Tom Hastings
Xerox Corporation
RobertBob Herriot
Sun Microsystems, Inc.
Norm Jacobs
Sun Microsystems, Inc.
Jay Martin
Underscore, Inc.
July 3013, 1997

Mapping between LPD and IPP Protocols

<draft-ietf-ipp-lpd-ipp-map-01??.txt>

Expires Jan 3013, 1998

Status of this Memo

Abstract

This Internet-Draft 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.

WARNING: RFC 1179 was not on standards track. While RFC 1179 was intended to record existing practice, in some areas 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 implementeors.

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 [1], the IPP Protocol specification [2], and the Line Printer Daemon (LPD) protocol specification [3] 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~~directives~~. 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 ~~so~~ 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[4] and IPP[1] [2], though they could be registered if desired. See the Printer MIB specification [4] and/or the IPP Model specification [1] 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".

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [6].

RFC 1179 uses the word "command" in two contexts: for over-the-wire operations and for command file functions. This document SHALL use the word "command" for the former and the phrase "functions" for the latter.

The syntax of the LPD commands is given using ABNF [6].

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 ~~on the wire~~ 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-Attributes (printer) and Get-Jobs
remove-jobs	Cancel-Job

3.1 Print any waiting jobs

Command syntax:

print-waiting-jobs = %x01 p~~P~~rinter- ~~queue-~~name LF

~~In LPD, this comment starts the daemon, if it isn't already running. Such an equivalent operation is not provided in IPP, since the IPP Printer is assumed to always be running, where as in LPD, the client makes sure that the daemon is running using this command.~~

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 ~~an LPD-to-IPP~~ mapper receives this LPD command, it SHALL ignore it and send no IPP operation.

3.2 Receive a printer job

Command syntax:

receive-job = %x02 p~~P~~rinter- ~~queue-~~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 ~~with~~ a single data file or
- the Create-Job operation followed by one~~a~~ 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 PrintJob operation, but MAY use the Create-Job and Send-Document operations.
- more that one data file, the mapper SHALL 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 use the PrintJob operation.
- more that one data file, the mapper submit each received LPD data file as a separate Print-Job operation (thereby converting a single LPD job into multiple IPP jobs).

~~If a job consists of a single data file, the PrintJob operation is RECOMMENDED.~~

~~If a job consists of more than one data file, Create Job followed by Send-Document for each data file is REQUIRED. If the IPP Printer doesn't support the Create-Job and Send-Document operations, the LPD-to-IPP mapper SHALL submit each 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.

~~ISSUE: Ok that I changed so that the mapper shall break a multi-document job into separate jobs, one IPP job for each LPD data file, instead of error return?~~

~~NOTE: if Create-Job is used, it MUST precede the Send-Document operation even if the LPD control file, which supplies attributes for Create-Job, arrives after all documents.~~NOTE: This specification does *not* specify how the mapper maps: the LPD Printer-name operand to the IPP "printer-uri" parameter.

~~This section describes the mapping between LPD sub-commands and IPP operations. Each of the following sub-sections appear as sub-sections of section 6 of RFC 1179. The operands of the sub-commands appear in parens in the sub-headings~~

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

3.2.1 ~~01 –~~ Abort job ~~()~~

Sub-command syntax:

abort-job = %x~~0~~1 LF

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

If the mapper receives this sub-command, it SHALL 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 it performs 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.

~~If an IPP Create-Job operation and/or a Send-Document operation were performed on behalf of the receive job command that is being aborted, an IPP Cancel-Job operation should be issued for the job URI that was~~

~~returned by the Printer on which the Create-Job operation was performed. Also, any temporary files created while processing the 'Receive job request' should be cleaned up, and the connection to the client should be closed. Finally, 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 job request'.~~

~~ISSUE: is IPP defined at this point to abort a job whose connection is closed before the job has been fully received. If so, that is an alternate and simpler way to abort the job.~~

3.2.1 ~~02 –~~ Receive control file

~~Sub-command syntax: %x02 Number-of-bytes-in-control-file, Name-of-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

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

The mapper SHALL 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.

