Internet Printing Protocol WG
INTERNET-DRAFT
<draft-ietf-ipp-ops-set2-04.txt>
Updates: RFC 2911, RFC 3380
[Target Category: standards track
Expires: January 15, 2005

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

July 15, 2004

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



## Status of this Memo



## Abstract

This document specifies the following 16 additional OPTIONAL system administration operations for use with the Internet Printing Protocol/1.1 (IPP) [RFC2910, RFC2911]:

| 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 |
| Deactivate-Printer and Activate-Printer | Resume-Job |
| Restart-Printer | Promote-Job |
| Shutdown-Printer and Startup-Printer | Schedule-Job-After |

plus a few associated attributes, values, and status codes and using the IPP Printer object to manage printer fan-out and fan-in.

# Table of Contents

## List of Tables

## List of Figures

# 1 Introduction

The Internet Printing Protocol (IPP) is an application level protocol that can be used for distributed printing using Internet tools and technologies. IPP version 1.1 ([RFC2911, RFC2910]) focuses on end user functionality with a few administrative operations included. This document defines additional OPTIONAL end user, operator, and administrator operations used to control Jobs and Printers. In addition, this document extends the semantic model of the Printer object by allowing them to be configured into trees and/or inverted trees that represent Printer object Fan-Out and Printer object Fan-In, respectively. The special case of a tree with only a single Subordinate node represents Chained Printers. This document is a registration proposal for an extension to IPP/1.0 and IPP/1.1 following the registration procedures in those documents.

The requirements and use cases for this document are defined in [RFC3239].

# 2 Terminology

This section defines terminology used throughout this document.

## 2.1 Conformance Terminology

Capitalized terms, such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED NOT, and OPTIONAL, have special meaning relating to conformance as defined in RFC 2119 [RFC2119] and [RFC2911] section 12.1. If an implementation supports the extension defined in this document, then these terms apply; otherwise, they do not. These terms define conformance to *this document only*; they do not affect conformance to other documents, unless explicitly stated otherwise.

## 2.2 Other terminology

This document uses terms such as "**client**", "**Printer**", "**Job**", "**attributes**", "**keywords**", "**operation**" and "**support**". These terms have special meaning and are defined in the model terminology [RFC2911] section 12.2.

In addition, the following capitalized terms are defined:

**IPP Printer object (**or **Printer** for short**) -** a software abstraction defined by [RFC2911].

**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 [RFC2911]), there is no formal object called an **Output Device** defined in this document (or [RFC2911]).

**Output Device Fan-Out** - a configuration in which an IPP **Printer** controls more that one **Output Device**.

**Printer Fan-Out** - a configuration in which an IPP **Printer** object controls more than one **Subordinate** IPP **Printer** object.

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

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

**Leaf Printer** - an IPP **Printer** object that has no **Subordinate Printers**.

**Non-Leaf Printer** - an IPP **Printer** object that has one or more **Subordinate Printers**. A **Non-Leaf Printer** is also called a **Parent Printer**.

**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 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 [RFC2911] section 3.1.

The Printer Operations defined in this document are summarized in Table 1:

**Table 1 - Printer Operation Operation-Id assignments**

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

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

## 3.1 The Disable and Enable Printer Operations

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

These operations allow the operator to control whether or not the Printer will accept new Job Creation (Print-Job, Print-URI, and Create-Job) operations. These operations have no other effect on the Printer, so that the Printer continues to accept all other operations and continues to schedule and process jobs normally. In other words, these operation control the "input of new jobs" to the IPP Printer while the Pause and Resume operations (see section 3.2) independently control the "output of new jobs" from the IPP Printer to the Output Device.

### 3.1.1 Disable-Printer Operation

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

The IPP Printer MUST accept the request in any state. The Printer sets the value of its "printer-is-accepting-jobs" READ-ONLY Printer Description attribute to 'false' (see [RFC2911] 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 [RFC2911] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [RFC2911] 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 [RFC2911] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see section 6).

### 3.1.2 Enable-Printer Operation

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

The IPP Printer MUST accept the request in any state. The Printer sets the value of its "printer-is-accepting-jobs" READ-ONLY Printer Description attribute to 'true' (see [RFC2911] 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 [RFC2911] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [RFC2911] 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 [RFC2911] sections 3.2.8.1 and 3.2.8.2), including the new "printer-message-from-operator" operation attribute (see section 6).

## 3.2 The Pause and Resume Printer Operations

This section leaves the OPTIONAL IPP/1.1 Pause-Printer (see [RFC2911] 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-Current-Job operation to be after the current job. These operations

affect the scheduling of IPP jobs. If either of these Pause Printer operations are supported, then the Resume-Printer operation MUST be supported.

These operations allow the operator to control whether or not the Printer will send new IPP jobs to the associated Output Device(s) that the IPP Printer object represents. These operations have no other effect on the Printer, so that the Printer continues to accept all operations. In other words, these operation control the "output of new jobs" to the Output Device(s) while the Disable and Enable Printer Operations (see section 3.1) independently control the "input of new jobs" to the IPP Printer.

**Table 2 - Pause and Resume Printer Operations**

| Pause and Resume Printers | 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 [RFC2911]. |
| 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. |

### 3.2.1 Pause-Printer-After-Current-Job operation

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

If the IPP Printer is not sending an IPP Job to the Output Device or Subordinate Printer (whether or not the Output Device or Subordinate Printer is busy processing any jobs), the IPP Printer object transitions immediately to the 'stopped' state by setting its "printer-state" attribute to 'stopped', removing the 'moving-to-paused' value, if present, from its "printer-state-reasons" attribute, and adding the 'paused' value to its "printer-state-reasons" attribute.

If the implementation will take appreciable time to complete sending an IPP job that it has started sending to an Output Device or Subordinate Printer, the IPP Printer adds the 'moving-to-paused' value to the Printer object's "printer-state-reasons" attribute (see section [RFC2911] 4.4.12). When the IPP Printer has completed sending IPP jobs that it was in the process of sending, the Printer object transitions to the 'stopped' state by setting its "printer-state" attribute to 'stopped', removing the 'moving-to-paused' value, if present, from its "printer-state-reasons" attribute, and adding the 'paused' value to its "printer-state-reasons" attribute.

This operation MUST NOT affect the acceptance of Job Creation requests (see Disable-Printer section 3.1.1).

