

INTERNET-DRAFT
<draft-ietf-ipp-ops-set2-01.txt>

Carl Kugler
IBM Corporation
T. Hastings
Xerox Corporation
H. Lewis
IBM Corporation
July 6, 2000

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

Copyright (C) The Internet Society (2000). All Rights Reserved.

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of \[rfc2026\]](#). Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed as
<http://www.ietf.org/shadow.html>.

Abstract

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:	Job operations:
Enable-Printer and Disable-Printer	Reprocess-Job
Pause-Printer-After-Current-Job	Cancel-Current-Job
Hold-New-Jobs and Release-Held-New-Jobs	Suspend-Current-Job and Resume-Job
Deactivate-Printer and Activate-Printer	Promote-Job
Restart-Printer	Redirect-Job
Shutdown-Printer and Startup-Printer	Schedule-Job-After

New Printer Description attributes: "subordinate-printers-supported", "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 'forwarded-operation-failed' event code.
New status code: 'server-error-printer-is-deactivated'.

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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](#)]

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

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

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

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

requests is given, including error checking. Motivation for some of the specification decisions is also included.

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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

Table of Contents

1	Introduction.....	6
2	Terminology.....	6
2.1	Conformance Terminology.....	6
2.2	Other terminology.....	6
3	Requirements and Use Cases.....	7
3.1	List of the Printer and Device Operations.....	11
4	Use of the Printer object to represent IPP Printer fan-out and IPP Printer fan-in.....	11
4.1	IPP Printer Fan-Out.....	12
4.2	IPP Printer Fan-In.....	12
4.3	Printer object attributes used to represent Printer fan-out and Printer fan-in.....	13
4.4	Subordinate Printer URI.....	13
4.5	Printer object attributes used to represent Output Device Fan-Out	14
4.6	Figures to show all possible configurations.....	15
4.7	Forwarding requests.....	17
4.7.1	Forwarding requests that affect Printer objects.....	17
4.7.2	Forwarding requests that affect Jobs.....	18
5	New Operation attributes.....	20
6	New Printer Description Attributes.....	21
6.1	subordinate-printers-supported (1setOf uri).....	22
6.2	parent-printers-supported (1setOf uri).....	22
6.3	redirection-printers-supported (1setOf uri).....	22
7	Additional Values for "printer-state-reasons".....	22
7.1	'hold-new-jobs'.....	23
7.2	'deactivated'.....	23
8	Additional Values for "job-state-reasons".....	23
8.1	'job-suspended'.....	23
9	Additional events.....	23
10	Additional status codes.....	23
10.1	'server-error-printer-is-deactivated' (0x????).....	24
11	Definition of the Printer Operations.....	24

11.1The Disable and Enable Printer Operations.....	26
11.1.1 Disable-Printer Operation.....	26
11.1.2 Enable-Printer Operation.....	27
11.2The Pause and Resume Printer Operations.....	27
11.2.1 Pause-Printer-After-Current-Job operation.....	28
11.3Hold and Release New Jobs operations.....	30
11.3.1 Hold-New-Jobs operation.....	30
11.3.2 Release-Held-New-Jobs operation.....	30

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

11.4	Deactivate and Activate Printer Operations.....	31
11.4.1	Deactivate-Printer operation.....	31
11.4.2	Activate-Printer operation.....	32
11.5	Restart-Printer, Shutdown-Printer, and Startup-Printer operations	32
11.5.1	Restart-Printer operation.....	33
11.5.2	Shutdown-Printer Operation.....	34
11.5.3	Startup-Printer operation.....	34
12	Definition of the Job Operations.....	36
12.1	Reprocess-Job Operation.....	37
12.2	Cancel-Current-Job Operation.....	38
12.3	Suspend and Resume Job operations.....	39
12.3.1	Suspend-Current-Job operation.....	39
12.3.2	Resume-Job operation.....	40
12.4	Promote-Job operation.....	41
12.5	Redirect-Job operation.....	41
12.6	Schedule-Job-After operation.....	43
13	Conformance Requirements.....	45
14	IANA Considerations.....	46
15	Internationalization Considerations.....	46
16	Security Considerations.....	46
17	Author's Addresses.....	46
18	References.....	47
19	Appendix A : Full Copyright Statement.....	48