~~Once the control file has been has been received, it’s contents should be translated, and an appropriate IPP Create-Job operation performed.~~

~~However, some information, such as Document-Name go in the Send-Document operation.~~

3.2.1 ~~03 –~~ Receive data file

Sub-command syntax: %x~~0~~3 n~~N~~umber-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

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

The mapper SHALL 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 reject a job that has a value of 0 in the number-of-bytes field.

~~If the control file has been previously received, and it's corresponding IPP Create-Job operation performed, an IPP Send-Document operation can be performed using the job URI returned by the IPP Create-Job operation.~~

~~When performing the Send-Document operation, the size of the document data MUST be specified. Unfortunately RFC-1179 alludes to a method for passing an arbitrary length data file. This is described as being done by using an octet-count of zero, however the description isn't complete, and in practice, no implementations allow sending or receiving arbitrary length data files.~~

3.3 Send queue state (short)

Command syntax: ~~%x03 Printer-queue-name *(SP (User-Name / job-number))~~

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

~~If the LPD command contains only the Printer-queue-name operand, the LPD-to-IPP mapper SHALL use the Get-Attributes operation of the corresponding IPP Printer to get printer-state information and the Get-Jobs operation of the Printer to get information about all of the jobs. With Get-Attributes, it SHALL request the "printer-state" and "printer-state-reasons" attributes. With Get-Jobs, it SHALL request the "number-of-intervening-jobs", "job-originating-user", "job-name", "document-name" (or "document-uri"), and "job-k-octets".~~

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 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 is:

```
no entries
```

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

```
killtree is ready and printing
Rank    Owner       Job             Files                   Total Size
active  fred        123             stuff                   1204 bytes
1st     smith       124             resume                  34576 bytes
2nd     fred        125             more                    99 bytes
```

The column numbers of above headings and job entries are:

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

The mapper SHALL 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.
rank	number-of-intervening-jobs	the mapper SHALL the format above
owner	job-originating-user	unspecified conversion; job-originating-user may be the mapper's user-name
job	job-uri	unspecified conversion
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
total-size	job-k-octets	the mapper shall multiple the value of job-k-octets by 1024.

ISSUE: is job-k-octets the sum of the bytes of each document times the number of copies? If so, "total-size" is 1024 times job-k-octets. The model document needs clarification.

In order to obtain the information specified above, The LPD-to-IPP mapper SHALL use the Get-Attributes operation of the printer 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 handles the filtering of the response because Get-Jobs has no way to limit jobs to those of a particular user. If the LPD command contains job-numbers but no user-names, the mapper MAY use Get-Attributes on each converted job-number rather than Get-Jobs.

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

~~NOTE: This specification does *not* specify how the LPD-to-IPP mapper maps: (1) the LPD Printer-queue-name operand to the IPP "printer-uri" parameter or (2) the LPD job-number operand to the IPP "job-uri" parameter, since the format of these URIs is opaque in the IPP protocol and is implementation-dependent.~~

~~NOTE: RFC 1179 does not specify what attributes are returned in response to a 'Send queue state' (short) command, but leaves it up to implementation. The IPP attributes specified in this specification reflect existing practice.~~

~~If the LPD command contains one or more User-name operands, the LPD-to-IPP mapper SHALL get all the jobs as above using the Get-Jobs operation on the Printer and then do its own filtering on the returned value of the "job-originating-user" attribute for each job.~~

~~If the LPD command contains only job-number operands, the LPD-to-IPP mapper SHALL either (1) get all the jobs as above using the Get-Jobs operation on the Printer and then do its own filtering or (2) get each specified job individually using separate Get-Attributes operations (multiple jobs may be requested in a single IPP connection with multiple Get-Attribute operations, one for each job).~~

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 produce a response in the following format which consists of a printer-status line optionally followed by blank line, and a list of jobs. 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 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, the fields are identical to the short form except for the copies and host fields which areonly in the long form. For fields other than the host field, see the preceding section. For the host field see the table below.

