INTERNET-DRAFT

Robert Herriot (editor)
Sun Microsystems, Inc.
~~draft-ietf-ipp-lpd-ipp-map-03.txt~~draft-ietf-ipp-lpd-ipp-map-04.txt
Tom Hastings
Xerox Corporation
Norm Jacobs
Sun Microsystems, Inc.
Jay Martin
Underscore, Inc.
~~December 19, 1997~~June 30, 1998

Mapping between LPD and IPP Protocols

Status of this Memo

Copyright Notice

Abstract

This document is one of a set of documents, which together describe all aspects of a new Internet Printing Protocol (IPP). IPP is an application level protocol that can be used for distributed printing using Internet tools and technologies. The protocol is heavily influenced by the printing model introduced in the Document Printing Application (DPA) [ISO10175] standard. Although DPA specifies both end user and administrative features, IPP version 1.0 (IPP/1.0) focuses only on end user functionality.

The full set of IPP documents includes:

- Design Goals for an Internet Printing Protocol [ipp-req] (informational)
- Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [ipp-rat] (informational)
- Internet Printing Protocol/1.0: Model and Semantics [ipp mod]
- Internet Printing Protocol/1.0: Encoding and Transport [ipp-pro]
- Mapping between LPD and IPP Protocols (this document) (informational)

The design goals document, “Design Goals for an Internet Printing Protocol”, 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. The design goals document calls out a subset of end user requirements that are satisfied in IPP/1.0. Operator and administrator requirements are out of scope for version 1.0. The rationale document, “Rationale for the Structure and Model and Protocol for the Internet Printing Protocol”, describes IPP from a high level view, defines a roadmap for the various documents that form the suite of IPP specifications, and gives background and rationale for the IETF working group’s major decisions. The document, “Internet Printing Protocol/1.0: Model and Semantics”, describes a simplified model with abstract objects, their attributes, and their operations. The model introduces a Printer and a Job. The Job supports multiple documents per Job. The model document also

addresses how security, internationalization, and directory issues are addressed. The protocol specification, “Internet Printing Protocol/1.0: Encoding and Transport”, is a formal mapping of the abstract operations and attributes defined in the model document onto HTTP/1.1. The protocol specification defines the encoding rules for a new Internet media type called “application/ipp”.

~~This Internet-Draft~~The “Mapping between LPD and IPP Protocols” gives some advice to implementors of gateways between IPP and LPD (Line Printer Daemon) implementations. It specifies the mapping between (1) the commands and operands of the "Line Printer Daemon (LPD) Protocol" specified in RFC 1179 and (2) the operations and parameters of the Internet Printing Protocol (IPP). One of the purposes of this document is to compare the functionality of the two protocols. Another purpose is to facilitate implementation of gateways between LPD and ~~IPP.~~

IPP. This document~~is an informational document that is not on the standards track. It is intended to help implementors of gateways between IPP and LPD. It~~ also provides an example, which gives additional insight into IPP~~.~~

WARNING: RFC 1179 was not on standards track. While RFC 1179 was intended to record existing practice, it fell short in some areas. However, this specification maps between (1) the actual current practice of RFC 1179 and (2) IPP. This document does not attempt to map the numerous divergent extensions to the LPD protocol that have been made by many implementers.

TABLE OF CONTENTS

Mapping between the LPD and IPP Protocols

1. Introduction

The reader of this specification is expected to be familiar with the IPP Model and Semantics specification [ipp-mod], the IPP ~~Protocol specification~~Encoding and Transport [ipp-pro], and the Line Printer Daemon (LPD) protocol specification [rfc1179] as described in RFC 1179.

RFC 1179 was written in 1990 in an attempt to document existing LPD protocol implementations. Since then, a number of undocumented extensions have been made by vendors to support functionality specific to their printing solutions. All of these extensions consist of additional control file commands. This document does not address any of these vendor extensions. Rather it addresses existing practice within the context of the features described by RFC 1179. Deviations of existing practice from RFC 1179 are so indicated.

Other LPD control file commands in RFC 1179 are obsolete. They are intended to work on "text" only formats and are inappropriate for many contemporary document formats that completely specify each page. This document does not address the support of these obsolete features.

In the area of document formats, also known as page description languages (PDL), RFC 1179 defines a fixed set with no capability for extension. Consequently, some new PDL's are not supported, and some of those that are supported are sufficiently unimportant now that they have not been registered for use with the Printer MIB[rfc1759] and IPP[ipp-mod] [ipp-pro], though they could be registered if desired. See the Printer MIB specification [rfc1759] and/or the IPP Model specification [ipp-mod] for instructions for registration of document-formats with IANA. IANA lists the registered document-formats as "printer languages".

This document addresses the protocol mapping for both directions: mapping of the LPD protocol to the IPP protocol and mapping of the IPP protocol to the LPD protocol. The former is called the "LPD-to-IPP mapper" and the latter is called the "IPP-to-LPD mapper".

