue consisted of jobs: A, B, C, D, and E, in that order. A Schedule-Job-After with job E as the target and B as the specified job would result in the following queue: A, B, E, C, D. A subsequent Schedule-Job-After with Job D as the target and B as the specified job would result in the following queue: A, B, D, E, C. In other words, the link between the two jobs in a Schedule-Job-After operation is ~~ephemeral~~not retained, i.e., there is no attribute on either job that points to the other job as a result of this operation~~, rather than setting an attribute of either of the jobs~~.

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

The Schedule-Job-After Request have the same attribute groups and attributes as the Cancel-Job operation (see [ipp-mod] section 3.3.3), plus the new “job-message-from-operator” operation attribute (see section 6). In addition, the following operation attributes are defined:

“predecessor-job-id”:

The client OPTIONALLY supplies this attribute. The Printer MUST support it, if it supports this operation. This attribute specifies the job after which the target job is to be ~~scheduled~~processed. If the client omits this attribute, the Printer MUST ~~schedule~~ process the target job next, i.e., after the current job, if any.

The Schedule-Job-After Response has the same attribute groups, attributes, and status codes as the Cancel-Job operation (see [ipp-mod] section 3.3.3). The following status codes have particular meaning for this operation:

‘client-error-not-possible’ - the target job was not in the ‘pending’ state or the predecessor job was no in the ‘pending’, ‘processing’, or ‘processing-stopped’ states.

‘client-error-not-found’ - either the target job or the predecessor job was not found.

5 Additional status codes

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

5.1 ‘server-error-printer-is-deactivated’ (0x050A)

The Printer has been deactivated using the Deactivate-Printer operation and is only accepting the Activate-Printer (see section 3.5.1), Get-Job-Attributes, Get-Jobs, Get-Printer-Attributes, and any other Get-Xxxx operations. An operator can perform the Activate-Printer operation to allow the Printer to accept other operations.

6 Use of Operation Attributes that are Messages from the Operator

This section summarizes the usage of the “printer-message-from-operator” and “job-message-from-operator” operation attributes that set the corresponding Printer and Job Description attributes (see [ipp-set-ops] for the definition of these operation attributes). These operation attributes are defined for most of the Device and Job operations that operators are likely to perform, respectively, so that operators can indicate the reasons for their actions.

Table 5 shows the operation attributes that are defined for use with the Printer Operations.

Legend:

REQ - REQUIRED for a Printer to support
OPT - OPTIONAL for a Printer to support; the Printer ignores the attribute if not supported
<blank> - not defined for use with the operation; the Printer ignores the attribute

Table 5 - Operation attribute support for Printer Operations

Operation Attribute	Pause-Printer, Pause-Printer-After-Current-Job, Resume-Printer	Hold-New-Jobs, Release-Held-New-Jobs	Purge-Jobs	Get-Printer-Attributes, Set-Printer-Attributes	Enable-Print, Disable-Printer	Restart-Printer	Shut down-Printer, Startup-Printer
attributes-charset	REQ	REQ	REQ	REQ	REQ	REQ	REQ
attributes-natural-language	REQ	REQ	REQ	REQ	REQ	REQ	REQ
printer-uri	REQ	REQ	REQ	REQ	REQ	REQ	REQ
requesting-user-name	REQ	REQ	REQ	REQ	REQ	REQ	REQ
printer-message-from-operator	OPT	OPT	OPT		OPT	OPT	OPT

Table 6 shows the operation attributes that are defined for use with the Job operations.

Legend:

REQ - REQUIRED for a Printer to support
O - OPTIONAL for a Printer to support; the Printer ignores the attribute if supplied, but not supported
<blank> - not defined for use with the operation; the Printer ignores the attribute

Table 6 - Operation attribute support for Job operations

Operation Attribute	Cancel-Job	Cancel-Current-Job	Hold-Job, Release-Job	Suspend-Current-Job	Resume-Job	Get-Job-Attributes, Set-Job-Attributes	Restart-Job	Reprocess-Job	Promote-Job	Schedule-Job-After
attributes-charset	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ
attributes-natural-language	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ
printer-uri	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ
job-uri	REQ		REQ		REQ	REQ	REQ	REQ	REQ	REQ
job-id	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ

Operation Attribute	Cancel-Job	Cancel-Current-Job	Hold-Job, Release-Job	Suspend-Current-Job	Resume-Job	Get-Job-Attributes, Set-Job-Attributes	Restart-Job	Reprocess-Job	Promote-Job	Schedule-Job-After
requesting-user-name	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ	REQ
job-message-from-operator	OPT	OPT	OPT	OPT	OPT		OPT	OPT	OPT	OPT
message [to-operator]	OPT		OPT	OPT	OPT		OPT	OPT	OPT	OPT
job-hold-until			OPT*					OPT**		

* 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.

