PWG DRAFT~~ to become an INTERNET-DRAFT~~
draft-ietf-ipp-ops-set1-00-980909.txt

R. Bergman
Data ~~P~~products Corp.
T. Hastings
Xerox Corporation
R. Herriot
Sun Microsystems
P. Moore
Microsoft

~~July 27~~September 9, 1998

# Internet Printing Protocol/1.0: Additional Optional Operations - Set 1

## Status of this Memo

This document is a Printer Working Group (PWG) DRAFT~~an Internet-Draft~~. The previous version (980727) was posted to the IPP mailing list for comment on 7/29/98 and reviewed at the August, 19-20, 1998 PWG meeting in Toronto. This version contains the changes agreed to at that meeting, and the IPP telecons on 9/2/98 and 9/9/98 and need to be reviewed by the participants on the mailing list. ~~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".~~

~~To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).~~

## Abstract

This document specifies ~~seven~~ six OPTIONAL operations for use with the Internet Printing Protocol/1.0 (IPP) [ipp-mod, ipp-pro]. The defined Set 1 operations are 3 Job object operations that end-users may perform on their jobs and operators/administrators may perform on any job:

- Hold-Job
- Release-Job
- Restart-Job
- ~~Reprocess-Job~~

and 3 Printer object operations that operators/administrators may perform on a Printer object:

- Pause-Printer
- Resume-Printer
- Purge-Jobs

## Table of Contents

# 1 Summary of Set 1 and Operation-Id Assignments

The Set 1 operations are summarized in the following table:

| Operation Name | Operation-Id | Brief description |
|---|---|---|
| Hold-Job | 0x000C | Holds a pending job so that it cannot be scheduled for processing |
| Release-Job | 0x000D | Allows a previously held job to be scheduled for processing |
| Restart-Job | 0x000E | Restarts a ~~completed~~ job as the same job on the same Printer object |
| ~~Reprocess-Job~~ | 0x000F | Reserved for the future.~~Reprocesses a completed job as a new copy of the job on the same Printer object~~ |
| Pause-Printer | 0x0010 | Stops the device(s) as soon as possible from processing jobs |
| Resume-Printer | 0x0011 | Resumes the device(s) processing jobs |
| Purge-Jobs | 0x0012 | Removes all jobs from the Printer regardless of job state |

All of the attributes in Set 1 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.

# 2 Job Operations

The job operations in Set 1 are for use by end users on their jobs and by operators and administrators on any jobs. The operation attributes in requests and responses for the job operations are the same as the standard Cancel-Job operation (see [model] 3.3.3). Additional operation attributes are specified that the client MAY supply in a request. In addition, the IPP object MUST return the "job-state" [ipp-mod 4.3.7] attribute and, if supported, the "job-state-reasons" [ipp-mod 4.3.8] attribute in the response as Group 3 Job Attributes in order to indicate the effect of the operation on the job object.

~~Note: In order to keep the operations in Operation Set 1 simple, they are rejected when the job is in the 'processing' or 'processing-stopped' states. If operations are needed to affect jobs while in these states, they will be added as additional operations, rather than overloading these operations. Then it is clear to clients by querying the Printer object's "operations-supported" [ipp-mod 4.4.13] what the behavior is.~~

## 2.1 Hold-Job

This operation allows a client to hold a pending job in the queue so that it is not eligible for scheduling. If the Hold-Job operation is supported, then the Release-Job operation MUST be supported, and vice-versa.

### 2.1.1 "job-hold-until" (type3 keyword | name(MAX)) operation attribute

The client OPTIONALLY supplies this attribute. The IPP object MUST support this operation attribute in a Hold-Job request, if it supports the "job-hold-until" Job template attribute in create operations. See [ipp-mod] section 4.2.2. Otherwise, the IPP object NEED NOT support the "job-hold-until" operation attribute in a Hold-Job request. If supplied and supported, the IPP object copies the attribute to the Job object, replacing the job’s previous "job-hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied named time period.