This document is an informational document that is not on the standards track. It is intended to help implementors of gateways between IPP and LPD. It also provides an example, which gives additional insight into IPP.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", ~~"SHALL", "SHALL NOT", "SHOULD",~~MUSTMUST"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [abnf].

RFC 1179 uses the word "command" in two contexts: for over-the-wire operations and for command file functions. This document ~~SHALL use~~uses the word "command" for the former and the phrase "functions" for the latter. The syntax of the LPD commands is given using ABNF [abnf].

The following tokens are used in order to make the syntax more readable:

LF stands for %x0A (linefeed)
SP stands for %x20. (space)
DIGIT stands for %x30-39 ("0" to "9")

3. Mapping from LPD Commands to IPP Operations

This section describes the mapping from LPD commands to IPP operations. Each of the following sub-sections appear as sub-sections of section 5 of RFC 1179.

The following table summarizes the IPP operation that the mapper uses when it receives an LPD command. Each section below gives more detail.

LPD command	**IPP operation**
print-any-waiting-jobs	*ignore*
receive-a-printer-job	Print-Job or Create-Job/Send-Document
send queue state (short or long)	Get-Printer-Attributesand Get-Jobs
remove-jobs	Cancel-Job

3.1 Print any waiting jobs

Command syntax:

print-waiting-jobs = %x01 printer-name LF

This command causes the LPD daemon check its queue and print any waiting jobs. An IPP printer handles waiting jobs without such a nudge.

If the mapper receives this LPD command, it ~~SHALL~~MUSTMUST ignore it and send no IPP operation.

3.2 Receive a printer job

Command syntax:

receive-job = %x02 printer-name LF

The control file and data files mentioned in the following paragraphs are received via LPD sub-commands that follow this command. Their mapping to IPP commands and attributes is described later in this section.

The mapper maps the 'Receive a printer job' command to either:

- the Print-Job operation which includes a single data file or
- the Create-Job operation followed by one Send-Document operation for each data file.

If the IPP printer supports both Create-Job and Send-Document, and if a job consists of:

- a single data file, the mapper SHOULD use the Print-Job operation, but MAY use the Create-Job and Send-Document operations.
- more than one data file, the mapper ~~SHALL~~MUST use Create-Job followed by one Send-Document for each received LPD data file.

If the IPP printer does not support both Create-Job and Send-Document, and if a job consists of:

- a single data file, the mapper ~~SHALL~~MUST use the PrintJob operation.

- more than one data file, the mapper ~~SHALL~~MUST submit each received LPD data file as a separate Print-Job operation (thereby converting a single LPD job into multiple IPP jobs).

If the mapper uses Create-Job and Send-Document, it MUST send the Create-Job operation before it sends any Send-Document operations whether the LPD control file, which supplies attributes for Create-Job, arrives before or after all LPD data files.

NOTE: This specification does *not* specify how the mapper maps: the LPD Printer-name operand to the IPP "printer-uri" parameter.

The following 3 sub-sections gives further details about the mapping from LPD receive-a-printer-job sub-commands. Each of the following sub-sections appear as sub-sections of section 6 of RFC 1179.

3.2.1 Abort job

Sub-command syntax:

abort-job = %x1 LF

This sub-command of receive-a-printer-job is intended to abort any job transfer in process.

If the mapper receives this sub-command, it ~~SHALL~~MUST cancel the job that it is in the process of transmitting.

If the mapper is in the process of sending a Print-Job or Create-Job operation, it terminates the job either by closing the connection, or performing the Cancel-Job operation with the job-uri that it received from the Print-Job or Create-Job operation.

NOTE: This sub-command is implied if at any time the connection between the LPD client and server is terminated before an entire print job has been transferred via an LPD Receive-a-printer-job request.

3.2.2 Receive control file

Sub-command syntax:

```
receive-control-file = %x2 number-of-bytes SP name-of-control-file LF
number-of-bytes = 1*DIGIT
name-of-control-file = “cfA” job-number client-host-name
                       ; e.g. “cfA123woden”
job-number = 3DIGIT
client-host-name = <a host name>
```

This sub-command is roughly equivalent to the IPP Create-Job operation.

The mapper ~~SHALL~~MUST use the contents of the received LPD control file to create IPP parameter and attribute values to transmit with the Print-Job or Create-Job operation.

3.2.3 Receive data file

Sub-command syntax: %x3 number-of-bytes-in-data-file Name-of-data-file

receive-data-file = %x03 number-of-bytes SP name-of-data-file LF

number-of-bytes = 1*DIGIT
name-of-data-file = “df” letter job-number client-host-name
; e.g. “dfA123woden for the first file
letter = %x41-5A / %x61-7A ; “A” to “Z”, “a” to “z”
; first file is “A”,
; second “B”, and 52nd file is “z”
job-number = 3DIGIT
client-host-name = <a host name>

This sub-command is roughly equivalent to the IPP Send-Document operation.