7 New Printer Description Attributes

The following new Printer Description attributes are needed to support the new operations defined in this document and the concepts of Printer Fan-Out (see section 11).

7.1 subordinate-printers-supported (1setOf uri)

This Printer attribute is REQUIRED if an implementation supports Subordinate Printers (see section 11) and contains the URIs of the immediate Subordinate Printer object(s) associated with this Printer object. 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.

The precise format of the Subordinate Printer URIs is implementation dependent (see section 11.4).

If the Printer object does not have an associated Output Device, the Printer MAY automatically copy 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.

7.2 parent-printers-supported (1setOf uri)

This Printer attribute is REQUIRED if an implementation supports Subordinate Printers (see section 11) and 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.

~~8.3redirection-printers-supported (1setOf uri)~~

~~This Printer attribute is REQUIRED if an implementation supports the Redirect-Job operation (see section 12.5). It specifies the URIs that the Printer supports for redirection jobs to other Printers (on the same server).~~

8 Additional Values for “printer-state-reasons”

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

8.1 ‘hold-new-jobs’

‘hold-new-jobs’: The operator has issued the Hold-New-Jobs operation (see section 3.3.1) 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 Hold-New-Jobs operation is supported and the implementation takes significant time to pause a device in certain circumstances.

8.2 ‘deactivated’

‘deactivated’: A client has issued a Deactivate-Printer operation for the Printer object (see section 3.4.1) 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 3.1.1) and return the ‘server-error-service-unavailable’ status code.

9 Additional Values for “job-state-reasons”

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

9.1 ‘job-suspended’

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

10 Additional events

The following Printer events are defined for use with [ipp-ntfy]:

‘forwarded-operation-failed’ - an operation that a Printer forwarded to a Subordinate Printer (see section 11.7) failed.

11 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.

11.1 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 document 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 supports 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.

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 [ipp-mod] 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.

11.2 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 11.1), when a Printer object is a Non-Leaf Printer, it MUST NOT have an associated Output Device. As with Fan-Out, a Leaf Printer object has one or more associated Output Devices. 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.

11.3 Printer object attributes used to represent Printer Fan-Out and Printer Fan-In

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).

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”.

11.4 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 subscribe to event notification when the Subordinate Printer object supports IPP notification).

11.5 Printer object attributes used to represent Output Device Fan-Out

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 Leaf Printer’s “output-devices-supported” attribute SHOULD have only one value.

Non-Leaf Printer MUST NOT have associated Output Devices. However, a Non-Leaf Printer SHOULD support an “output-devices-supported” (1setOf name(127)) Printer Description attribute that contains all the values of its immediate Subordinate Printers. Since such Subordinate Printers MAY be Leaf or Non-Leaf, the same rules apply to them, etc. Thus any Non-Leaf Printer SHOULD have an “output-devices-supported” (1setOf name(127)) attribute that contains all the values of the Output Devices associated with Leaf Printers of its complete sub-tree.

When adding, removing, or changing a configuration of Printers and Output Devices, there can be moments in time when the tree structure is not consistent. In other words, times when a Non-Leaf Printer’s “subordinate-printers-supported” does not agree with the Subordinate Printer’s “parent-printers-supported”. Therefore, the operator SHOULD first Deactivate all Printers that are being configured in this way, update all pointer attributes, and then reactivate. A useful client tool would validate a tree structure before Activating the Printers involved.