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

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

**Table 3 - State Transition Table for Pause-Printer-After-Current-Job operation**

| Current "printer-state" | New "printer-state" | "printer-state-reasons" | IPP Printer's response status code and action:<br><br>REQUIRED/OPTIONAL state transition for a Printer to support |
|---|---|---|---|
| 'idle' | 'stopped' | 'paused' | REQUIRED: 'successful-ok' |
| 'processing' | 'processing' | 'moving-to-paused' | OPTIONAL: 'successful-ok';<br>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 [RFC2911] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [RFC2911] Sections 1 and 8.5).

The Pause-Printer-After-Current-Job Request and Pause-Printer-After-Current-Job Response have the same attribute groups and attributes as the Pause-Printer operation (see [RFC2911] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see section 6).

## 3.3 Hold and Release New Jobs operations

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

### 3.3.1 Hold-New-Jobs operation

This OPTIONAL operation allows a client to condition the Printer to complete the current 'pending' and 'processing' IPP Jobs but not start processing any subsequently created IPP Jobs. If the IPP Printer is in the middle of sending an IPP job to an Output Device or Subordinate Printer, the IPP Printer MUST complete sending that Job. Furthermore, the IPP Printer MUST send all of the current 'pending' IPP Jobs to the Output Device(s) or Subordinate IPP Printer object(s). Any subsequently received Job Creation operations will cause the IPP Printer to put the Job into the 'pending-held' state with the 'job-held-on-create' value being added to the job's "job-state-reasons" attribute. Thus all newly accepted jobs will be automatically held by the Printer.

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

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

The IPP Printer MUST accept the request in any state, MUST NOT transition the Printer to any other "printer-state", and MUST add the 'hold-new-jobs' value to the Printer's "printer-state-reasons" attribute (whether the value was present or not).

*Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [RFC2911] 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 [RFC2911] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see section 6).

### 3.3.2 Release-Held-New-Jobs operation

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

*Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [RFC2911] 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 [RFC2911] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see section 6).

## 3.4 Deactivate and Activate Printer Operations

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

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

### 3.4.1 Deactivate-Printer operation

This OPTIONAL operation allows a client to stop the Printer object from starting to send IPP jobs to any of its Output Devices or Subordinate Printers (Pause-Printer-After-Current-Job) and stop the Printer object from accepting any, but query requests. The Printer performs a Disable-Printer and a Pause-Printer-After-Current-Job operation immediately, including use of all of the "printer-state-reasons" if these two operations cannot be completed immediately. In addition, the Printer MUST immediately reject all requests, except Activate-Printer, queries (Get-Printer-Attributes, Get-Job-Attributes, Get-Jobs, etc.), Send-Document, and Send-URI (so that partial job submission can be completed - see section 3.1.1) and return the 'server-error-service-unavailable' status code.

The IPP Printer MUST accept the request in any state. Immediately, the Printer MUST set the 'deactivated' value in its "printer-state-reasons" attribute. Note: neither the Disable-Printer nor the Pause-Printer-After-Current-Job set the 'deactivated' value.

*Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [RFC2911] 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 [RFC2911] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see section 6).

### 3.4.2 Activate-Printer operation

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

The IPP Printer MUST accept the request in any state. Immediately, the Printer MUST immediately remove the 'deactivated' value from its "printer-state-reasons" attribute (whether present or not).

*Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [RFC2911] 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 [RFC2911] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see section 6).

## 3.5 Restart-Printer, Shutdown-Printer, and Startup-Printer operations

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

### 3.5.1 Restart-Printer operation

This OPTIONAL operation allows a client to restart a Printer object whose operation is in need of initialization because of incorrect or erratic behavior, i.e., perform the effect of a software re-boot. The implementation MUST attempt to save any information about Jobs and the Printer object before re-initializing. However, this operation MAY have drastic consequences on the running system, so the client SHOULD first try the Deactivate-Printer operation 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 [RFC2911] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [RFC2911] 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 [RFC2911] sections 3.2.8.1 and 3.2.8.2), including the new "printer-message-from-operator" operation attribute (see section 6).

### 3.5.2 Shutdown-Printer Operation

This OPTIONAL operation allows a client to shutdown a Printer, i.e., stop processing jobs without losing any jobs and make the Printer object no longer available for any operations using the IPP protocol. There is no way to bring the instance of the Printer object back to being used, except for the Startup-Printer (see section 3.5.3) which starts up a new instance of the Printer object for hosted implementations. The purpose of Shutdown-Printer is to shutdown the Printer for an extended period, not to reset the device(s) or modify a Printer attribute. See Restart-Printer (section 3.5.1) and Startup-Printer (section 3.5.3) for the way to initialize the software. See the Disable-Printer operation (section 3.1) for a way for the client to stop the Printer from accepting Job Creation requests without stopping processing or shutting down.

The Printer MUST add the 'shutdown' value (see [RFC2911] section 4.4.11) immediately to its "printer-state-reasons" Printer Description attribute and performs a Deactivate-Printer operation (see section 3.4.1) which performs a Disable-Printer and Pause-Printer-After-Current-Job operation).

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

The Printer object MUST accept this operation in any state and transition the Printer object through the "printer-states" and "printer-state-reasons" defined for the Pause-Printer-After-Current-Job operation until the activity is completed and the Printer object disappears.

*Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [RFC2911] 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 [RFC2911] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see section 6).

### 3.5.3 Startup-Printer operation

This OPTIONAL operation allows a client to startup an instance of a Printer object, provided that there isn't one already instantiated. The purpose of Startup-Printer is to allow a hosted implementation of the IPP Printer object (i.e., a Server that implements an IPP Printer on behalf of a networked or local Output Device) to be started after the host is available (by means outside this document). See Restart-Printer (section 3.5.1) for the way to initialize the software or reset the Output Device(s) when the IPP Printer object has already been instantiated.

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

The result of this operation MUST be with the Printer object's "printer-state" set to 'idle', the state reasons removed from its "printer-state-reasons" attribute, and its "printer-is-accepting-jobs" attribute set to 'false'. Then the operator can reconfigure the Printer before performing an Enable-Printer operation. However, when a Printer is first powered up, it is RECOMMENDED that its "printer-is-accepting-jobs" attribute be set to 'true' in order to achieve easy "out of the box" operation.

*Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [RFC2911] 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 [RFC2911] sections 3.2.7.1 and 3.2.7.2), including the new "printer-message-from-operator" operation attribute (see section 6).