The mapper ~~SHALL~~MUST use the contents of the received LPD data file as the data to transmit with the IPP Print-Job or Send-Document operation.

Although RFC-1179 alludes to a method for passing an unspecified length data file by using an octet-count of zero, no implementations support this feature.. The mapper ~~SHALL~~MUST reject a job that has a value of 0 in the number-of-bytes field.

3.3 Send queue state (short)

Command syntax:

send-queue-short = %x03 printer-name *(SP(user-name / job-number)) LF

The mapper’s response to this command includes information about the printer and its jobs. RFC 1179 specifies neither the information nor the format of its response. This document requires the mapper to follow existing practice as specified in this document.

The mapper ~~SHALL~~MUST produce a response in the following format which consists of a printer-status line optionally followed by a heading line, and a list of jobs. This format is defined by examples below. Appendix A contains the ABNF syntax.

For an printer with no jobs, the response starts in column 1 and is:

```
no entries
```

For a printer with jobs, an example of the response is:

```
~~killtree~~pinetree is ready and printing
Rank   Owner      Job          Files              Total Size
active fred       123          stuff              1204 bytes
1st    smith      124          resume, foo        34576 bytes
2nd    fred       125          more               99 bytes
3rd    mary       126          mydoc              378 bytes
4th    jones      127          statistics.ps      4567 bytes
5th    fred       128          data.txt           9 bytes
```

The column numbers of above headings and job entries are:

```
|      |          |            |                  |
01     08         19           35                 63
```

The mapper ~~SHALL~~MUST produce each field above from the following IPP attribute:

LPD field	IPP attribute	special conversion details
~~printer-status~~	~~printer-state and printer-state-reasons~~	~~For a printer-state of idle or processing, the mapper SHALL use the formats above. For stopped, the mapper SHALL use printer-state-reasons to produce an unspecified format for the error.~~
printer-status	printer-state and printer-state-reasons	For a printer-state of idle or processing, the mapper MUST use the formats above. For stopped, the mapper MUST use printer-state-reasons to produce an unspecified format for the error.
~~rank~~	~~number-of-intervening-jobs~~	~~the mapper SHALL the format above~~
rank	number-of-intervening-jobs	the mapper MUST the format above
owner	job-originating-user-name	unspecified conversion; job-originating-user-name may be the mapper's user-name
~~job~~	~~job-id~~	~~the mapper shall use the job-id~~
job	job-id	the mapper MUST use the job-id
~~files~~	~~document-name~~	~~the mapper shall create a comma separated list of the document-names and then truncate this list to the first 24 characters~~
files	document-name	the mapper MUST create a comma separated list of the document-names and then truncate this list to the first 24 characters
~~total-size~~	~~job-k-octets*copies*1024~~	~~the mapper shall multiple the value of job-k-octets by 1024 and by the value of the "copies" attribute.~~
total-size	job-k-octets*copies*1024	the mapper MUST multiple the value of job-k-octets by 1024 and by the value of the "copies" attribute.

A mapper SHOULD use the job attribute number-of-intervening-jobs rather than the job's position in a list of jobs to determine 'rank' because a Printer may omit jobs that it wants to keep secret. If a printer doesn't support the job attribute number-of-intervening-jobs, a mapper MAY use the job's position.

Note: a Printer may set the value of job-originating-user-name to the authenticated user or to the value of "requesting-user-name", depending on the implementation and configuration. For a gateway, the authenticated user is the user-id of the gateway, but the "requesting-user-name" may contain the name of the user who is the gateway's client.

In order to obtain the information specified above, The LPD-to-IPP mapper ~~SHALL~~MUST use the Get-Printer-Attributes operation to get printer-status and SHOULD use the Get-Jobs operation to get information about all of the jobs. If the LPD command contains job-numbers or user-names, the mapper MAY handle the filtering of the response. If the LPD command contains job-numbers but no user-names, the mapper MAY use Get-Job-Attributes on each converted job-number rather than Get-Jobs. If the LPD command contains a single user-name but no job-numbers, the mapper MAY use Get-Jobs with the my-jobs option if the server supports this option and if the server allows the client to be a proxy for the LPD user.

NOTE: This specification does *not* define how the mapper maps the LPD Printer-name operand to the IPP "printer-uri" parameter.

3.4 Send queue state (long)

Command syntax:

send-queue-long = %x04 printer-name *(SP(user-name / job-number)) LF

The mapper's response to this command includes information about the printer and its jobs. RFC 1179 specifies neither the information nor the format of its response. This document requires the mapper to follow existing practice as specified in this document.

The mapper ~~SHALL~~MUST produce a response in the following format which consists of a printer-status line optionally followed a list of jobs, where each job consists of a blank line, a description line, and one line for each file. The description line contains the user-name, rank, job-number and host. This format is defined by examples below. Appendix B contain the ABNF syntax.