LPD field	IPP attribute	special conversion details
host	job-originating-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.

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

~~Same mapping as the 'Send queue state' (short) command. The IPP client supplies a longer list of requested attributes to the Get-Jobs or Get-Attributes operations.~~

~~The LPD-to-IPP mapper should specify additional attributes than the ones listed for the 'Send queue state' (short) command.~~

~~NOTE: RFC 1179 does not specify what attributes are returned in response to a 'Send queue state' (short) command, but leaves it up to implementation.~~

3.5 Remove jobs

Command syntax: ~~%x05 Printer-queue-name SP agent *(SP (User-name / job-number))~~

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 ~~'R~~remove~~'~~ command.~~-~~ The special user-name 'root' indicates a privileged user who can remove jobs whose user-name differs from the agent..

~~The LPD-to-IPP mapper shall map this command to the Cancel-Job operation.~~

The mapper SHALL 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-Job to determine the job-uri of implicitly referenced jobs.

The mapper SHALL 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 remove-jobs command is to succeed for a job J with an agent which is an end user name, then the agent SHALL be the same as the user name specified with the 'P' function in the receive-a-printer-job command for job J. The mapper SHALL have the same alignment between the job-originating-user and the caller of the Cancel-Job command. This requirement means that a mapper either can act on behalf of another user, or all jobs it submits have a job-originating-user of the mapper's user-name.

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".

~~There is no IPP equivalent for the LPD 'Remove jobs' command with just the Printer-queue-name operand supplied, since IPP provides no way to cancel the current job.~~

~~There is no IPP equivalent for the LPD 'Remove jobs' command with a User-name operand supplied, since IPP provides no way to cancel a job specified by user name.~~

~~This command with the Printer-queue-name operand and one job-number operand is the same as the IPP Cancel-Job operation when the client supplies just the job URI. Multiple jobs may be canceled in IPP in a single connection with multiple Cancel-Job operations. In IPP only a privileged operator may cancel jobs belonging to another user.~~

~~NOTE: This specification does *not* specify how the LPD-to-IPP mapper maps: (1) the LPD Printer-queue-name to the IPP "printer-uri" or (2) the LPD job-number to the IPP "job-uri", since the format of these URIs is opaque in the IPP protocol and is implementation-dependent.~~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.

~~The LPD-to-IPP mapper shall map a Cancel-Job operation to this command.~~

4. ~~There are some major issues about setting the agent.~~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. ~~for the~~The mapper receives the control file lines via the LPD receive-control-file sub-command. ~~Print-Job, Create-Job, and Send-Document operations~~. Each of the LPD functions ~~following sub-sections~~ 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.

~~ISSUE: somewhere, we need to map the LPD query format to IPP attributes.~~

~~In LPD 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 when mapping from LPD to IPP. However, when mapping from IPP to LPD, there may be some data loss if the IPP parameters exceed the maximum length of the LPD equivalent operands.~~

~~In the following table, IPP input parameter names are indicated in double quotes (") and input parameter values are indicated in single quotes (). Values of the IPP "document-format" attribute that could be registered, but are not currently, are indicated with "**".~~

~~Where there is a one-to-one mapping, both directions are specified. Where IPP has none, the LPD-to-IPP the attribute is ignored, and in the IPP-to-LPD the LPD feature is left unspecified.~~

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.

3.6 Required Job Functions

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

LPD function			**IPP**	
name	**value**	**description**	**name**	**value**
H	*h*	Originating Host		*h* (in security layer)
P	*u*	User identification		*u* (in security layer)
		none	best-effort	'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 be compatible with the Cancel-Job operation. The IPP operation MAY have no way to specify an originating host-name.

The mapper SHALL include best-effort=true so that it doesn't have to determine which attributes a printer supports.