# 4 Definition of the Job Operations

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

The Job Operations defined in this document are summarized in Table 4:

**Table 4 - Job operation Operation-Id assignments**

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

## 4.1 Reprocess-Job Operation

This OPTIONAL operation is a create job operation that allows a client to re-process a copy of a job that had been retained in the queue after processing completed, was canceled, or was aborted (see [RFC2911] section 4.3.7.2). This operation is the same as the Restart-Job operation (see [RFC2911] section 3.3.7), except that the Printer creates a new job that is a copy of the target job and the target job is unchanged. The new job is assigned new values to the “job-uri” and “job-id” attributes and the new job’s Job Description attributes that accumulate job progress, such as “job-impressions-completed”, “job-media-sheets-completed”, and “job-k-octets-processed”, are initialized to 0 as with any create job operation. The target job moves to the Job History after a suitable period, independent of whether one or more Reprocess-Job operations have been performed on it.

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

## 4.2 Cancel-Current-Job Operation

This OPTIONAL operation allows a client to cancel the current job on the target Printer or the specified job if it is the current job on the Printer. See [RFC2911] 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 [RFC2911] 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 [RFC2911] 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 [RFC2911] section 3.2.8), including the new “job-message-from-operator” operation attribute (see section 6), with the addition of the following Group 1 Operation attributes in the request:

“job-id” (integer(1:MAX)):
The client OPTIONALLY supplies this Operation attribute in order to verify that the identified job is still the current job on the target Printer object. The IPP object MUST supports this operation attribute, if it supports this operation.

## 4.3 Suspend and Resume Job operations

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

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

The Hold-Job and Release-Job operations ([RFC2911] section 3.3.5) are for holding and releasing held jobs, not suspending and resuming suspended jobs.

### 4.3.1 Suspend-Current-Job operation

This OPTIONAL operation allows a client to stop the current job on the target Printer or the specified job if it is the current job on the Printer, and allow other jobs to be processed instead. The Printer moves the current job or the target job to the 'processing-stopped' state and sets the 'job-suspended' value (see section 9.1) in the job's "job-state-reasons" attribute and processes other jobs.

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

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

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

*Access Rights:* The authenticated user (see [RFC2911] 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 [RFC2911] Sections 1 and 8.5).

The Suspend-Current-Job Request and Suspend-Current-Job Response have the same attribute groups and attributes as the Pause-Printer operation (see [RFC2911] section 3.2.8 ), including the new "job-message-from-operator" operation attribute (see section 6), with the addition of the following Group 1 Operation attributes in the request:

"job-id" (integer(1:MAX)):
The client OPTIONALLY supplies this Operation attribute in order to verify that the identified job is still the current job on the target Printer object. The IPP object MUST supports this operation attribute, if it supports this operation.

### 4.3.2 Resume-Job operation

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

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

*Access Rights:* The authenticated user (see [RFC2911] 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 [RFC2911] 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 [RFC2911] section 3.3.6), including the new "job-message-from-operator" operation attribute (see section 6).

## 4.4 Job Scheduling Operations

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

### 4.4.1 Promote-Job operation

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

If the target job is in the 'pending' state, this operation does not change the job's state, but causes the job to be processed after the current job(s) complete. If the target job is not in the 'pending' state, the Printer MUST reject the request and return the 'client-error-not-possible' status code.

If the Printer implements the "job-priority" Job Template attribute (see [RFC2911] section 4.2.1), the Printer sets the job's "job-priority" to the highest value supported (so that the job will print before any of the other pending jobs). The Printer returns the target job immediately after the current job(s) in a Get-Jobs response (see [RFC2911] section 3.2.6) for the 'not-completed' jobs.

When the current job completes, is canceled, suspended (see section 4.3.1), or aborted, the target of this operation is processed next.

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

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

*Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation must be an operator or administrator of the Printer object (see [RFC2911] 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 [RFC2911] section 3.3.3), including the new "job-message-from-operator" operation attribute (see section 6).

### 4.4.2 Schedule-Job-After operation

This OPTIONAL operation allows a client to request the Printer to schedule the target job so that it will be processed immediately after the specified predecessor job, all other scheduling factors being equal. This operation is specially useful in a production printing environment where the operator is involved in job scheduling.

If the target job is in the 'pending' state, this operation does not change the job's state, but causes the job to be processed after the predecessor job completes. The predecessor job can be in the 'pending', 'processing', or 'processing-stopped' states. If the target job is not in the 'pending' state or the predecessor job is not in the 'pending', 'processing', or 'processing-stopped' states, the Printer MUST reject the request and returns the 'client-error-not-possible' status code, since the job cannot have its position changed.

If the Printer implements the "job-priority" Job Template attribute (see [RFC2911] section 4.2.1), the Printer sets the job's "job-priority" to that of the predecessor job (so that the job will print after the predecessor job). The Printer returns the target job immediately after the predecessor in a Get-Jobs response (see [RFC2911] section 3.2.6) for the 'not-completed' jobs.

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

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

IPP is specified not to require queues for job scheduling, since there are other implementation techniques for scheduling multiple jobs, such as re-evaluating a criteria function for each job on a scheduling cycle. However, if an implementation does implement queues for jobs, then the Schedule-Job-After operation puts the specified job immediately after the specified job in the queue. A subsequent Schedule-Job-After operation specifying the same job will cause its target job to be placed after that job, even though it is between the first target job and the specified job. For example, suppose the job queue consisted of jobs: A, B, C, D, and E, in that order. A Schedule-Job-After with job E as the target and B as the specified job would result in the following queue: A, B, E, C, D. A subsequent Schedule-Job-After with Job D as the target and B as the specified job would result in the following queue: A, B, D, E, C. In other words, the link between the two jobs in a Schedule-Job-After operation is not retained, i.e., there is no attribute on either job that points to the other job as a result of this operation.

*Access Rights:* The authenticated user (see [RFC2911] section 8.3) performing this operation must be operator or administrator of the Printer object (see [RFC2911] Sections 1 and 8.5).

The Schedule-Job-After Request have the same attribute groups and attributes as the Cancel-Job operation (see [RFC2911] section 3.3.3), plus the new "job-message-from-operator" operation attribute (see section 6). In addition, the following operation attributes are defined:

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

The Schedule-Job-After Response has the same attribute groups, attributes, and status codes as the Cancel-Job operation (see [RFC2911] section 3.3.3). The following status codes have particular meaning for this operation:

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

# 5 Additional status codes

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

## 5.1 'server-error-printer-is-deactivated' (0x050A)

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

# 6 Use of Operation Attributes that are Messages from the Operator

This section summarizes the usage of the "printer-message-from-operator" and "job-message-from-operator" operation attributes [RFC3380] that set the corresponding Printer and Job Description attributes (see [RFC2911] for the definition of these Description attributes). These operation attributes are defined for most of the Printer and Job operations that operators are likely to perform, respectively, so that operators can indicate the reasons for their actions.