List of Tables

Table 1 - List of Printer Operations and corresponding Device Operations	11
Table 2 - Forwarding operations that affect Printer objects.....	17
Table 3 - Forwarding operations that affect Jobs objects.....	18
Table 4 - Operation attribute support for Printer Operations.....	20
Table 5 - Operation attribute support for Job operations.....	21
Table 6 - Printer Operation Operation-Id assignments.....	24
Table 7 - Pause and Resume Printer and Device Operations.....	28
Table 8 - Job operation Operation-Id assignments.....	36
Table 9 - Conformance Requirement Dependencies for Operations.....	45
Table 10- Conformance Requirement Dependencies for "printer-state- reasons" Values.....	45

Table 11- Conformance Requirement Dependencies for "job-state-reasons"	
Values.....	46

List of Figures

Figure 1 - Embedded Printer object.....	15
Figure 2 - Hosted Printer object.....	15
Figure 3 - Output Device fan out.....	15

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

Figure 4 - Chained IPP Printer.....	16
Figure 5 - IPP Printer fan out.....	16
Figure 6 - IPP Printer fan in.....	16

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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. This document is a registration proposal for an extension to IPP/1.0 and IPP/1.1 following the registration procedures in those documents.

2 Terminology

This section defines terminology used throughout this document.

2.1 Conformance Terminology

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

The following specialization of these terms apply to this document:

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

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

2.2 Other terminology

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.

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 than one output-device.

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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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

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

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

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

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

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

3 Requirements and Use Cases

The following requirements and usage cover both the "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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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**

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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-**

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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

3.1 List of the Printer and Device Operations

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

Table 1 - List of Printer Operations and corresponding Device Operations

Printer Operation	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.

4 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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

is used to represent other IPP Printer objects. Fan-in is where several IPP Printer objects are used to represent another IPP Printer object.

4.1 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 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.

4.2 IPP Printer Fan-In

The IPP/1.1 Model and Semantics did not preclude the semantic concept of multiple IPP Printer objects that represent a single Output Device (see [[ipp-mod](#)] [section 2.1](#)). However, there was no way for the client to determine that there was a fan-in configuration, nor was there a way to perform operations on the Subordinate device. This specification

generalizes the semantics of the Printer object to allow several Non-Leaf IPP Printer objects to represent a single Subordinate Printer object. Thus a Non-Leaf Printer object MAY share a Subordinate Printer object with one or more other Non-Leaf Printer objects in order to represent IPP Printer fan-in.

As with fan-out (see [section 4.1](#)), 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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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.

4.3 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".

4.4 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).

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

4.5 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.