For an printer with no jobs the response is:

```
no entries
```

For a printer with jobs, an example of the response is:

```
~~killtree~~pinetree is ready and printing

fred: active                          [job 123 tiger]
        2 copies of stuff             602 bytes

smith: 1st                            [job 124 snail]
        2 copies of resume            7088 bytes
        2 copies of foo               10200 bytes

fred: 2nd                             [job 125 tiger]
        more                          99 bytes
```

The column numbers of above headings and job entries are:

```
|       |                             |
01      09                            41
```

Although the format of the long form is different from the format of the short form, their fields are identical except for a) the copies and host fields which are only in the long form, and b) the "size" field contains the single copy size of each file. Thus the sum of the file sizes in the "size" field times the value of the "copies" field produces the value for the "Total Size" field in the short form. For fields other than the host and copies fields, see the preceding section. For the host field see the table below.

LPD field	IPP attribute	special conversion details
host		unspecified conversion; job-originating-host may be the mapper's host
~~copies~~	~~copies~~	~~the mapper shall assume the value of copies precedes the string "copies of "; otherwise, the value of copies is 1.~~
copies	copies	the mapper MUST assume the value of copies precedes the string "copies of "; otherwise, the value of copies is 1.

NOTE: This specification does *not* define how the mapper maps the LPD Printer-name operand to the IPP printer-uri parameter.

3.5 Remove jobs

Command syntax:

```
remove-jobs = %x05 printer-name SP agent
              *(SP(user-name / job-number)) LF
```

The agent operand is the user-name of the user initiating the remove-jobs command. The special user-name 'root' indicates a privileged user who can remove jobs whose user-name differs from the agent..

The mapper ~~SHALL~~MUST issue one Cancel-Job operation for each job referenced by the remove-jobs command. Each job-number in the remove-jobs command references a single job. Each user-name in the remove-jobs command implicitly references all jobs owned by the specified user. The active job is implicitly referenced when the remove-jobs command contains neither job-numbers nor user-names. The mapper MAY use Get-Jobs to determine the job-uri of implicitly referenced jobs.

The mapper ~~SHALL~~MUST not use the agent name of 'root' when end-users cancel their own jobs. Violation of this rule creates a potential security violation, and it may cause the printer to issue a notification that misleads a user into thinking that some other person canceled the job.

If the agent of a remove-jobs command for a job J is the same as the user name specified with the 'P' function in the control file for job J, then the mapper ~~SHALL~~MUST ensure that the caller of the Cancel-Job command for job J is the same as job-originating-user for job J.

Note: This requirement means that a mapper must be consistent in who the receiver perceives as the caller of IPP operations. The mapper either acts as itself or acts on behalf of another user. The latter is preferable if it is possible. This consistency is necessary between Print-Job/Create-Job and Cancel-Job in order for Cancel-Job to work, but it is also desirable for other operations. For example, Get-Jobs may give more information about job submitted by the caller of this operation.

NOTE: This specification does *not* define how the mapper maps: (1) the LPD printer-name to the IPP "printer-uri" or (2) the LPD job-number to the IPP "job-uri".

NOTE: This specification does not specify how the mapper maps the LPD user-name to the IPP job-originating-user because the mapper may use its own user-name with jobs.

4. Mapping of LPD Control File Lines to IPP Parameters

This section describes the mapping from LPD control file lines (called 'functions') to IPP operation input parameters. The mapper receives the control file lines via the LPD receive-control-file sub-command.. Each of the LPD functions appear as sub-sections of section 7 of RFC 1179.

In LPD control file lines, the text operands have a maximum length of 31 or 99 while IPP input parameters have a maximum of 255 characters. Therefore, no data is lost.

The mapper converts each supported LPD function to its corresponding IPP parameter as defined by tables in the subsections that follow. These subsections group functions according to whether they are:

- required with a job,
- optional with a job
- required with each document.

In the tables below, each LPD value is given a name, such as 'h'. If an IPP value uses the LPD value, then the IPP value column contains the LPD name, such as 'h' to denote this. Otherwise, the IPP value column specifies the literal value.

4.1 Required Job Functions

The following LPD functions MUST be in a received LPD job. The mapper ~~SHALL~~MUST receive each of the following LPD functions and ~~SHALL~~MUST include the information as a parameter with each IPP job. The functions SHOULD be in the order

‘H’, ‘P’ and they SHOULD be the first two functions in the control file, but they MAY be anywhere in the control file and in any order.

LPD function name	value	description	IPP name	value
H	*h*	Originating Host		*h* (in security layer)
P	*u*	User identification	requesting-user-name	*u* (and in security layer)
		none	ipp-attribute-fidelity	‘true’

A mapper MAY sends its own host rather than the client’s host, and a mapper MAY send its own user-name as user identification rather than the client user. But in any case, the values sent ~~SHALL~~MUST be compatible with the Cancel-Job operation. The IPP operation MAY have no way to specify an originating host-name.