As with all operations, if the client supplies the "job-hold-until" (or any OPTIONAL) Operation attribute that is unknown or unsupported or the value is unsupported, the IPP object MUST accept and perform the operation, ignoring the unknown or unsupported operation attribute and returning the ignored or unsupported attributes and/or values in Group 2 Unsupported Attributes (see [ipp-mod] sections 3.3.3.2 and 16.3.6).

If the client supplies the ’no-hold’ value [ipp-mod 4.2.2] (meaning don’t hold the job) and the IPP object supports the "job-hold-until" operation attribute, the IPP object MUST reject the operation and return the ’client-error-bad-syntax’ error status code.

The following new keyword value is defined for use with the "job-hold-until" Job Template attribute in job create operations and the "job-hold-until" operation attribute in Hold-Job and Restart-Job operations:

> ’indefinite’: - the job is held indefinitely, until a client performs a Release-Job or Restart-Job operation

If the client does not supply a "job-hold-until" operation attribute in the request~~Hold-Job operation~~, the IPP object MUST populate the job object with a "job-hold-until" attribute with the ’indefinite’ value (if IPP object supports the "job-hold-until" attribute) and hold the job indefinitely, until a client performs a Release-Job or Restart-Job operation.

The IPP object SHOULD support the "job-hold-until" Job Template attribute for use in job create operations with at least the ’indefinite’ value, if it supports the Hold-Job operation. Otherwise, a client cannot create a job and hold it immediately (without picking some supported time period in the future).

The IPP object MUST accept or reject the request based on the job's current state, transition the job to the indicated new state, and return the indicated new "job-state" attribute and status code as follows:

| Current "job-state" | New "job-state" | IPP object's response status code and action: |
|---|---|---|
| 'pending' | 'pending-held' | 'successful-ok' See Note 1 |
| 'pending-held' | 'pending-held' | 'successful-ok' See Note 1 |
| 'processing' | 'processing' | 'client-error-not-possible' |
| 'processing-stopped' | 'processing-stopped' | 'client-error-not-possible' |
| 'completed' | 'completed' | 'client-error-not-possible' |
| 'canceled' | 'canceled' | 'client-error-not-possible' |
| 'aborted' | 'aborted' | 'client-error-not-possible' |

Note 1: If the OPTIONAL "job-state-reasons" attribute is supported and if the implementation supports multiple reasons for a job to be in the 'pending-held' state, the IPP object MUST add the 'job-hold-until-specified' value to the job's "job-state-reasons" attribute.

Note: In order to keep the Hold-Job operation simple, such a request is rejected when the job is in the 'processing' or 'processing-stopped' states. If an operation is needed to hold jobs while in these states, it will be added as an additional operation, rather than overloading the Hold-Job operation. Then it is clear to clients by querying the Printer object's "operations-supported" [ipp-mod 4.4.13] and the Job object's "job-state" [ipp-mod 4.3.7] attributes which operations are possible.

*Access Rights:* The requesting user must either be the submitter of the job or an operator or administrator of the P~~p~~rinter object (see [ipp-mod] Section 1). Otherwise, the IPP object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

### 2.2 Release-Job

This operation allows a client to release a previously held job so that it is again eligible for scheduling. This operation removes the "job-hold-until" job attribute, if present, from the job object that had been supplied in the create or most recent Hold-Job or Restart-Job operation and remove its effect on the job.

If the Hold-Job operation is supported, then the Release-Job operation MUST be supported, and vice-versa.

If the OPTIONAL "job-state-reasons" attribute is supported, the IPP object MUST remove the 'job-hold-until-specified' value from the job's "job-state-reasons" attribute, if present.