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

**Table 5 - Operation attribute support for Printer Operations**

| Operation Attribute | | | | A | B | | |
|---|---|---|---|---|---|---|---|
| attributes-charset | | | | REQ | REQ | | |
| attributes-natural-language | | | | REQ | REQ | | |
| printer-uri | | | | REQ | REQ | | |
| requesting-user-name | | | | REQ | REQ | | |
| printer-message-from-operator | | | | Note | OPT | | |

Legend:
A: Get-Printer-Attributes, Set-Printer-Attributes
B: All other Printer administrative operations, including, but not limited to: Pause-Printer, Pause-Printer-After-Current-Job, Resume-Printer, Hold-New-Jobs, Release-Held-New-Jobs, Purge-Jobs, , Enable-Print, Disable-Printer, Restart-Printer, Shutdown-Printer, and Startup-Printer.

REQ - REQUIRED for a Printer to support
OPT - OPTIONAL for a Printer to support; the Printer ignores the attribute if not supported
Note - According to [RFC3380], the Client MUST NOT supply the “printer-message-from-operator” operation attribute in a Get-Printer-Attributes or Set-Printer-Attributes operation; the Printer MUST ignore this operation attribute in these two operations. Instead, the client when used by an operator MUST supply the “printer-message-from-operator” as (one of the) explicit attributes being set on the Printer object with the Set-Printer-Attributes operation.

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

**Table 6 - Operation attribute support for Job operations**

| Operation Attribute | A | B | C | | | F | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| attributes-charset | REQ | REQ | REQ | | | REQ | | | | |
| attributes-natural-language | REQ | REQ | REQ | | | REQ | | | | |
| printer-uri | REQ | REQ | REQ | | | REQ | | | | |
| job-uri | REQ | | REQ | | | REQ | | | | |
| job-id | REQ | REQ | REQ | | | REQ | | | | |
| requesting-user-name | REQ | REQ | REQ | | | REQ | | | | |
| job-message-from-operator | OPT | OPT | OPT | | | Note | | | | |
| message*** | OPT | OPT | OPT | | | n/a | | | | |
| job-hold-until | n/a | n/a | OPT* | | | n/a | | | | |

Legend:
A: Cancel-Job, Resume-Job, Restart-Job, Promote-Job, Schedule-Job-After
B: Cancel-Current-Job, Suspend-Current-Job
C: Hold-Job, Release-Job, Reprocess-Job
F: Get-Job-Attributes, Set-Job-Attributes

REQ - REQUIRED for a Printer to support
OPT - OPTIONAL for a Printer to support; the Printer ignores the attribute if supplied, but not supported
n/a - not applicable for use with the operation; the Printer ignores the attribute
Note - According to [RFC3380], the Client MUST NOT supply the “job-message-from-operator” operation attribute in a Get-Job-Attributes or Set-Job-Attributes operation; the Printer MUST ignore this operation attribute in these two operations. Instead, the client when used by an operator MUST supply the “job-message-from-operator” as (one of the) explicit attributes being set on the Job object with the Set-Job-Attributes operation.
* - The Printer MUST support the “job-hold-until” operation attribute if it supports the “job-hold-until” Job Template attribute. For the Reprocess-Job operation the client can hold the job and then modify the job before releasing it to be processed.
** - In [RFC2911] the “message” operation attribute is defined to contain a message *to* the operator but [RFC2911] does not define a Job Description to store the message.

# 7 New Printer Description Attributes

The following new Printer Description attributes are needed to support the new operations defined in this document and the concepts of Printer Fan-Out (see section 10).

### 7.1 subordinate-printers-supported (1setOf uri)

This Printer attribute is REQUIRED if an implementation supports Subordinate Printers (see section 10) 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 [RFC2911] section 4.1), depending on implementation.

The precise format of the Subordinate Printer URIs is implementation dependent (see section 10.4).

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" attribute to the Job object's "output-device-assigned" attribute (see [RFC2911] section 4.3.13). The "output-device-assigned" Job attribute identifies the Output Device to which the Printer object has assigned a job, for example, when a single Printer object is supporting Device Fan-Out or Printer Fan-Out.

### 7.2 parent-printers-supported (1setOf uri)

This Printer attribute is REQUIRED if an implementation supports Subordinate Printers (see section 10) 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 [RFC2911] section 4.1), depending on implementation.

## 8 Additional Values for the "printer-state-reasons" Printer Description attribute

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

### 8.1 'hold-new-jobs' value

'hold-new-jobs': The operator has issued the Hold-New-Jobs operation (see section 3.3.1) or other means, but the output-device(s) are taking an appreciable time to stop. Later, when all output has stopped, the "printer-state" becomes 'stopped', and the 'paused' value replaces the 'moving-to-paused' value in the "printer-state-reasons" attribute. This value MUST be supported, if the Hold-New-Jobs operation is supported and the implementation takes significant time to pause a device in certain circumstances.

### 8.2 'deactivated' value

'deactivated': A client has issued a Deactivate-Printer operation for the Printer object (see section 3.4.1) and the Printer is in the process of becoming deactivated or has become deactivated. The Printer MUST reject all requests except Activate-Printer, queries (Get-Printer-Attributes, Get-Job-

673 Attributes, Get-Jobs, etc.), Send-Document, and Send-URI (so that partial job submission can be
674 completed - see section 3.1.1) and return the ‘server-error-service-unavailable’ status code.

# 675 9 Additional Values for the “job-state-reasons” Job Description attribute

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

## 677 9.1 ‘job-suspended’ value

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

# 681 10 Use of the Printer object to represent IPP Printer Fan-Out and IPP Printer 682 Fan-In

683 This section defines how the Printer object MAY be used to represent IPP Printer Fan-Out and IPP
684 Printer Fan-In. Fan-Out is where an IPP Printer is used to represent other IPP Printer objects. Fan-In
685 is where several IPP Printer objects are used to represent another IPP Printer object.

## 686 10.1 IPP Printer Fan-Out

687 The IPP/1.1 Model and Semantics introduces the semantic concept of an IPP Printer object that
688 represents more than one Output Device (see [RFC2911] section 2.1). This concept is called “Output
689 Device Fan-Out”. However, there was no way to represent the individual states of the Output Devices
690 or to perform operations on a specific Output Device when there was Fan-Out. This document
691 generalizes the semantics of the Printer object to represent such Subordinate Fan-Out Output Devices
692 as IPP Printer objects. This concept is called “Printer object Fan-Out”. A Printer object that has a
693 Subordinate Printer object is called a Non-Leaf Printer object. Thus a Non-Leaf Printer object
694 supports one or more Subordinate Printer objects in order to represent Printer object Fan-Out. A
695 Printer object that does not have any Subordinate Printer objects is called a Leaf Printer object.