The mapper ~~SHALL~~MUST include ipp-attribute-fidelity =true so that it doesn’t have to determine which attributes a printer supports.

4.2 Optional Job Functions

The following LPD functions MAY be in a received job. These function SHOULD follow the required job functions and precede the document functions, but they MAY be anywhere in the control file.

If the mapper receives such an LPD function, the mapper ~~SHALL~~MUST include the corresponding IPP attribute with the value converted as specified in the table below. If the mapper does not receive such an LPD attribute, the mapper ~~SHALL~~MUST NOT include the corresponding IPP attribute, except the ‘L’ LPD function whose absence has a special meaning as noted in the table.

LPD function name	value	description	IPP name	value
J	*j*	Job name for banner page	job-name	*j*
L	*l*	Print banner page	job-sheets	‘standard’ if ‘L’ is present ‘none’ if ‘L’ is present
M	*m*	Mail When Printed		IPP has no notification mechanism. To support this LPD feature, the gateway must poll

~~.~~

4.3 Required Document Functions

The mapper ~~SHALL~~MUST receive one set of the required document functions with each copy of a document, and ~~SHALL~~MUST include the converted information as parameters with each IPP document

If the control file contains required and recommended document functions, the required functions SHOULD precede the recommended ones and if the job contains multiple documents, all the functions for each document are grouped together as shown in the example of section 6.3 “Required Document Functions”. However, the document functions MAY be in any order.

LPD function	IPP

name	value	description	name	value
f	fff	Print formatted file	document-format	'application/octet-stream'
l	fff	Print file leaving control characters	document-format	'application/octet-stream'
o	fff	Print Postscript output file	document-format copies	'application/PostScript' see note

Note: In practice, the 'f' LPD function is often overloaded. It is often used with any format of document data including PostScript and PCL data.

Note: In practice, the 'l' LPD function is often used as a rough equivalent to the 'f' function.

Note: When RFC 1179 was written, no implementation supported the 'o' function; instead 'f' was used for PostScript. Windows NT now sends 'o' function for a PostScript file.

Note: the value 'fff' of the 'f', 'l' and 'o' functions is the name of the data file as transferred, e.g. "dfA123woden".

If the mapper receives any other lower case letter, the mapper ~~SHALL~~MUST reject the job because the document contains a format that the mapper does not support.

The mapper determines the number of copies by counting the number of occurrences of each 'fff' file with one of the lower-case functions above. For example, if 'f dfA123woden' occurs 4 times, then copies has a value of 4. Although the LPD protocol allows the value of copies to be different for each document, the commands and the receiving print systems don't support this.

4.4 Recommended Document Functions

The mapper SHOULD receive one set of the recommended document functions with each document, and SHOULD include the converted information as parameters with each IPP document. The functions SHOULD be received in the order 'U' and 'N', but they MAY arrive in any order.

LPD function name	value	description	IPP name	value
U	*fff*		*ignored*	
N	*n*	Name of source file	document-name	*n*

Note: the value '*fff*' of the 'U' function is the name of the data file as transferred, e.g. "dfA123woden".

5. Mapping from IPP operations to LPD commands

If the IPP-to-LPD mapper receives an IPP operation, the following table summarizes the LPD command that it uses. Each section below gives the detail. Each of the following sub-sections appear as sub-sections of section 3 in the document "Internet Printing Protocol/1.0: Model and Semantics" [ipp-mod].

IPP operation	LPD command
Print-Job or Print-URI or Create-Job/Send-Document/Send-URI	receive-a-printer-job and then print-any-waiting-jobs
Validate-Job	implemented by the mapper
Cancel-Job	remove-jobs

Get-Printer-Attributes, Get-Job-Attributes or Get-Jobs send queue state (short or long)

5.1 Print-Job

The mapper ~~SHALL~~MUST send the following commands in the order listed below:

- receive-a-printer-job command
- both receive-control-file sub-command and receive-data-file sub-command (unspecified order, see Note below)
- print-any-waiting-jobs command, except that if the mapper is sending a sequence of receive-a-printer-job commands, it MAY omit sending print-any-waiting-jobs after any receive-a printer-job command that is neither the first nor last command in this sequence

Note: it is recommended that the order of the receive-control-file sub-command and the receive-data-file sub-command be configurable because either order fails for some print systems. Some print systems assume that the control file follows all data files and start printing immediately on receipt of the control file. When such a print system tries to print a data file that has not arrived, it produces an error. Other print systems assume that the control file arrives before the data files and start printing when the first data file arrives. Such a system ignores the control information, such as banner page or copies.

NOTE: This specification does not define the mapping between the IPP printer-uri and the LPD printer-name.

The mapper ~~SHALL~~MUST send the IPP parameters and attributes received from the operation to the LPD printer by using the LPD receive-control-file sub-command. The mapper ~~SHALL~~MUST create the LPD job-number for use in the control file name, but the receiving printer MAY, in some circumstances, assign a different job-number to the job. The mapper ~~SHALL~~MUST create the IPP job-id and IPP job-uri returned in the Print-Job response.