The IPP object MUST accept or reject the request based on the job's current state, transition the job to the indicated new state, and return the indicated new "job-state" attribute and status code as follows:

| Current "job-state" | New "job-state" | IPP object’s response status code and action: |
|---|---|---|
| ’pending’ | ’pending’ | ’successful-ok’ No effect on the job. |
| ’pending-held’ | ’pending-held’ | ’successful-ok’ See Note 1 |
| ’pending-held’ | ’pending’ | ’successful-ok’ |
| ’processing’ | ’processing’ | ’successful-ok’ No effect on the job. |
| ’processing-stopped’ | ’processing-stopped’ | ’successful-ok’ No effect on the job. |
| ’completed’ | ’completed’ | ’client-error-not-possible’ |
| ’canceled’ | ’canceled’ | ’client-error-not-possible’ |
| ’aborted’ | ’aborted’ | ’client-error-not-possible’ |

Note 1: If there are other reasons to keep the job in the ’pending-held’ state, such as ’resources-are-not-ready’, the job remains in the ’pending-held’ state. Thus the ’pending-held’ state is not just for jobs that have the ’job-hold-until’ applied to them, but are for any reason to keep the job from being a candidate for scheduling and processing, such as ’resources-are-not-ready’. See the "job-hold-until" attribute ([ipp-mod] Section 4.2.2).

*Access Rights:* The requesting user must either be the submitter of the job or an operator or administrator of the P~~p~~rinter object. Otherwise, the IPP object MUST reject the operation and return: ’client-error-forbidden’, ’client-error-not-authenticated’, or ’client-error-not-authorized’ as appropriate.

## 2.3 Restart-Job

This operation allows a client to restart a job that is retained in the queue after processing has completed. As an implementation option, a job in the ’processing’ and/or ’processing-stopped’ states MAY be restarted.

The job is moved to the ’pending’ job state and restarts at the beginning on the same IPP Printer object with the same attribute values. The Job Description attributes that accumulate job progress, such as "job-impressions-completed", "job-media-sheets-completed", and "job-k-octets-processed", MUST be reset to 0 so that they give an accurate record of the job from its restart point. The job object MUST continue to use the same "job-uri" and "job-id" attribute values.

### 2.3.1 "job-hold-until" (type3 keyword | name(MAX)) operation attribute

The client OPTIONALLY supplies this attribute. The IPP object MUST support this operation attribute in a Restart-Job request, if it supports the "job-hold-until" Job template attribute in create operations. See [ipp-mod] section 4.2.2. Otherwise, the IPP object NEED NOT support the "job-hold-until" operation attribute in a Restart-Job request. If supplied and supported, the IPP object copies the attribute to the Job object, replacing the job’s previous "job-hold-until" attribute, if present, and makes the job a candidate for scheduling during the supplied named time period.

See Section 2.1.1 for the common semantics of the "job-hold-until" operation attribute for the Hold-Job operation and Restart-Job operation.

Note: In the future an OPTIONAL Modify-Job operation may be specified that allows the client to modify other attributes before releasing the restarted job.

The IPP object MUST accept or reject the request based on the job’s current state, transition the job (or new job, depending on implementation) to the indicated new state, and return the indicated new "job-state" attribute and status code as follows:

| Current "job-state" | New "job-state" | IPP object’s response status code and action: |
|---|---|---|
| ’pending’ | ’pending’ | ’client-error-not-possible’. |
| ’pending-held’ | ’pending-held’ | ’client-error-not-possible’. |
| ’processing’ | ’pending’ | OPTION 1: ’successful-ok’ - job is started over. See Note 1. |
| ’processing’ | ’processing’ | OPTION 2: ’client-error-not-possible’. |
| ’processing-stopped’ | ’pending’ | OPTION 1: ’successful-ok’ - job is started over. See Note 1 |
| ’processing-stopped’ | ’processing-stopped’ | OPTION 2: ’client-error-not-possible’. |
| ’completed’ | ’pending’ | ’successful-ok’ - job is started over. |
| ’canceled’ | ’pending’ | ’successful-ok’ - job is started over. |
| ’aborted’ | ’pending’ | ’successful-ok’ - job is started over. |

Note 1: For OPTION 1, the IPP object SHOULD indicate to clients that a Restart-Job operation is possible while the job is in the ’processing’ and/or ’processing-stopped’ states, by populating the job’s "job-state-reasons" attribute with the ’job-restartable’ value.