3.7 Optional Job Functions

The follow LPD functions MAY be in a received job. If the mapper receives such an LPD function, the mapper SHALL 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 NOT include the corresponding IPP attribute, except the 'L' LPD function whose absence has a special meaning as noted in the table.

LPD function			**IPP**	
name	**value**	**description**	**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	notification-events	'job-completion'
			notification-method	'mailto://*'m'@'h*

Note: '*m*' is the user name and not an email address. The mapper can fabricate an email address with the source host. This email address mail fail when the 'h' is some intermediary host that doesn't know about user 'm'. But there is no better solution.

3.8 Recommended Document Functions

The mapper SHOULD receive one set of the following LPD functions with each document, and SHALL include the converted information as parameters with each IPP document

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

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

3.9 Required Document-format Functions

The mapper SHALL receive exactly one of the following LPD functions with each document, and SHALL include the converted information as parameters with each IPP document

LPD function			IPP	
name	value	description	name	value
f	fff	Print formatted file	document-format	37 (langAutomatic)
l	fff	Print file leaving control characters	document-format	37 (langAutomatic)
o	fff	Print Postscript output file	document-format	6 (langPS).

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 reject the job because the document contains a format that the mapper does not support.

ISSUE: should we register DVI, ditroff and troff. At least DVI and one of the troff is still used.

4. 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" [1].

IPP operation	LPD command
Get-Operations	implemented by the mapper
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-Attributes (printer or job) or Get-Jobs	send queue state (short or long)

4.1 Get-Operations

The mapper SHALL return a list of the operations that it supports. It SHALL support at least those operations that are mandatory according to the IPP model document [1].

4.2 Print-Job

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

- receive-a-printer-job command
- 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 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 create the 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 create the job-uri returned in the Print-Job response.

NOTE: This specification does not specify how the mapper determines the job-number or the job-uri of a job that it creates nor does it specify the relation ship between the job-uri and the job-number, both of which the mapper creates.

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

~~An IPP-to-LPD mapper SHALL send this LPD command after it has finished sending all pending 'Receive a printer job' command.~~

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 terminate the LPD job either with the abort sub-command or the remove-jobs command.

ISSUE: error code conversion.

4.3 Print-URI

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

4.4 Validate-Job

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

4.5 Create-Job

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

- the mapper SHALL 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 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 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.

4.6 Send-Document

The mapper performs a receive-data-file sub-command on the received data. See the preceding section 4.5 "Create-Job" for the details.

4.7 Send-URI

The mapper SHALL obtain the data referenced by the "document-uri" parameter, and SHALL then treat that data as if it had been received via a Send-Document operation. See the preceding section 4.6 "Send-Document" for the details.

~~An IPP-to-LPD mapper SHALL map the following IPP operations to this LPD command:~~

- ~~Print-Job~~
- ~~Print-uri~~
- ~~Create-Job followed by Send-Document or Send-URI for each document~~

~~See the next section for the mapping of the LPD "second level commands" to IPP input-parameters.~~

~~The IPP-to-LPD mapper shall use the long version of this command. See that command.~~

4.8 Cancel-Job

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

- the printer is the one containing the job specified by the IPP job-uri,
- the agent is the authenticated user-name of the IPP client,
- the job-number is the one corresponding to the IPP job-uri parameter.

NOTE: This specification does *not* specify how the mapper maps the IPP "job-uri" to the LPD printer-name or LPD job-number.

ISSUE: does the model offer a solution for mapping jobs to printers either with a new job attribute "printer-uri" or with all operation targets being a printer-uri.

4.9 Get-Attributes

LPD severely limits the set of attributes that the mapper is able to return in its response for this operation.

When the mapper receives a Get-Attributes operation for a printer object, it SHALL support, at most, the following printer attributes:

- printer-state

- printer-state-reasons

When the mapper receives a Get-Attributes operation for a job object, it SHALL support, at most, the following job attributes:

- number-of-intervening-jobs
- job-originating-user
- job-uri
- job-originating-host
- 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-originating-host” or “copies” and supports those attribute it SHALL use the long form; otherwise, it SHALL use the short form.