NOTE: This specification does not specify how the mapper determines the LPD job-number, the IPP job-id or the IPP job-uri of a job that it creates nor does it specify the relation ship between the IPP job-uri, IPP the job-id and the LPD job-number, both of which the mapper creates. However, it is likely that the mapper will use the same integer value for both theLPD job-number and the IPP job-id, and that the IPP Job-uri is the printer's URI with the job-id concatenated on the end.

The mapper ~~SHALL~~MUST send data received in the IPP operation to the LPD printer by using the LPD receive-data-file sub-command. The mapper ~~SHALL~~MUST specify the exact number of bytes being transmitted in the number-of-bytes field of the receive-data-file sub-command. It ~~SHALL~~MUST NOT use a value of 0 in this field.

If the mapper, while it is transmitting a receive-a-printer-job command or sub-command, either detects that its IPP connection has closed or receives a Cancel-Job operation, the mapper ~~SHALL~~MUST terminate the LPD job either with the abort sub-command or the remove-jobs command.

~~ISSUE: error code conversion.~~Error code conversion is not specified in this document..

5.2 Print-URI

The mapper ~~SHALL~~MUST handle this operation in the same way as a Print-Job operation except that it ~~SHALL~~MUST obtain data referenced by the "document-uri" parameter and ~~SHALL~~MUST then treat that data as if it had been received via a Print-Job operation.

5.3 Validate-Job

The mapper ~~SHALL~~MUST perform this operation directly. Because LPD supports very few attributes, this operation doesn't have much to check.

5.4 Create-Job

The mapper ~~SHALL~~MUST handle this operation like Print-Job, except

- the mapper ~~SHALL~~MUST send the control file after it has received the last Send-Document or Send-URI operation because the control file contains all the document-name and document-format values specified in the Send-Document and Send-URI operations.
- the mapper ~~SHALL~~MUST perform one receive-data-file sub-command for each Send-Document or Send-URI operation received and in the same order received.
- the mapper ~~SHALL~~MUST send the control file either before all data files or after all data files. (See the note in the section on Print-Job about the dilemma of sending the control file either before or after the data files.

5.5 Send-Document

The mapper performs a receive-data-file sub-command on the received data. See the preceding section 5.4 "Create-Job" for the details.

5.6 Send-URI

The mapper ~~SHALL~~MUST obtain the data referenced by the "document-uri" parameter, and ~~SHALL~~MUST then treat that data as if it had been received via a Send-Document operation. See the preceding section 5.5 "Send-Document" for the details.

5.7 Cancel-Job

~~□~~The mapper ~~SHALL~~MUST perform a remove-jobs command with the following parameters:

- the printer is the one to which the job was submitted, that is the IPP printer-uri is mapped to an LPD printer-name by the same mechanism as for all commands.~~,~~
- the agent is the authenticated user-name of the IPP client,
- the job-number is the job-id returned by the Print-Job command, that is, the LPD job-number has the same value as the IPP job-id for likely implementations.~~.~~

5.8 Get-Printer-Attributes

LPD severely limits the set of attributes that the mapper is able to return in its response for this operation. The mapper ~~SHALL~~MUST support, at most, the following printer attributes:

- printer-state
- printer-state-reasons

The mapper uses either the long or short form of the "send queue state" command.

The mapper ~~SHALL~~MUST assume that the LPD response that it receives has the format and information specified in section 3.3 "Send queue state (short)" and section 3.4 "Send queue state (long)". The mapper ~~SHALL~~MUST determine the value of each requested attribute by using the inverse of the mapping specified in the two aforementioned sections.

Note: the mapper can determine the response from the printer-status line without examining the rest of the LPD response.

5.9 Get-Job-Attributes

LPD severely limits the set of attributes that the mapper is able to return in its response for this operation. The mapper ~~SHALL~~MUST support, at most, the following job attributes:

- number-of-intervening-jobs
- job-originating-user-name
- job-id
- document-name
- job-k-octets
- copies

The mapper uses either the long or short form of the "send queue state" command. If it receives a request for the "job-k-octets" or "copies" and supports the attribute it ~~SHALL~~MUST use the long form; otherwise, it ~~SHALL~~MUST use the short form.

Note: the value of job-k-octets is the value in the short form divided by the number of "copies" which is on the long form only. Its value can also be determined by adding the "size" field values for each document in the job in the long form.

The mapper ~~SHALL~~MUST assume that the LPD response that it receives has the format and information specified in section 3.3 "Send queue state (short)" and section 3.4 "Send queue state (long)". The mapper ~~SHALL~~MUST determine the value of each requested attribute by using the inverse of the mapping specified in the two aforementioned sections.

Note: when the mapper uses the LPD short form, it can determine the response from the single LPD line that pertains to the job specified by the Get-Job-Attributes operation.