~~Note: Resetting the job progress attributes, allows job monitoring applications to function unchanged for a job that has been restarted. However, there is a problem for accounting applications that "pull" the job accounting data from the IPP object after the job completes using the Get-Job-Attributes or Get-Jobs operations (or SNMP MIBs). Since the "job-id" and "job-uri" for the restarted job are the same as the original job and the accounting attributes are reset, the accounting program may not be able to detect that the job was restarted and is using additional resources. It is recommended that the Reprocess-Job operation (see section 2.4) be used when accurate accounting data is desired to be made available to accounting programs that pull the data from the IPP Printer after the job completes, since a new job with a new "job-id" and "job-uri" is created while the old job remains for the accounting program to query accounting attributes. On the other hand, if an IPP object "pushes" the accounting data to the accounting application when the job completes, say, using event notification [ipp-not], then support of the Restart-Job operation is not in conflict with such "pull" accounting.~~

*Access Rights:* The requesting user must either be the submitter of the job or an operator or administrator of the ~~P~~printer object. Otherwise, the IPP object MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

# 3 The Job History concept and additional "job-state-reasons"

This section explains the so-called Printer object's "job-history" that contains the recently completed, canceled, and aborted jobs. This section also specifies the 'job-restartable' value of the "job-state-reasons" Job Description attribute for use with the Restart-Job operations.

## 3.1 The "job history" concept

When a job is completed, canceled, or aborted, the IPP Printer object MAY retain the job with its document data in a restartable condition using the Restart-Job operation for an implementation-defined time period which may be zero seconds. If the IPP object supports the "job-state-reasons" attribute and the Restart-Job operation, then it SHOULD indicate that such jobs are restartable by adding the 'job-restartable' value to the job's "job-state-reasons" attribute (see Section 3.2) during that implementation-defined time period.

After the implementation-defined restartable time period expires, the Printer object deletes the document data for the job and the job becomes part of the "job history". The Print object MAY also delete any number of the job attributes. Since the job is no longer restartable, the Printer object MUST remove the 'job-restartable' value from the job's "job-state-reasons" attribute, if supported. Clients are able to query jobs in the Printer object's "job history" using Get-Job-Attributes and Get-Jobs operations.

Subsequently, the IPP Printer removes jobs from its "job history" in an implementation-defined manner, such as after a fixed time period (which MAY be zero seconds) or when the number of jobs exceeds a fixed number. Thereupon, the job can no longer be queried using the Get-Job-Attributes and Get-Jobs operations and the IPP object returns the 'client-error-not-found' or 'client-error-gone' as appropriate.

## 3.2 Add a new 'job-restartable' value to the "job-state-reasons" attribute

The following new keyword value is specified for use with the "job-state-reasons" Job Description attribute and the Restart-Job operation (see Section 2.3):

'job-restartable' - This job is currently able to be restarted using the Restart-Job operation. With which job states this values is used depends on implementation, i.e., OPTION 1 vs. OPTION 2 in the job state transition table in Section 2.3. Whenever the IPP object will reject a Restart-Job operation for the job with the 'client-error-not-possible' error status code because the job is not restartable, the IPP object MUST remove this value from the job's "job-state-reasons" attribute.

For example, after a job is completed (job state is ’completed’, ’aborted’, or ’canceled), the implementation MAY retain the job in a restartable condition for an implementation-defined time period. When that time elapses, an implementation MAY delete the document data, but MAY retain some or all of the job attributes as a "job history" for an additional implementation-defined time period. During this second time period, the implementation removes the ’job-restartable’ value from the job’s "job-state-reasons" attribute, since the job can no longer be restarted.