696 Each Non-Leaf Printer object submits jobs to its immediate Subordinate Printers and otherwise
697 controls the Subordinate Printers using IPP or other protocols. Whether pending jobs are kept in the
698 Non-Leaf Printer until a Subordinate Printer can accept them or are kept in the Subordinate Printers
699 depends on implementation and/or configuration policy. Furthermore, a Subordinate Printer object
700 MAY, in turn, have Subordinate Printer objects. Thus a Printer object can be both a Non-Leaf Printer
701 and a Subordinate Printer.

702 A Subordinate Printer object MUST be a conforming Printer object, so it MUST support all of the
703 REQUIRED [RFC2911] operations and attributes. However, with access control, the Subordinate
704 Printer MAY be configured so that end-user clients are not permitted to perform any operations (or just
705 Get-Printer-Attributes) while one or more Non-Leaf Printer object(s) are permitted to perform any
706 operation.

## 10.2 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 [RFC2911] 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 10.1), when a Printer object is a Non-Leaf Printer, it MUST NOT have an associated Output Device. As with Fan-Out, a Leaf Printer object has one or more associated Output Devices. As with Fan-Out, the Non-Leaf Printer objects submit jobs to their Subordinate Printer objects and otherwise control the Subordinate Printer. As with Fan-Out, whether pending jobs are kept in the Non-Leaf Printers until the Subordinate Printer can accept them or are kept in the Subordinate Printer depends on implementation and/or configuration policy.

## 10.3 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”.

## 10.4 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 MAY 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-ntfy] for means to subscribe to event notification when the Subordinate Printer object supports IPP notification). Alternatively, the parent Printer MAY wait until its “printer-state” and “printer-state-reasons” attributes are queried and then query all its Subordinate Printers in order to return the correct values.

## 10.5 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.

766

## 767 10.6 Figures to show all possible configurations

768 Figure 1, Figure 2, and Figure 3 are taken from [RFC2911] to show the configurations possible with
769 IPP/1.0 and IPP/1.1 where all Printer objects are Leaf Printer objects. The remaining figures show
770 additional configurations that this document defines using Non-Leaf and Leaf Printer objects. Legend
771 for all figures:

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

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

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

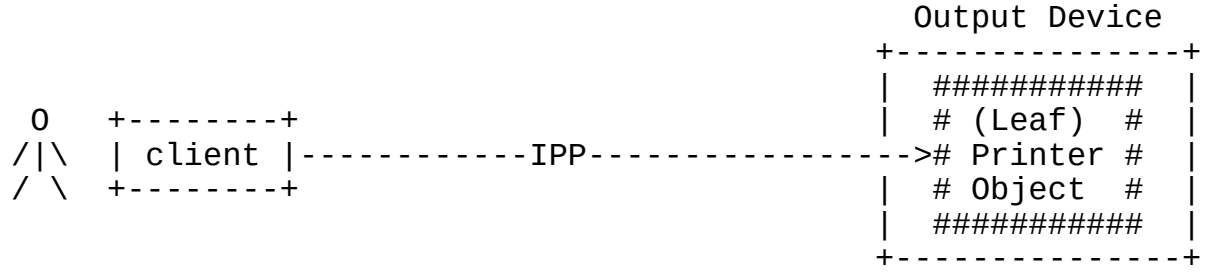



**Figure 1 - Embedded Printer object**

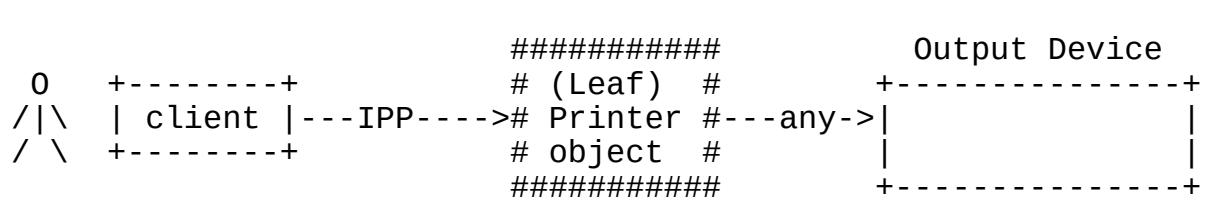



**Figure 2 - Hosted Printer object**

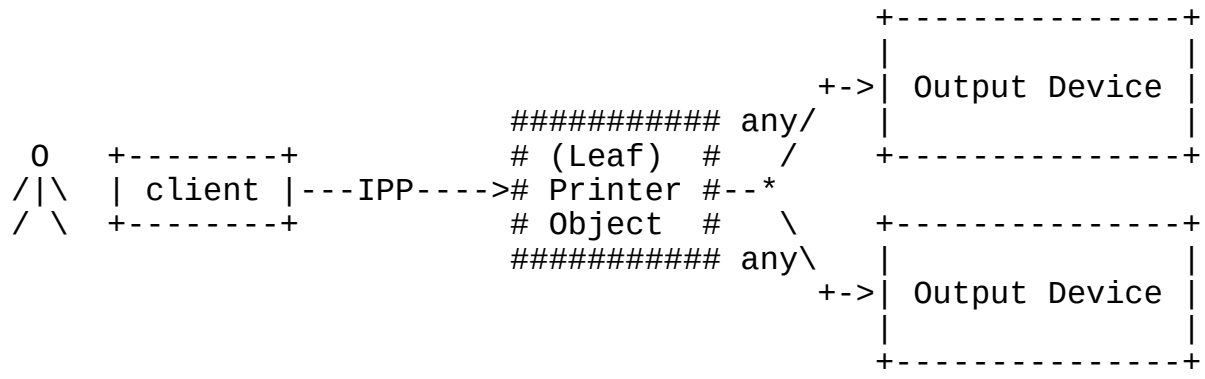
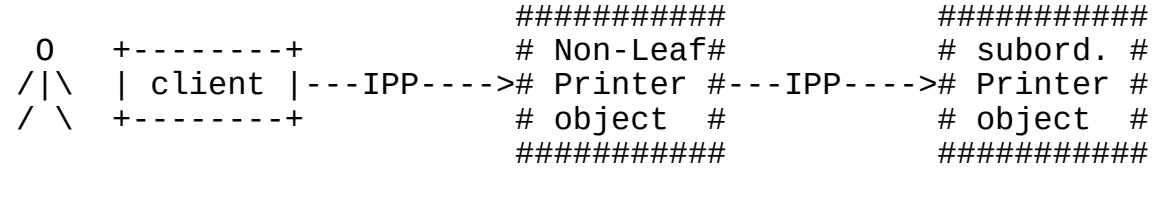
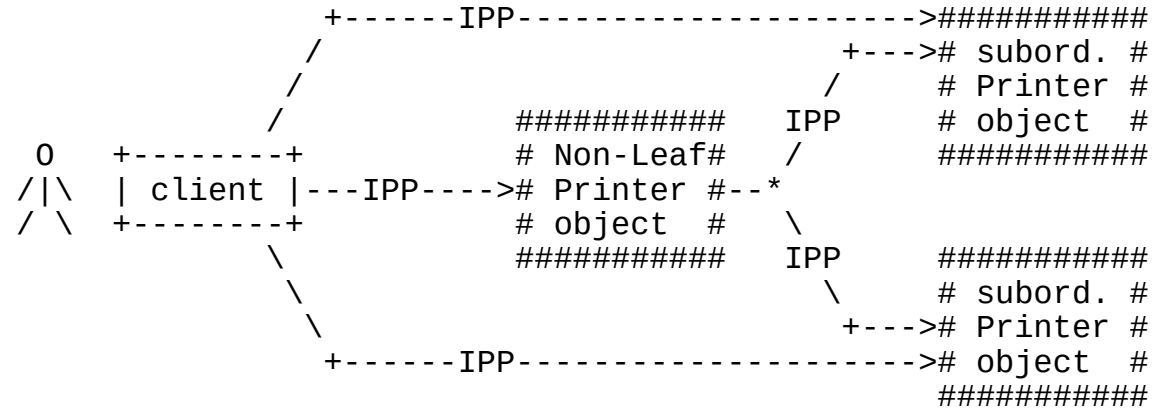



**Figure 3 - Output Device Fan-Out**



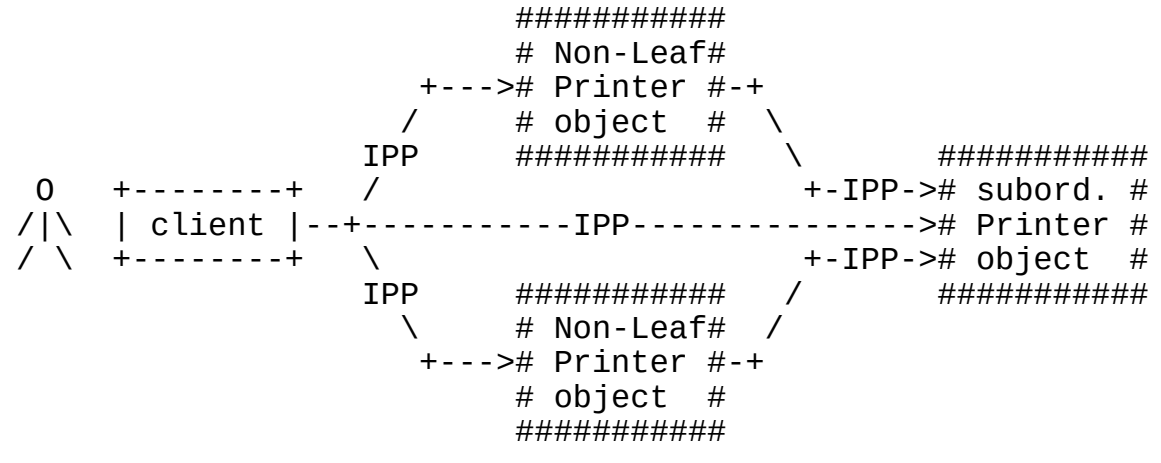

The Subordinate Printer can be a Non-Leaf Printer as in Figure 4 to Figure 6, or can be a Leaf Printer as in Figure 1 to Figure 3.

**Figure 4 - Chained IPP Printer Objects**




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

**Figure 5 - IPP Printer Object Fan-Out**




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 6 - IPP Printer Object Fan-In**

## 10.7 Forwarding requests

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

### 10.7.1 Forwarding requests that affect Printer objects

In Printer Fan-Out, Printer Fan-In, and Chained Printers, the Non-Leaf IPP Printer object MUST NOT forward the operations that affect Printer objects to its Subordinate Printer objects. If a client wants to explicitly target a Subordinate Printer, the client MUST specify the URI of the Subordinate Printer. The client can determine the URI of any Subordinate Printers by querying the Printer's "subordinate-printers-supported (1setOf uri) attribute (see section 7.1).

Table 7 lists the operations that affect Printer objects and the forwarding behavior that a Non-Leaf Printer MUST exhibit to its immediate Subordinate Printers. Operations that affect jobs have a different forwarding rule (see section 10.7.2 and Table 8):

**Table 7 - Forwarding operations that affect Printer objects**

| Printer Operation | Non-Leaf Printer action |
|---|---|
| Printer Operations: | |
| Enable-Printer | MUST NOT forward to any of its Subordinate Printers |
| Disable-Printer | MUST NOT forward to any of its Subordinate Printers |
| Hold-New-Jobs | MUST NOT forward to any of its Subordinate Printers |
| Release-Held-New-Jobs | MUST NOT forward to any of its Subordinate Printers |
| Deactivate-Printer | MUST NOT forward to any of its Subordinate Printers |
| Activate-Printer | MUST NOT forward to any of its Subordinate Printers |
| Restart-Printer | MUST NOT forward to any of its Subordinate Printers |
| Shutdown-Printer | MUST NOT forward to any of its Subordinate Printers |
| Startup-Printer | MUST NOT forward to any of its Subordinate Printers |
| IPP/1.1 Printer Operations: | See [RFC2911] |
| 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 [RFC3380] |
| Set-Printer-Attributes | MUST NOT forward to any of its Subordinate Printers |

### 10.7.2 Forwarding requests that affect Jobs

Unlike Printer Operations that only affect Printer objects (see section 10.7.1), a Non-Leaf Printer object MUST forward operations that directly affect jobs to the appropriate Job object(s) in one or more of its immediate Subordinate Printer objects. Forwarding is REQUIRED since the purpose of such a Job operation is to affect the indicated job which itself may have been forwarded. Such

forwarding MAY be immediate or queued, depending on the operation and the implementation. For example, a Non-Leaf Printer object MAY queue/spool jobs, feeding a job at a time to its Subordinate Printer(s), or MAY forward jobs immediately to one of its Subordinate Printers. In either case, the Non-Leaf Printer object is forwarding Job Creation operations to one of its Subordinate Printers. Only the time of forwarding of the Job Creation operations depends on whether the policy is to queue/spool jobs in the Non-Leaf Printer or the Subordinate Printer.