NOTE: the mapper can use its correspondence between the IPP job-id, job-uri and the LPD job-number.

5.10 Get-Jobs

The mapper ~~SHALL~~MUST perform this operation in the same way as Get-Job-Attributes except that the mapper converts all the LPD job-lines, and the IPP response contains one job object for each job-line in the LPD response..

6. Mapping of IPP Parameters to LPD Control File Lines

This section describes the mapping from IPP operation input parameters to LPD control file lines (called 'functions'). The mapper receives the IPP operation input parameters via the IPP operation. Each of the IPP operation input parameters appear as sub-sections of section 3 and 4.2 in the IPP model document [ipp-mod].

In the context of LPD control file lines, the text operands have a maximum length of 31 or 99 while IPP input parameters have a maximum of 255 characters. Therefore, there may be some data loss if the IPP parameters exceed the maximum length of the LPD equivalent operands.

The mapper converts each supported IPP parameter to its corresponding LPD function as defined by tables in the subsections that follow. These subsections group functions according to whether they are:

- required with a job,
- optional with a job
- required with each document.

In the tables below, each IPP value is given a name, such as 'h'. If an LPD value uses the IPP value, then the LPD value column contains the IPP name, such as 'h' to denote this. Otherwise, the LPD value column specifies the literal value.

6.1 Required Job Functions

The mapper ~~SHALL~~MUST include the following LPD functions with each job, and they ~~SHALL~~MUST have the specified value. They ~~SHALL~~MUST be the first functions in the control file and they ~~SHALL~~MUST be in the order "H" and then "P".

IPP name	**value**	**LPD function name**	**value**	**description**
(perhaps in security layer)	*h*	H	*gateway host*	Originating Host
requesting-user-name and in the security layer	*u*	P	*u*	User identification

A mapper ~~SHALL~~MUST sends its own host rather than the client's host, because some LPD systems require that it be the same as the host from which the remove-jobs command comes. A mapper MAY send its own user name as user identification rather than the client user. But in any case, the values sent ~~SHALL~~MUST be compatible with the LPD remove-jobs operation.

6.2 Optional Job Functions

The mapper MAY include the following LPD functions with each job. They ~~SHALL~~MUST have the specified value if they are sent. These functions, if present, ~~SHALL~~MUST follow the require job functions, and they ~~SHALL~~MUST precede the required document functions.

IPP attribute name	**value**	**LPD function name**	**value**	**description**
job-name	*j*	J	j	Job name for banner page
job-sheets	'standard'	L	*u*	Print banner page
job-sheets	'none'			omit 'L' function

Note: 'L' has special meaning when it is omitted. If 'J' is omitted, some undefined behavior occurs with respect to the banner page.

6.3 Required Document Functions

The mapper ~~SHALL~~MUST include one set of the following LPD functions with each document, and they ~~SHALL~~MUST have the specified values. For each document, the order of the functions ~~SHALL~~MUST be 'f', 'U' and then 'N', where 'f' is replicated once for each copy.

IPP attribute name	IPP attribute value	LPD function name	LPD function value	description
document-format	'application/octet-stream' or 'application/PostScript'	f	*fff*	Print formatted file
copies	*c*			replicate 'f' 'c' times
none		U	*fff*	Unlink data file
document-name	*n*	N	*n*	Name of source file

Note: the value '*fff*' of the 'f' and 'U' functions is the name of the data file as transferred, e.g. "dfA123woden".

Note: the mapper ~~SHALL not~~MUST NOT send the 'o' function

ISSUE: should we register DVI, troff or ditroff?

If the mapper receives no "ipp-attribute-fidelitybest-effort" or it has a value of false, then the mapper ~~SHALL~~MUST reject the job if it specifies attributes or attribute values that are not among those supported in the above tables.

Below is an example of the minimal control file for a job with three copies of two files 'foo' and 'bar':

```
H tiger
P jones
f dfA123woden
f dfA123woden
f dfA123woden
U dfA123woden
N foo
f dfB123woden
f dfB123woden
f dfB123woden
U dfB123woden
N bar
```

7. Security Considerations

There are no security issues beyond those covered in the IPP ~~protocol~~Encoding and Transport document [ipp-pro], the IPP model document [ipp-mod] and the LPD document [rfc1179].

8. References

[ipp-lpd] Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", draft-ietf-ipp-lpd-ipp-map-04.txt, June 1998.

[ipp-mod] ~~R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet~~Isaacson, S., deBry, R., Hastings, T., Herriot, R., Powell, P., "Internet Printing Protocol/1.0: Model and ~~Semantics", <draft-ietf-ipp-model-07.txt>, November 1997.~~Semantics" draft-ietf-ipp-mod-10.txt, June, 1998.

~~[ipp-pro] R. Herriot, S. Butler, P. Moore, R. Turner, "Internet~~[ipp-pro] Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: ~~Protocol specification", <draft-ietf-ipp-protocol-03.txt>, November 1997.~~Encoding and Transport", draft-ietf-ipp-pro-06.txt, June, 1998.