### ~~2.4Reprocess-Job~~

~~This operation allows a client to reprocess a copy of the job that is retained in the queue after processing is completed. A copy of the job restarts at the beginning on the same IPP Printer object with possibly different Job Template attributes supplied by the client in the request. Thus the Reprocess-Job operation is another create job operation and all of the semantics that [ipp-mod] specifies for "create job operations" also apply to the Reprocess-Job operation.~~

~~The client MAY supply any Job Template attributes as in a create job operation whether they were originally supplied in the job create operation or not. The Printer object performs a validation as in a create operation of the job that would be made up of any supplied attributes replacing the corresponding job’s attributes in combination with any of the job’s remaining Job Template attributes. If the Printer object supports the new combination of Job Template attributes, the Printer object accepts the Reprocess-Job operation, creates a new job, assigns new "job-id" and "job-uri" values, and makes a copy of the job attributes with their new values.~~

~~The IPP object initializes the Job Description attributes of the new job as in a create job operation, so that attributes such as "media-sheets-completed", and "job-k-octets-processed" start at 0 and the new job enters the ’pending’ or ’pending-held’ state, as after a job create operation. The returned groups are the same as for the Print-Job operation including the "job-id" and "job-uri" attributes with the new values assigned by the Printer object, whether the job has one or multiple documents.~~

~~The Printer object leaves the old ’completed’, ’canceled’, or ’aborted’ job as is and does not change any of its attributes. Therefore, the Job Description attributes are preserved for job monitoring and accounting purposes for the specified (old) job. Whether the document data is copied or shared between the old and the new job, depends on implementation, and cannot be detected by the client. Either the old job or the new job may be the target of subsequent Reprocess-Job operations.~~

~~The IPP object MUST accept or reject the request based on the job’s current state, transition the job to the indicated new state, and return the indicated new job’s new "job-state" attribute and status code as follows:~~

| ~~Old job’s current "job-state"~~ | ~~New job’s new "job-state"~~ | ~~IPP object’s response status code and action:~~ |
|---|---|---|

| | | |
|---|---|---|
| ~~’pending’~~ | ~~n/a~~ | ~~’client-error-not-possible’.~~ |
| ~~’pending-held’~~ | ~~n/a~~ | ~~’client-error-not-possible’.~~ |
| ~~’processing’~~ | ~~n/a~~ | ~~’client-error-not-possible’.~~ |
| ~~’processing-stopped’~~ | ~~n/a~~ | ~~’client-error-not-possible’.~~ |
| ~~’completed’~~ | ~~’pending’ or ’pending-held’~~ | ~~’successful-ok’~~ |
| ~~’canceled’~~ | ~~’pending’ or ’pending-held’~~ | ~~’successful-ok’~~ |
| ~~’aborted’~~ | ~~’pending’ or ’pending-held’~~ | ~~’successful-ok’~~ |

~~*Access Rights:* The requesting user must either be the submitter of the job or an administrator of the printer. Otherwise, the IPP object MUST reject the operation and return: ’client-error-forbidden’, ’client-error-not-authenticated’, or ’client-error-not-authorized’ as appropriate.~~

# ~~3~~4 Printer operations

The printer operations in Set 1 are for use by operators and administrators of each Printer object. The following figure is copied from [ipp-mod] with the addition of the following arrow:

```
     any****>
```