11.6 Figures to show all possible configurations

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

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

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

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

```
                                               Output Device
                                             +---------------+
                                             |  ###########  |
 O   +--------+                              |  # (Leaf)  #  |
/|\  | client |------------IPP----------------># Printer #  |
/ \  +--------+                              |  # Object  #  |
                                             |  ###########  |
                                             +---------------+
```

Figure 1 - Embedded Printer object

```
                          ###########          Output Device
 O   +--------+           # (Leaf)  #        +---------------+
/|\  | client |---IPP----># Printer #---any->|               |
/ \  +--------+           # object  #        |               |
                          ###########        +---------------+
```

Figure 2 - Hosted Printer object

```
                                             +---------------+
                                             |               |
                                          +->| Output Device |
                          ########### any/   |               |
 O   +--------+           # (Leaf)  #   /    +---------------+
/|\  | client |---IPP----># Printer #--*
/ \  +--------+           # Object  #   \    +---------------+
                          ########### any\   |               |
                                          +->| Output Device |
                                             |               |
                                             +---------------+
```

Figure 3 - Output Device Fan-Out

```
                          ###########           ###########
 O   +--------+           # Non-Leaf#           # subord. #
/|\  | client |---IPP---->#  Printer #---IPP---->#  Printer #
/ \  +--------+           # object  #           # object  #
                          ###########           ###########
```

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

Figure 4 - Chained IPP Printer Objects

```
                +------IPP--------------------->###########
               /                           +--->#  subord. #
              /                           /     # Printer #
             /            ###########   any     # object  #
 O   +--------+           # Non-Leaf#   /       ###########
/|\  | client |---IPP---->#  Printer #--*
/ \  +--------+           # object  #   \
             \            ###########   any     ###########
              \                           \     # subord. #
               \                           +--->#  Printer #
                +------IPP--------------------->#  object  #
                                                ###########
```

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

Figure 5 - IPP Printer Object Fan-Out

```
                          (Non-Leaf)
                          ###########
                          # Non-Leaf#
                     +--->#  Printer #-+
                    /     # object  #  \
                  IPP     ###########   \       ###########
 O   +--------+   /                      +-IPP-># subord. #
/|\  | client |--+-----------IPP--------------->#  Printer #
/ \  +--------+   \                      +-IPP-># object  #
                  IPP     ###########   /       ###########
                    \     # Non-Leaf#  /
                     +--->#  Printer #-+
                          # object  #
                          ###########
                          (Non-Leaf)
```

The Subordinate Printer can be a Non-Leaf Printer as in Figure 4, Figure 5, or Figure 6, or can be a Leaf Printer as in Figure 1, Figure 2, or Figure 3.

Figure 6 - IPP Printer Object Fan-In

11.7 Forwarding requests

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

11.7.1 Forwarding requests that affect Printer objects

In Printer Fan-Out, Printer Fan-In, and Chained Printers, the Non-Leaf IPP Printer object MUST NOT forward the 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 7.1).

Table 7 lists the operations that affect Printer objects and the forwarding behavior that a Non-Leaf Printer MUST exhibit to its immediate Subordinate Printers. Operations that affect jobs have a different forwarding rule (see section 11.7.2 and Table 8):

Table 7 - Forwarding operations that affect Printer objects

Printer Operation	Non-Leaf Printer action
Printer Operations:	
Enable-Printer	MUST NOT forward to any of its Subordinate Printers
Disable-Printer	MUST NOT forward to any of its Subordinate Printers
Hold-New-Jobs	MUST NOT forward to any of its Subordinate Printers
Release-Held-New-Jobs	MUST NOT forward to any of its Subordinate Printers
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:	See [ipp-mod]
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
Set operations:	See [ipp-set-ops]
Set-Printer-Attributes	MUST NOT forward to any of its Subordinate Printers

11.7.2 Forwarding requests that affect Jobs

Unlike Printer Operations that only affect Printer objects (see section 11.7.1), 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.

Table 8 lists the operations that directly affect jobs and the forwarding behavior that a Non-Leaf Printer MUST exhibit to its Subordinate Printers:

Table 8 - Forwarding operations that affect Jobs objects

Job operation	Non-Leaf Printer action
Job operations:	
Reprocess-Job	MUST forward to the appropriate Job in one of its Subordinate Printers
Cancel-Current-Job	MUST NOT forward
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

Job operation	Non-Leaf Printer action
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
IPP Set operations:	See [ipp-set-ops]
Set-Job-Attributes	MUST forward to the appropriate Job in one of its Subordinate Printers

When a Printer receives a request that REQUIRES forwarding, it does so on a “best efforts basis”, and returns a response to its client without waiting for responses from any of its Subordinate Printers. Such forwarded requests could fail. In order for a client to become aware of such a condition, a new ‘forwarded-operation-failed’ event is defined, which a client can subscribe to (see section 10 and [ipp-ntfy]).