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

4.6 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

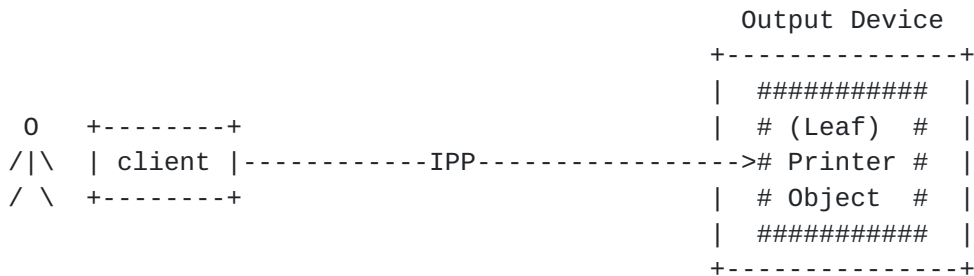


Figure 1 - Embedded Printer object

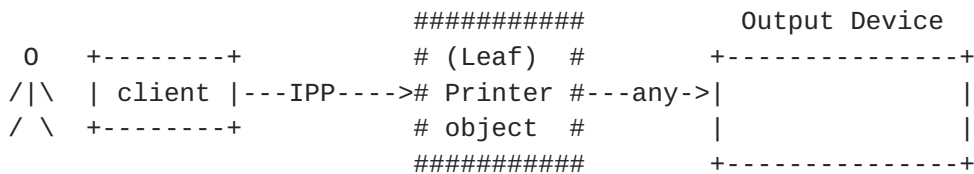
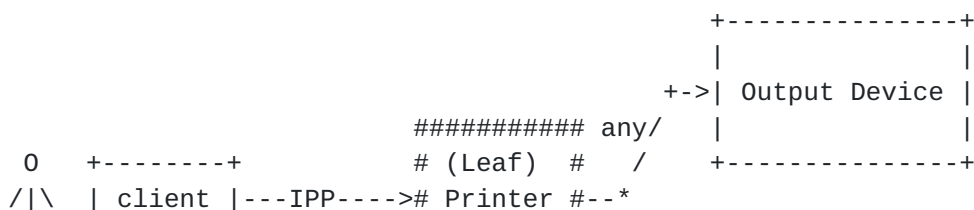


Figure 2 - Hosted Printer object



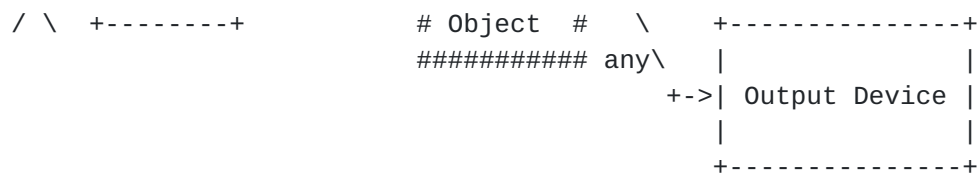


Figure 3 - Output Device fan out

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

```

                                #####
0  +-----+                  # 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