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

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

885 **Table 8 - Forwarding operations that affect Jobs objects**

| Job operation | Non-Leaf Printer action |
|---|---|
| Job operations: | |
| Reprocess-Job | MUST forward to the appropriate Job in one of its Subordinate Printers |
| Cancel-Current-Job | MUST NOT forward |
| Resume-Job | MUST forward to the appropriate Job in one of its Subordinate Printers |
| Promote-Job | MUST forward to the appropriate Job in one of its Subordinate Printers |
| IPP/1.1 Printer Operations: | |
| Print-Job | MUST forward immediately or queue to the appropriate Subordinate Printer |
| Print-URI | MUST forward immediately or queue to the appropriate Subordinate Printer |
| Validate-Job | MUST forward to the appropriate Subordinate Printer |
| Create-Job | MUST forward immediately or queue to the appropriate Subordinate Printer |
| Get-Jobs | MUST forward to *all* its Subordinate Printers |
| Purge-Jobs | MUST forward to *all* its Subordinate Printers |
| IPP/1.1 Job operations: | |
| Send-Document | MUST forward immediately or queue to the appropriate Job in one of its Subordinate Printers |
| Send-URI | MUST forward immediately or queue to the appropriate Job in one of its Subordinate Printers |
| Cancel-Job | MUST forward to the appropriate Job in one of its Subordinate Printers |
| Get-Job-Attributes | MUST forward to the appropriate Job in one of its Subordinate Printers, if the Non-Leaf Printer doesn't know the complete status of the Job object |
| Hold-Job | MUST forward to the appropriate Job in one of its Subordinate Printers |
| 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 [RFC3380] |
| Set-Job-Attributes | MUST forward to the appropriate Job in one of its Subordinate Printers |

886

887 When a Printer receives a request that REQUIRES forwarding, it does so on a "best efforts basis", and
888 returns a response to its client without waiting for responses from any of its Subordinate Printers.
889 Such forwarded requests could fail.

## 890 10.8 Additional attributes to help with fan-out

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

### 10.8.1 output-device-assigned (name(127)) Job Description attribute - from [RFC2911]

[RFC2911] defines “output-device-assigned” as: “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 “Device Fan-Out” see [RFC2911] section 2.1).” See also section 10.1 in this specification.

### 10.8.2 original-requesting-user-name (name(MAX)) operation and Job Description attribute

The operation attribute containing the user name of the original user, i.e., corresponds to the “requesting-user-name” operation attribute (see [RFC2911] section 3.2.1.1) that the original client supplied to the first Printer object. The Printer copies the “original-requesting-user-name” operation attribute to the corresponding Job Description attribute.

### 10.8.3 requesting-user-name (name(MAX)) operation attribute - additional semantics

The IPP/1.1 “requesting-user-name” operation attribute (see [RFC2911] section 3.2.1.1) 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.

### 10.8.4 job-originating-user-name (name(MAX)) Job Description attribute - additional semantics

The “job-originating-user-name” Job Description attribute (see [RFC2911] section 4.3.6) remains as the authenticated original user, not the parent Printer’s authenticated host, and is forwarded by each client without changing the value.

## 11 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,<br>Pause-Printer-After-Current-Job |
| Hold-New-Jobs | Release-Held-New-Jobs |
| Release-Held-New-Jobs | Hold-New-Jobs |
| Activate-Printer,<br>Disable-Printer,<br>Pause-Printer-After-Current-Job | Deactivate-Printer |
| Deactivate-Printer,<br>Enable-Printer,<br>Resume-Printer | Activate-Printer |
| Restart-Printer | none |
| Shutdown-Printer | none |
| Startup-Printer | none |
| Reprocess-Job | none |
| Cancel-Current-Job | none |
| Resume-Job | Suspend-Current-Job |
| Suspend-Current-Job | Resume-Job |
| Promote-Job | none |
| Schedule-Job-After | Promote-Job |

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

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

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

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

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

# 12 Normative References

[RFC2910]
Herriot, R., Butler, S., Moore, P., Tuner, R., “Internet Printing Protocol/1.1: Encoding and Transport”, RFC 2910, September 2000.

[RFC2911]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, “Internet Printing Protocol/1.0: Model and Semantics”, RFC 2911, September 2000.

[RFC3380]
Hastings, T., Herriot, R., Kugler, C., and H. Lewis, "Internet Printing Protocol (IPP): Job and Printer Set Operations", RFC 3380, September 2002.

# 13 Informative References

[ipp-ntfy]
Herriot, R., and T. Hastings, "Internet Printing Protocol/1.1: Event Notifications and Subscriptions", <draft-ietf-ipp-not-spec-12.txt>, June 21, 2004.

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

[RFC3196]
Hastings, T., Manros, C., Zehler, P., Kugler, C., and H. Holst, “Internet Printing Protocol/1.1: Implementer’s Guide”, RFC 3196, November 2001.

[RFC3239]
Kugler, C., Lewis, H., and T. Hastings, "Internet Printing Protocol (IPP): Requirements for Job, Printer, and Device Administrative Operations", RFC 3239, February 2002.

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

# 14 IANA Considerations

This section contains the registration information for IANA to add to the IPP Registry according to the procedures defined in RFC 2911 [RFC2911] section 6 to cover the definitions in this document. The resulting registrations will be published as additions to the http://www.iana.org/assignments/ipp-registrations file.