The following Job Description attributes are defined to help represent Job relationships for Fan-Out and forwarding of jobs:

1. “output-device-assigned” (name(127)) - from [ipp-mod]: This attribute identifies the Output Device to 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. The IPP/1.1 “requesting-user-name” operation attribute (see [ipp-mod]) is updated by each client to be itself on each hop, i.e., the “requesting-user-name” is the client forwarding the request, not the original client. The “job-originating-user-name” Job Description attribute remains as the authenticated original user, not the parent Printer’s authenticated host, and is forwarded by each client without changing the value.

12 Conformance Requirements

The Job and Printer Administrative operations defined in this document are OPTIONAL operations. However, some operations MUST be implemented if others are implemented as shown in Table 9.

Table 9 - Conformance Requirement Dependencies for Operations

Operations REQUIRED	If any of these operations are supported:

Operations REQUIRED	If any of these operations are supported:
Enable-Printer	Disable-Printer
Disable-Printer	Enable-Printer
Pause-Printer	Resume-Printer
Resume-Printer	Pause-Printer, Pause-Printer-After-Current-Job
Hold-New-Jobs	Release-Held-New-Jobs
Release-Held-New-Jobs	Hold-New-Jobs
Activate-Printer, Disable-Printer, Pause-Printer-After-Current-Job	Deactivate-Printer
Deactivate-Printer, Enable-Printer, Resume-Printer	Activate-Printer
Restart-Printer	none
Shutdown-Printer	none
Startup-Printer	none
Reprocess-Job	none
Cancel-Current-Job	none
Resume-Job	Suspend-Current-Job
Suspend-Current-Job	Resume-Job
Promote-Job	none
Schedule-Job-After	Promote-Job

Table 10 and Table 11 list the “printer-state-reasons” and “job-state-reasons” values that are REQUIRED if the indicated operations are supported.

Table 10- Conformance Requirement Dependencies for “printer-state-reasons” Values

“printer-state-reasons” values:	Conformance Requirement	If any of the following Printer Operations are supported:
‘paused’	REQUIRED	Pause-Printer, Pause-Printer-After-Current-Job, or Deactivate-Printer
‘hold-new-jobs’	REQUIRED	Hold-New-Jobs
‘moving-to-paused’	OPTIONAL	Pause-Printer, Pause-Printer-After-Current-Job, Deactivate-Printer
‘deactivated’	REQUIRED	Deactivate-Printer

Table 11- Conformance Requirement Dependencies for “job-state-reasons” Values

“job-state-reasons” values:	Conformance Requirement	If any of the following Job operations are supported:
‘job-suspended’	REQUIRED	Suspend-Current-Job
‘printer-stopped’	REQUIRED	always REQUIRED

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/ipp-admin-ops.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 CO

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 CO

Phone: (303) 924-5337

FAX:
e-mail: harryl@us.ibm.com

17 References

~~[ipp-device-ops]~~
~~Kugler, C., Hastings, T., Lewis, H., “Internet Printing Protocol (IPP): Device Administrative Operations”, <draft-ietf-ipp-ops-set3-00.txt>, December 8, 1999.~~

[ipp-iig]
Hastings, T., Manros, C., “Internet Printing Protocol/1.1: draft-ietf-ipp-implementers-guide-v11-01.txt, work in progress, May 9, 2000.

[ipp-mod]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, “Internet Printing Protocol/1.0: Model and Semantics”, <draft-ietf-ipp-model-v11-07.txt>, May 22, 2000.

[ipp-ntfy]
Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-04.txt>, July 13, 2000.

[ipp-ops-admin-req]
Hastings, T., "Internet Printing Protocol (IPP): Requirements for Job, Printer, and Device Administrative Operations", <draft-ietf-ipp-ops-admin-req-00.txt>, work in progress, July 19, 2000.

[ipp-pro]
Herriot, R., Butler, S., Moore, P., Tuner, R., “Internet Printing Protocol/1.1: Encoding and Transport”, draft-ietf-ipp-protocol-v11-06.txt, May 30, 2000.