indicating an additional source of job submission using IPP or any other job submission protocol that passes to the device but does NOT pass through the IPP Printer object in question.

```
Legend:

##### indicates a Printer object which is
      either embedded in an output device or is
      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

embedded printer:
                                          output device
                                any****>+---------------+
 O   +--------+                         |  ###########  |
/|\  | client |------------IPP------------># Printer #  |
/ \  +--------+                         |  # Object  #  |
                                        |  ###########  |
                                        +---------------+

hosted printer:
                                any****>+---------------+
 O   +--------+        ###########      |               |
/|\  | client |--IPP--># Printer #-any->| output device |
/ \  +--------+        # Object  #      |               |
                       ###########      +---------------+


                                 any****>+---------------+
fan out:                                 |               |
                                     +-->| output device |
                                 any/    |               |
 O   +--------+      ###########   /     +---------------+
/|\  | client |-IPP-># Printer #--*
/ \  +--------+      # Object  #   \     +---------------+
                     ########### any\    |               |
                                     +-->| output device |
                                         |               |
                                         +---------------+
```

The operation attributes for the Printer operation requests are as follows:-

Group 1: Operation Attributes

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

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

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

The operation attributes for the Printer operation responses are as follows:

Group 1: Operation Attributes

Status Message:
In addition to the REQUIRED status code returned in every response, the response OPTIONALLY includes a "status-message" (text) operation attribute as described in section 3.1.6 of [ipp-mod].

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

Group 2: Unsupported Attributes

This is a set of Operation attributes supplied by the client (in the request) that are not supported by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16).

Group 3: Printer Object Attributes

Printer state information~~"printer-state" and "printer-state-reasons~~:
The Printer object MUST return the "printer-state" [ipp-mod 4.4.10] and, if supported, the "printer-state-reasons" [ipp-mod 4.4.11] attributes in order to indicate the effect of the operation on the Printer object.

### ~~1.1~~4.1 Pause-Printer

This operation allows a client to stop the Printer object from scheduling jobs on all its devices. Depending on implementation, the Pause-Printer operation MAY also ~~and to~~ stop the Printer from processing the current job or jobs~~, if supported~~. Any job that is currently being printed is either stopped as soon as the implementation permits or is completed, depending on implementation. The Printer object MUST still accept create operations to create new jobs, but MUST prevent any jobs from entering the 'processing' state.

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

The IPP Printer stops the current job(s) on its device(s) that were in the 'processing' or 'processing-stopped' states as soon as the implementation permits. If the implementation supports the "printer-state-reasons" attribute and the devices will take appreciable time to stop, the IPP Printer adds the 'moving-to-paused' value to the Printer object's "printer-state-reasons" attribute. When the device(s) have all stopped, the IPP Printer transitions the Printer object to the 'stopped' state, removes the 'moving-to-paused' value, if present, and adds the 'paused' value to the Printer object's "printer-state-reasons" attribute.

When the current job(s) complete that were in the 'processing' state, the IPP Printer transitions them to the 'completed' state. When the current job(s) stop in mid processing that were in the 'processing' state, the IPP Printer transitions them to the 'processing-stopped' state and, if the "job-state-reasons" attribute is supported, adds the 'printer-stopped' value to the job's "job-state-reasons" attribute.

Note: 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 job's "job-state-reasons" attributes and only need return the 'printer-stopped' value when those jobs are queried (so-called "lazy evaluation").

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

The IPP Printer MUST accept the request in any state, transition the Printer to the indicated new "printer-state" before returning, and return the indicated "printer-state", "printer-state-reasons", and status code as follows:

| Current "printer-state" | New "printer-state" returned | "printer-state-reasons" | IPP Printer's response status code and action: |
|---|---|---|---|
| 'idle' | 'stopped' | 'paused' | 'successful-ok' |
| 'processing' | 'processing' | 'moving-to- | OPTION 1: 'successful-ok'; |

| | | | |
|---|---|---|---|
| | | paused’ | 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 |
| ’processing’ | ’stopped’ | ’paused’ | OPTION 2: ’successful-ok’;<br>all output stopped immediately |
| ’stopped’ | ’stopped’ | ’paused’ | ’successful-ok’ |

*Access Rights:* The requesting user must be an operator or administrator of the P~~p~~rinter object. Otherwise, the IPP Printer MUST reject the operation and return: ’client-error-forbidden’, ’client-error-not-authenticated’, or ’client-error-not-authorized’ as appropriate.