```

                                +-----IPP----->#####
                                /
                                /
                                /
0  +-----+                  ##### any                  #####
/|\  | client |---IPP----># Printer #--*
/\   +-----+                  # object # \
                                \          #####
                                \          # 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 fan out

```

                                (Non-Leaf)
                                #####
                                # Non-Leaf#
                                +----># Printer #-+
                                /   # object # \
                                IPP ##### \          #####
0  +-----+                  /          +-IPP-># subord. #
/|\  | client |--+-IPP-----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 fan in

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

[4.7](#) Forwarding requests

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

[4.7.1](#) 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 6.1](#)).

Table 2 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 0](#) and Table 3):

Table 2 - 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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

Forwarding requests that affect Jobs

Unlike Printer Operations that only affect Printer objects (see [section 0](#)), 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 3 lists the operations that directly affect jobs and the forwarding behavior that a Non-Leaf Printer **MUST** exhibit to its Subordinate Printers:

Table 3 - 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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

Get-Jobs	MUST forward to all its Subordinate Printers
Purge-Jobs	MUST forward to all its Subordinate Printers

IPP/1.1 Job operations:

Send-Document	MUST forward immediately or queue to the appropriate Job in one of its Subordinate Printers
Send-URI	MUST forward immediately or queue to the appropriate Job in one of its Subordinate Printers
Cancel-Job	MUST forward to the appropriate Job in one of its Subordinate Printers
Get-Job-Attributes	MUST forward to the appropriate Job in one of its Subordinate Printers, if the Non-Leaf Printer doesn't know the complete status of the Job object
Hold-Job	MUST forward to the appropriate Job in one of its Subordinate Printers
Release-Job	MUST forward to the appropriate Job in one of its Subordinate Printers
Restart-Job	MUST forward to the appropriate Job in one of its Subordinate Printers
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 [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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

the authenticated original user, not the parent Printer's authenticated host, and is forwarded by each client without changing the value.

5 New Operation attributes

This section summarizes the usage of the new "printer-message-from-operation" and "job-message-from-operator" operation attributes that set the corresponding Printer and Job Description 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. See [ipp-set-ops] for the definition of these operation attributes.

Table 4 shows the operation attributes that are defined for use with the Printer Operations.

Legend:

R - REQUIRED for a Printer to support

O - 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 4 - 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- Printe r- Attrib utes, Set- Printe r- Attrib utes	Enabl e- Print , Disab le- Print er	Rest art- Prin ter	Shut down - Prin ter, Star tup- Prin ter
attributes- charset	R	R	R	R	R	R	R
attributes- natural- language	R	R	R	R	R	R	R
printer-uri	R	R	R	R	R	R	R

requesting- user-name	R	R	R	R	R	R	R
printer- message- from- operator	0	0	0		0	0	0

Table 5 shows the operation attributes that are defined for use with the Job operations.

Legend:

R - REQUIRED for a Printer to support

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

0 - 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 5 - Operation attribute support for Job operations

Operation Attribute	Can cel - Job	Canc el- Curr ent- Job	Hol d- Job , Rel eas e- Job	Sus pen d- Cur ren t- Job	Re su me - Jo b Attrib utes	Get- Job- Attrib utes, Set- Job- Attrib utes	Rest art- Job	Repr oces s- Job	Pro mot e- Job	Red ire ct- Job	Sch edu le- Job - Aft er
attributes-charset	R	R	R	R	R	R	R	R	R	R	R
attributes-natural-language	R	R	R	R	R	R	R	R	R	R	R
printer-uri	R	R	R	R	R	R	R	R	R	R	R
job-uri	R		R		R	R	R	R	R	R	R
job-id	R	R	R	R	R	R	R	R	R	R	R
requesting-user-name	R	R	R	R	R	R	R	R	R	R	R
job-message-from-operator	0	0	0	0	0		0	0	0	0	0
message [to-operator]	0		0	0	0		0	0	0	0	0
job-hold-until			0*					0**			

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

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

6 New Printer Description Attributes

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

[6.1 subordinate-printers-supported \(1setOf uri\)](#)

This Printer attribute is REQUIRED if an implementation supports Subordinate Printers (see [section 4](#)) 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 4.4](#)).

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.

[6.2 parent-printers-supported \(1setOf uri\)](#)

This Printer attribute is REQUIRED if an implementation supports Subordinate Printers (see [section 4](#)) 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.

[6.3 redirection-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).

7 Additional Values for "printer-state-reasons"

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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

[7.1](#) 'hold-new-jobs'

'hold-new-jobs': The operator has issued the Hold-New-Jobs operation (see [section 0](#)) 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.

[7.2](#) 'deactivated'

'deactivated': A client has issued a Deactivate-Printer operation for the Printer object (see [section 0](#)) 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 0](#)) and return the 'server-error-service-unavailable' status code.

[8](#) Additional Values for "job-state-reasons"

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

[8.1](#) '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.

[9](#) 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 4.7](#)) failed.

10 Additional status codes

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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

[10.1](#) 'server-error-printer-is-deactivated' (0x????)

The Printer has been deactivated using the Deactivate-Printer operation and is only accepting the Activate-Printer (see [section 0](#)), 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.

[11](#) 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 are summarized in Table 6:

Table 6 - Printer Operation Operation-Id assignments

Operation Name	Operation-Id	Brief description
Enable-Printer	0x??	Allows the target Printer to accept Job Creation operations
Disable-Printer	0x??	Prevents the target Printer from accepting Job Creation operations
Pause-Printer-After-Current-Job	0x??	Pause the Printer after the current job has been sent to the Output Device.
Hold-New-Jobs	0x??	Finishes processing all currently pending jobs. Any new jobs are placed in the 'pending-held' state.
Release-Held-New-Jobs	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	0x??	Puts the Printer into a read-only deactivated state.
Activate-Printer	0x??	Restores the Printer to normal activity
Restart-Printer	0x??	Restarts the target Printer and re-

		initializes the software
Shutdown-Printer	0x??	Shuts down the target Printer so that it cannot be restarted or queried
Startup-Printer	0x??	Starts up the instance of the Printer object

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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

