Internet Printing Protocol WG
INTERNET-DRAFT
<draft-ietf-ipp-ops-admin-req-01.txt>
[Target Category: Informational]
Expires: January 17, 2002

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

July 17, 2001

# Internet Printing Protocol (IPP): Requirements for Job, Printer, and Device Administrative Operations



## Status of this Memo



## Abstract

This document specifies the requirements and use cases for some optional administrative operations for use with the Internet Printing Protocol/1.0 (IPP) [RFC2565, RFC2566] and IPP/1.1 [RFC2911, RFC2910]. Some of these administrative operations operate on the IPP Job and Printer objects. The remaining operations operate on a new Device object that more closely models a single output device (see [RFC2911]).

# Table of Contents



# List of Tables

# 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 (for a description of the base IPP documents, see section 9). .This document defines the requirements and use cases for additional optional end user, operator, and administrator operations used to control Job objects, Printer objects (see [RFC2911]) and a new Device object. The new Device object more closely models a single output device and has no notion of a job, while the Printer object models a print service which understands jobs and may represent one or more output devices.

The scope of IPP is characterized in RFC 2526 [RFC2526] “Design Goals for an Internet Printing Protocol”. It is not the intent of this document to revise or clarify this scope or conjecture as to the degree of industry adoption or trends related to IPP within printing systems. It is the intent of this document to extend the original set of operations - in a similar fashion to the Set1 extensions which referred to IPP/1.0 and were later incorporated into IPP/1.1.

# 2 Terminology

This section defines terminology used throughout this document and the corresponding documents that define the Administrative operations on Job, Printer, and Device objects.

This document uses terms such as “client”, “Printer”, “Job”, “attributes”, “keywords”, 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.

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

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

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

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

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

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

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

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

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

# 3 Requirements and Use Cases

The Administrative operations for Job and Printer objects will be defined in one document [ipp-admin-ops]. The Administrative operations for Device objects will be defined in a separate document. The requirements are presented here together to show the parallelism.

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

2. Support fan-out of Printer objects.

3. Support fan-out of Output Devices.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

11. Need a Printer Operation to stop the processing of IPP jobs after all of the currently accepted jobs have been processed, but any newly accepted jobs go into the 'processing-held' state.

    Usage: This allows an operator to reconfigure the Output Device in order to let jobs that are held waiting for resources, such as special media, to get a chance. Then the operator uses another operation after reconfiguring. He repeats the two operations to restore the Output Device to its normal media.

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

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

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

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

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

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

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

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

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

    Suggested name and operation: **Deactivate-Printer, Activate-Printe**r

15. Need a Device Operation that combines Disable-Device, Pause-Device-After-Current-Job, and rejects all other Device Operations, except Job and Printer queries and the companion operation

158 to resume activity. In other words, this operation makes the Output Device a read-only object in
159 a graceful manner.

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

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

163 16. Need a Printer Operation to recover from the IPP Printer software that has gotten confused (run
164 out of heap memory or gotten into a state that it doesn't seem to be able to get out of). This is a
165 condition that shouldn't happen, but does in real life. Any volatile information is saved if
166 possible before the software is re-initialized. No companion operation is needed to undo this.
167 We don't want to go back to the "confused" state :-).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Suggest name and operation: Power-On-Device

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

**Table 1 - List of Printer Operations and corresponding Device Operations**

| Printer Operation | Corresponding Device Operation equivalent |
|---|---|
| Disable-Printer | Disable-Device |
| Enable-Printer | Enable-Device |
| Pause-Printer (IPP/1.1 - [RFC2911] - one interpretation) | Pause-Device-Now |
| no | Pause-Device-After-Current-Copy |
| Pause-Printer-After-Current-Job | Pause-Device-After-Current-Job |
| Resume-Printer (IPP/1.1 - [RFC2911]) | Resume-Device |
| Hold-New-Jobs | no |
| Release-Held-New-Jobs | no |
| Deactivate-Printer | Deactivate-Device |
| Activate-Printer | Activate-Device |
| Purge-Jobs (IPP/1.1 - [RFC2911]) | Purge-Device |
| Restart-Printer | Reset-Device |
| Shutdown-Printer | Power-Off-Device |
| Startup-Printer | Power-On-Device |

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

# 4 IANA Considerations

This document does not define anything to be registered. When a document is produced that defines operations that meet the requirements in this document, those operations will be registered according to the procedures in [RFC2911] section 6.4.

# 5 Internationalization Considerations

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

# 6 Security Considerations

This document defines the requirements for operations that are intended to be used by an operator or system administrator. These operations, when defined, would affect how the Printer behaves and establish policy and/or operating behavior that ordinary users shouldn't be able to perform. Printer implementations that support such operations should authenticate users and authorized them as being an operator or a system administrator for the system. Otherwise, unprivileged users could affect the policy and behavior of IPP

Printers, thereby affecting other users. Similarly clients that supports such operations should be prepared to provide the necessary authentication information. See the security provisions in [RFC2911] for authentication, such as TLS.

# 7 Author's Addresses

Carl Kugler
IBM
Boulder CO

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

Tom Hastings
Xerox Corporation
737 Hawaii St. ESAE 231
El Segundo, CA 90245

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

Harry Lewis
IBM
Boulder CO

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

IPP Web Page: http://www.pwg.org/ipp/
IPP Mailing List: ipp@pwg.org

To subscribe to the ipp mailing list, send the following email:

1) send it to majordomo@pwg.org
2) leave the subject line blank
3) put the following two lines in the message body:

subscribe ipp
end

Implementers of this specification document are encouraged to join the IPP Mailing List in order to participate in any discussions of clarification issues and review of registration proposals for additional attributes and values. In order to reduce spam the mailing list rejects mail from non-subscribers, so you must subscribe to the mailing list in order to send a question or comment to the mailing list.

# 8 References

[ipp-iig]
Hastings, T., Manros, C., “Internet Printing Protocol/1.1: draft-ietf-ipp-implementers-guide-v11-03.txt, work in progress, July 17, 2001.

[ipp-ntfy]
Herriot, R., Hastings, T., Isaacson, S., Martin, J., deBry, R., Shepherd, M., Bergman, R., “Internet Printing Protocol/1.1: IPP Event Notifications and Subscriptions”, <draft-ietf-ipp-not-spec-07.txt>, work in progress, July 17, 2001.

[ipp-ops-set2]
Kugler, C., , Hastings, T., Lewis, H, "Internet Printing Protocol (IPP): Job and Printer Administrative Operations", <draft-ietf-ipp-ops-set2-03.txt>, work in progress, July 17, 2001.

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

[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.

# 9 Appendix A: Description 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 [ipp-iig]
Mapping between LPD and IPP Protocols [RFC2569]
Internet Printing Protocol (IPP): IPP Event Notifications and Subscriptions [ipp-ntfy]

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

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

The “Internet Printing Protocol/1.1: Model and Semantics” 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.

The "IPP Event Notifications and Subscriptions" document defines an extension to IPP/1.0 [RFC2566, RFC2565] and IPP/1.1 [RFC2911, RFC2910]. This extension allows a client to subscribe to printing related Events and defines the semantics for delivering asynchronous *Event Notifications* to the specified *Notification Recipient* via a specified *Delivery Method* (i.e., protocols) defined in (separate) Delivery Method documents.

# 10 Appendix B: Full Copyright Statement



**Acknowledgement**

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