#### ~~1.1.1~~4.1.1 Add a new ’moving-to-paused’ value to the "printer-state-reasons" attribute

The following new keyword value is specified for use with the "printer-state-reasons" Printer Description attribute:

’moving-to-paused’: ~~Someone~~ The Printer object’s operator or administrator has paused the Printer object using the Pause-Printer operation or other means, but it has not yet stopped producing output. When all the devices stop producing output, the Printer object MUST replace this value with the ’paused’ value.

### ~~1.2~~4.2 Resume-Printer

This operation allows a client to resume the Printer object scheduling jobs on all its devices. If the Printer object supports the "printer-state-reasons" attribute, it MUST remove the ’paused’ and ’moving-to-paused’ values from the Printer object’s "printer-state-reasons" attribute, if present. If there are no other reasons to keep a device paused (such as media-jam), the IPP Printer transitions itself to the ’processing’ or ’idle’ states, depending on whether there are jobs to be processed or not, respectively, and the device(s) resume processing jobs.

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

The IPP Printer removes the ’printer-stopped’ value from any job’s "job-state-reasons" attributes contained in that Printer.

The IPP Printer MUST accept the request in any state, transition the Printer object to the indicated new state, and return the indicated "printer-state" and status code as follows:

| Current "printer-state" | New "printer-state" returned | IPP Printer’s response status code and action: |
|---|---|---|
| ’idle’ | ’idle’ | ’successful-ok’ |
| ’processing’ | ’processing’ | ’successful-ok’ |
| ’stopped’ | ’processing’ | ’successful-ok’; |

| | | |
|---|---|---|
| | | when there are jobs to be processed |
| 'stopped' | 'idle' | 'successful-ok';<br>when there are no jobs to be processed. |

*Access Rights:* The requesting user must be an operator or administrator of the P~~p~~rinter object. Otherwise, the IPP Printer MUST reject the operation and return: 'client-error-forbidden', 'client-error-not-authenticated', or 'client-error-not-authorized' as appropriate.

### ~~1.3~~4.3 Purge-Jobs

This operation allows a client to remove all jobs from an IPP Printer object, regardless of their job states, including jobs in the Printer object's "job-history" (see Section 3.1). After a Purge-Jobs operation has been performed, a Printer object MUST return no jobs in subsequent Get-Job-Attributes and Get-Jobs responses (until new jobs are submitted).

Whether the Purge-Jobs (and Get-Jobs) operation affects jobs that were submitted to the device from other sources than the IPP Printer object (see the any***> arrow in the figure in Section 4) in the same way that the Purge-Jobs operation affects jobs that were submitted to the IPP Printer object using IPP, depends on implementation, i.e., on whether the IPP protocol is being used as a universal management protocol or just to manage IPP jobs, respectively.

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

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

*Access Rights:* The requesting user must be an operator or administrator of the P~~p~~rinter object. Otherwise, the IPP object MUST reject the operation and return: client-error-forbidden, client-error-not-authenticated, and client-error-not-authorized as appropriate.

## ~~4~~5 References

[ipp-mod]

Isaacson, S., deBry, R., Hastings, T., Herriot, R., Powell, P., “Internet Printing Protocol/1.0: Model and Semantics” draft-ietf-ipp-mod-10.txt, June, 1998.

~~[ipp-not]~~

~~Isaacson, S., Martin, J., deBry, R., Hastings, T., “IPP Event Notifications (Very Short)” <ipp-notifications-very-short-980701.doc>, July 1, 1998.~~

[ipp-pro]

Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and Transport", draft-ietf-ipp-pro-06.txt, June, 1998.

~~[ISO-10175]~~

~~ISO/IEC 10175 Document Printing Application (DPA), June 1996.~~