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

Change History of this document is available at:
ftp://ftp.pwg.org/pub/pwg/ipp/new_OPS/ipp-ops-set2-change-history.txt

~~18Change History~~

~~19.1This 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.~~

~~18.2-“--”--’“-”“--”----Fan-InChanges to the February 3, 2000 version to make the July 6, 2000 version~~

~~The following changes to the February 3, 2000 version to make the July 6, 2000 version as a result of the February 2000 IPP WG meeting and subsequent email and telecons:~~

~~1.Renamed the Pause-Printer-After-All-Current-Jobs operation to Hold-New-Jobs and added a complementary Release-Held-New-Jobs operation. This sets the ‘hold-new-jobs’ value (instead of the ‘moving-to-paused-all’ which is gone) in the Printer’s “printer-state-reasons” so that new jobs are held. The Printer eventually goes idle when all the current jobs have been processed.~~

~~2.Added the Redirect-Job operation to redirect a job from one Printer to another on the same server. It had been previously called Move-Job, but no movement is required.~~

~~3.Added the Schedule-Job-After operation to schedule a job immediately a specified job.~~

~~4.Added Printer Description attribute: “redirection-printers-supported” for validating the Printers that the Redirect-Job operation supports.~~

~~5.Added the ‘forwarded-operation-failed’ event code.~~

~~6.Left IPP/1.1 Pause Printer ambiguous as to whether it pauses immediately or after the current job. So the Pause-Printer-After-Current-Job is the unambiguously after the current job.~~

~~7.Capitalized the terms throughout the document.~~

~~8.Clarified that either the Printer or the Device operations or both can be supported independently of each other.~~

~~9.Clarified that it is the client’s responsibility to keep the Printer’s subordinate and parent pointers correct, not the Printer’s.~~

~~10.Clarified that forwarding operations is done on a best efforts basis and not before returning a response. The ‘forwarded-operation-failed’ event helps indicate such problems.~~

~~11.Changed Startup-Printer so that “printer-is-accepting-jobs” is set to ‘false’. But SHOULD be true when the Printer is powered up, so that it works out of the box.~~

~~18.3Changes to the December 8, 1999 version to make the February 3, 2000 version~~

~~The following changes to the December 8, 1999 version to make the February 3, 2000 version as a result of the December 1999 IPP WG meeting:~~

~~1.The Set-Printer-Attributes and Set-Job-Attributes operations were moved to a new "Job and Printer Set operations" spec [ipp-set-ops], along with the "printer-message-from-operator" & "job-message-from-operator" operation attributes, the "printer-settable-attributes", "job-settable-attributes", "printer-message-time" (integer), and "printer-message-date-time" (dateTime) Printer Description attributes, the 'client-error-attributes-not-settable' status code, and the 'not-settable' out-of-band value.~~

~~2.Deleted the "printer-message-operation: (type2 keyword) altogether.~~

~~3.Add a requirement to startup a powered-off device, say, Power-On-Device.~~

~~4.Deleted the Interpreter object. Functionality moved to the [ipp-set-ops] spec through the addition of a "document-format-varying-attributes" (1setOf type2 keyword) Printer Description attribute instead.~~

~~5.Clarified that, while a Non-Leaf Printer MUST NOT have associated devices, it SHOULD have an "output-devices-supported" (1setOf name(127)) Printer Description attribute which is a roll up of its subordinate "output-devices-supported" attributes.~~

~~6.Changed Suspend-Current-Job operation so that the Printer MUST NOT forward it to subordinate Printers.~~

~~7.Clarified that as jobs are forwarded, the IPP/1.1 "requesting-user-name" operation attribute is the immediate submitting client while the "job-originating-user-name" Job Description attribute is the authenticated original user.~~

~~8.Left IPP/1.1 Pause-Printer operation unchanged with multiple interpretations. The Pause-Printer-After-Current-Job, Pause-Device-Now, Pause-Device-After-Current-Copy, and Pause-Device-After-Current-Job all provide unambiguous interpretations.~~

~~9.Clarified that the 'paused' values is REQUIRED if the Pause-Printer or Pause-Printer-After-Current-Job operations are supported, but that 'moving-to-paused' depends on implementation.~~

~~10.Clarified that the 'paused' and 'moving-to-paused-all' values is REQUIRED if the Pause-Printer-After-All-Jobs operation is supported.~~

~~11.Clarified that the Shutdown-Printer operation MUST NOT lose any jobs.~~

~~12.Added a Conformance section which as a "Conformance Requirement Dependencies For Operations" table and a "Conformance Requirement Dependencies for State Reasons Values" table.~~

~~18.4Changes 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-deactivated'.~~

~~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).~~

~~18.5Changes to the November 1, 1999 version to make the November 16, 1999 version~~

~~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.~~

~~18.6Changes 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-device-ops] 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-device-ops] 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.~~

~~18.7Changes 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.~~

~~18.8Changes 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.~~

~~18.9Changes 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.~~

18 Appendix A: Full Copyright Statement