11.1 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 11.2](#)) 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).

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 5](#)).

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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 5](#)).

[11.2](#) 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 11.1](#)) 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).

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

This document and [[ipp-device-ops](#)] define distinct operations in order to disambiguate the Pause-Printer operation as shown in Table 7. 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 7 - 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.

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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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 0](#)).

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](#)).

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

The Pause-Printer-After-Current-Job Request and Pause-Printer-After-Current-Job Response have the same attribute groups and attributes as the Pause-Printer operation (see [[ipp-mod](#)] sections [3.2.7.1](#) and [3.2.7.2](#)), including the new "printer-message-from-operator" operation attribute (see [section 5](#)).

[11.3](#) Hold and Release New Jobs operations

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

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 0](#)), 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 5](#)).

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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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 5](#)).

[11.4](#) Deactivate and Activate Printer Operations

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

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

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

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 0](#)) and return the 'server-error-service-unavailable' status code.

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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 5](#)).

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 5](#)).

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

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

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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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 5](#)).

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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 0](#)) 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 0](#)), 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 11.1](#)) 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 0](#)) 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 0](#)) 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 5](#)).

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 0](#)) 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.

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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 5](#)).

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

12 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 are summarized in Table 8:

Table 8 - Job operation Operation-Id assignments

Operation Name	Operation-Id	Brief description
Reprocess-Job	0x??	Creates a copy of a completed target job with a new Job ID and processes it
Cancel-Current-Job	0x??	Cancels the current job on the target Printer or the specified job if it is the current job
Suspend-Current-Job	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	0x??	Resume the suspended target job
Promote-Job	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	0x??	Schedule the target job immediately after the specified job, all other scheduling factors being equal.

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

12.1 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.

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

12.2 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 5](#)), 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.

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

12.3 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.

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 8.1](#)) in the job's "job-state-reasons" attribute and processes other jobs.

If the client does not supply a "job-id" operation attribute, the Printer MUST accept the request and 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](#)).

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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 5](#)), 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 support this operation attribute, if it supports this operation.

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 5](#)).

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

12.4 Promote-Job operation

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

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

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

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

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 5](#)).

12.5 Redirect-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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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 preserve 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 of 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 any 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 has 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.

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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).

[12.6](#) 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 job, all other scheduling factors being equal.

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 is ephemeral, rather than setting an attribute of either of

the jobs.

If the target job is not in the 'pending' state, the Printer MUST reject the request and returns the 'client-error-not-possible' status code, since the job cannot have its position changed. The predecessor job can be in the 'pending', 'processing', or 'processing-stopped' states.

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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 5](#)). 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. If the client omits this attribute, the Printer MUST schedule 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 not in the 'pending', 'processing', or 'processing-stopped' states.

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

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

13 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:
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

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-

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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

14 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

15 Internationalization Considerations

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

16 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.

17 Author's Addresses

Carl Kugler

IBM

Boulder CO

Phone: (303) 924-5060

FAX:

e-mail: kugler@us.ibm.com

Tom Hastings

Kugler, Hastings, Lewis

[Page 46]

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

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

18 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-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.

Expires: January 6, 2001

INTERNET-DRAFT

IPP/1.1: Set 2 Operations

July 6, 2000

19 Appendix A: Full Copyright Statement

Copyright (C) The Internet Society (1998,1999). All Rights Reserved

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Expires: January 6, 2001