*Note to RFC Editors: Replace [RFCnnnn] below with the RFC number for this document, so that it accurately reflects the content of the information for the IANA Registry.*

## 14.1 Attribute Registrations

The following table lists all the attributes defined in this document. These are to be registered according to the procedures in RFC 2911 [RFC2911] section 6.2.

| Name | Reference | Section |
|---|---|---|
| Job Description attributes: | | |
| original-requesting-user-name (name(MAX)) | [RFCnnnn] | 10.8.2 |
| Printer Description attributes: | | |
| subordinate-printers-supported (1setOf uri) | [RFCnnnn] | 7.1 |
| parent-printers-supported (1setOf uri) | [RFCnnnn] | 7.2 |
| Operation attributes: | | |
| original-requesting-user-name (name(MAX)) | [RFCnnnn] | 10.8.2 |

## 14.2 Attribute Value Registrations

This section lists the additional values that are defined in this document for existing attributes.

| Attribute<br>Value | Reference | Section |
|---|---|---|
| job-state-reasons (1setOf type2 keyword) | | |
| job-suspended | [RFCnnnn] | 9.1 |
| printer-state-reasons (1setOf type2 keyword) | | |
| hold-new-jobs | [RFCnnnn] | 8.1 |
| deactivated | [RFCnnnn] | 8.2 |

## 14.3 Additional Enum Attribute Value Registrations

The following table lists all the new enum attribute values defined in this document. These are to be registered according to the procedures in RFC 2911 [RFC2911] section 6.1.

| Attribute (attribute syntax) Value | Name | Reference | Section |
|---|---|---|---|
| operations-supported (1setOf type2 enum) | | [RFC2911] | 4.4.1 |
| 0x0022 | Enable-Printer | [RFCnnnn] | 3 |
| 0x0023 | Disable-Printer | [RFCnnnn] | 3 |
| 0x0024 | Pause-Printer-After-Current-Job | [RFCnnnn] | 3 |
| 0x0025 | Hold-New-Jobs | [RFCnnnn] | 3 |
| 0x0026 | Release-Held-New-Jobs | [RFCnnnn] | 3 |
| 0x0027 | Deactivate-Printer | [RFCnnnn] | 3 |
| 0x0028 | Activate-Printer | [RFCnnnn] | 3 |
| 0x0029 | Restart-Printer | [RFCnnnn] | 3 |
| 0x002A | Shutdown-Printer | [RFCnnnn] | 3 |
| 0x002B | Startup-Printer | [RFCnnnn] | 3 |
| 0x002C | Reprocess-Job | [RFCnnnn] | 4 |
| 0x002D | Cancel-Current-Job | [RFCnnnn] | 4 |
| 0x002E | Suspend-Current-Job | [RFCnnnn] | 4 |
| 0x002F | Resume-Job | [RFCnnnn] | 4 |
| 0x0030 | Promote-Job | [RFCnnnn] | 4 |
| 0x0031 | Schedule-Job-After | [RFCnnnn] | 4 |

## 14.4 Operation Registrations

The following table lists all of the operations defined in this document. These are to be registered according to the procedures in RFC 2911 [RFC2911] section 6.4.

| Name | Reference | Section |
|---|---|---|
| Activate-Printer | [RFCnnnn] | 3.4.2 |
| Cancel-Current-Job | [RFCnnnn] | 4.2 |
| Deactivate-Printer | [RFCnnnn] | 3.4.1 |
| Disable-Printer | [RFCnnnn] | 3.1.1 |
| Enable-Printer | [RFCnnnn] | 3.1.2 |
| Hold-New-Jobs | [RFCnnnn] | 3.3.1 |
| Pause-Printer-After-Current-Job | [RFCnnnn] | 3.2.1 |
| Promote-Job | [RFCnnnn] | 4.4.1 |
| Release-Held-New-Jobs | [RFCnnnn] | 3.3.2 |
| Reprocess-Job | [RFCnnnn] | 4.1 |
| Restart-Printer | [RFCnnnn] | 3.5.1 |
| Resume-Job | [RFCnnnn] | 4.3.2 |
| Schedule-Job-After | [RFCnnnn] | 4.4.2 |
| Shutdown-Printer | [RFCnnnn] | 3.5.2 |
| Startup-Printer | [RFCnnnn] | 3.5.3 |
| Suspend-Current-Job | [RFCnnnn] | 4.3.1 |

### 14.5 Status code Registrations

The following table lists the status code defined in this document. This is to be registered according to the procedures in RFC 2911 [RFC2911] section 6.6.

```
Value   Name                                    Reference  Section
------  ------------------------                ---------  -------
0x0000:0x00FF - "successful"
none at this time

0x0100:0x01FF - "informational"
none at this time

0x0300:0x03FF - "redirection"    -    -- See RFC 2911 Errata
none at this time

0x0400:0x04FF - "client-error"
none at this time

0x0500:0x05FF - "server-error"
0x050A  server-error-printer-is-deactivated   [RFCnnnn]  5.1
```

## 15 Internationalization Considerations

This document has the same localization considerations as the [RFC2911].

## 16 Security Considerations

The IPP Model and Semantics document [RFC2911] 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.

Printer operations defined in this specification (see section 3) and Pause-Printer, Resume-Printer, and Purge-Job (defined in [RFC2911]) are intended for use by an operator and/or administrator. Job operations defined in this specification (see section 4) and Cancel-Job, Hold-Job, Release-Job defined in [RFC2911]) are intended for use by the job owner or may be an operator or administrator of the Printer object. These operator and administrative operations affect the service of all users. In appropriate use of an administrative operation by an un-authenticated end user could affect the quality of service for all users. Therefore, for both inter-net and intra-net, conformance to this specification REQUIRES that initial configuration of IPP Printer implementations MUST require successful certificate-based TLS [RFC2246] client authentication and successful operator and administrator authorization (see [RFC2911] sections 5.2.7 and 8 and [RFC2910]) for any administrative operations defined in this document. [RFC2910] REQUIRES the IPP Printer to support the minimum cypher

suite required for TLS/1.0. The means for authorizing an operator or administrator of the Printer object are outside the scope of this specification, [RFC2911], and [RFC2910].

The use of TLS and Client Authentication solves the Denial of Service, Man in the Middle, and Masquerading security threats.

# 17 Author's Addresses

Carl Kugler
P.O. Box 1900
IBM
Boulder CO 80301-9191

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

Tom Hastings, editor
Xerox Corporation
701 Aviation Blvd. ESAE 231
El Segundo, CA 90245

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

Harry Lewis
P.O. Box 1900
IBM
Boulder CO 80301-9191

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

# 18 IPR Notice

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org."

# 19 Summary of Base IPP Documents

The base 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 [RFC2911]
Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
Internet Printing Protocol/1.1: Implementer's Guide [RFC3196]
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: Model and Semantics" document describes a simplified model with abstract objects, their attributes, and their operations that are independent of encoding and transport. It introduces a Printer and a Job object. The Job object optionally supports multiple documents per Job. It also addresses security, internationalization, and directory issues.

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

# 20 Full Copyright Statement



**Acknowledgement**

Funding for the RFC Editor function is currently provided by the Internet Society.