[ipp-rat] Zilles, S., “Rationale for the Structure and Model and Protocol for the Internet Printing Protocol”, draft-ietf-ipp-rat-03.txt, June, 1998.

[ipp-req] Wright, D., "Design Goals for an Internet Printing Protocol", draft-ietf-ipp-req-02.txt, June, 1998.

[rfc1179] L. McLaughlin, "Line Printer Daemon Protocol", RFC 1179, August 1990.

[rfc1759] Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759, March 1995.

[rfc2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March 1997

[abnf] D. Crocker et al., “Augmented BNF for Syntax Specifications: ABNF”, draft-ietf-drums-abnf-05.txt.

9. Author’s Addresses

Robert Herriot (editor)
Sun Microsystems Inc.
901 San Antonio.Road., MPK-17
Mountain View, CA 94043

Phone: 650-786-8995
Fax: 650-786-7077
Email: robert.herriot@eng.sun.com

Norm Jacobs
Sun Microsystems Inc.
1430 Owl Ridge Rd.
Colorado Springs, CO 80919

Phone: 719-532-9927
Fax: 719-535-0956
Email: Norm.Jacobs@Central.sun.com

Thomas N. Hastings
Xerox Corporation
701 S. Aviation Blvd., ESAE-231
El Segundo, CA 90245

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

Jay Martin
Underscore, Inc.
41-C Sagamore Park Road
Hudson, NH 03051-4915

Phone: 603-889-7000
Fax: 603-889-2699
Email: jkm@underscore.com

10. Appendix A: ABNF Syntax for response of Send-queue-state (short)

The syntax in ABNF for the response to the LPD command ‘send-queue-state (long)’ is:

```
status-response = empty-queue / nonempty-queue
empty-queue = “no-entries” LF
nonempty-queue = printer-status LF heading LF *(job LF)
printer-status =  OK-status / error-status
OK-status = printer-name SP “ready and printing” LF
error-status = < implementation dependent status information >
heading = “Rank” 3SP “Owner” 6SP “Job” 13SP “Files”
            23SP “Total Size” LF
              ; the column headings and their values below begin at the columns
              ; 1, 8, 19, 35 and 63
job = rank *SP owner *SP job *SP files *SP total-size “bytes”
                    ; jobs are in order of oldest to newest
rank = “active” / “1st” / “2nd” / “3rd” / integer “th”
              ; job that is printing is “active”
```

539 ; other values show position in the queue
540 owner = <user name of person who submitted the job>
541 job = 1*3DIGIT ; job-number
542 files = <file name> *(“,” <file name>) ; truncated to 24 characters
543 total-size = 1*DIGIT ; combined size in bytes of all documents

544 11. Appendix B: ABNF Syntax for response of Send-queue-state (long)

545 The syntax in ABNF for the response to the LPD command ‘send-queue-state (long)’ is:

546 status-response = empty-queue / nonempty-queue
547 empty-queue = “no-entries” LF
548 nonempty-queue = printer-status LF *job
549 printer-status = OK-status / error-status
550 OK-status = printer-name SP “ready and printing” LF
551 error-status = < implementation dependent status information >
552 job = LF line-1 LF line-2 LF
553 line-1 = owner “:” SP rank 1*SP “[job” job SP host “]”
554 line-2 = file-name 1*SP document-size “bytes”
555 ; jobs are in order of oldest to newest
556 rank = “active” / “1st” / “2nd” / “3rd” / integer “th”
557 ; job that is printing is “active”
558 ; other values show position in the queue
559 owner = <user name of person who submitted the job>
560 job = 1*3DIGIT
561 file-name = [1*DIGIT “copies of” SP] <file name>
562 ; truncated to 24 characters
563 document-size = 1*DIGIT ;size of single copy of the document.

564 12. Appendix C: Unsupported LPD functions

565 The follow LPD functions have no IPP equivalent. The LPD-to-IPP mapper ignores them and the IPP-to-LPD mapper does not
566 send them.

LPD command name	**description**
C	Class for banner page
I	Indent Printing
H	Host of client
M	Mail when printed
S	Symbolic link data
T	Title for pr
W	Width of output
1	troff R font
2	troff I font
3	troff B font
4	troff S font

567

568 The follow LPD functions specify document-formats which have no IPP equivalent, unless someone registers them. The LPD-to-
569 IPP mapper rejects jobs that request such a document format, and the IPP-to-LPD mapper does not send them.

LPD command name	description
c	Plot CIF file
d	Print DVI file
g	Plot file
k	reserved for Kerberized clients and servers
n	Print ditroff output file
p	Print file with 'pr' format
r	File to print with FORTRAN carriage control
t	Print troff output file
v	Print raster file
z	reserved for future use with the Palladium print system

13. Appendix D: Full Copyright Statement