Note: the value of job-k-octets is the value in the short form, but it can be computed from the copies and file size fields in the long form.

The mapper SHALL 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 determine the value of each requested attribute by using the inverse of the mapping specified in the two aforementioned sections.

Note: when the mapper receives the Get-Attributes operation for a printer, it can determine the response from the printer-status line without examining the rest of the LPD response. When the mapper receives the Get-Attributes operation for a job and uses the LPD short form, it can determine the response from the single line that pertains to the job specified by the Get-Attributes operation.

NOTE: For Get-Attributes of a job, this specification does *not* specify how the mapper maps the IPP "job-uri" to the LPD printer-name or LPD job-number.

~~The IPP-to-LPD mapper shall use this command to get what attributes it can from the LPD server.~~

4.10 Get-Jobs

The mapper SHALL perform this operation in the same way as Get-Attributes of a printer except that the mapper converts the job-lines and the IPP response contains one job object for each job in the LPD response..

5. 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 [1].

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.

5.1 Required Job Functions

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

IPP		**LPD function**		
name	**value**	**name**	**value**	**description**
(in security layer)	*h*	H	*gateway host*	Originating Host
(in security layer)	*u*	P	*u*	User identification

A mapper SHALL 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 be compatible with the LPD remove-jobs operation.

5.2 Optional Job Functions

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

IPP attribute		**LPD function**		
name	**value**	**name**	**value**	**description**
job-name	*j*	J	j	Job name for banner page
job-sheets	'standard'	L	*u* or	Print banner page

			addr	
job-sheets	'none'			omit 'L' function
notification-events	'job-completion'	M	*user*	Mail When Printed
notification-method	'mailto://'*user@host*			

Note: 'L' has special meaning when it is omitted. If 'M' is omitted, there is no notification. If 'J' is omitted, some undefined behavior occurs with respect to the banner page.

Note: the 'user' for the 'M' function comes from a substring of the notification-method's value.

5.3 Required Document Functions

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

IPP attribute		**LPD function**		
name	**value**	**name**	**value**	**description**
document-format	'37' (langAutomatic) or '6' (langPS).	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 send the 'o' function

ISSUE: should we register DVI, troff or ditroff?

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

For example, three copies of two files 'foo' and 'bar' would have the minimal control file of

```
H tiger
P jones
f dfA123woden
f dfA123woden
f dfA123woden
U dfA123woden
```

N foo

f dfB123woden

f dfB123woden

f dfB123woden

U dfB123woden

N bar

6. References

[1] R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and Semantics", <draft-ietf-ipp-model-02.txt>, July 1997.

[2] R. Herriot, S. Butler, P. Moore, R. Turner, "Internet Printing Protocol/1.0: Protocol specification", <draft-ietf-ipp-protocol-00.txt>, July 1997.

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

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

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

[6] D. Crocker et al., “Augmented BNF for Syntax Specifications: ABNF”, draft-ietf-drums-abnf-03~~2~~.txt.

7. Author's Addresses

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

Robert Herriot
Sun Microsystems Inc.
901 San Antonio.Road~~2550 Garcia Ave~~., MPK-17
Mountain View, CA 94043

Phone: 650~~415~~-786-8995
Fax: 650~~-415~~-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

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

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

8. 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 newestrank = "active" / "1st" / "2nd" / "3rd" / integer "th"

; job that is printing is "active"

; other values show position in the queue

owner = <user name of person who submitted the job>

job = 1*3DIGIT ; job-number

files = <file name> *("," <file name>) ; truncated to 24 characters

total-size = < combined size in bytes of all documents.>

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

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 LF *(job LF)

printer-status = OK-status / error-status

OK-status = printer-name SP "ready and printing" LF

error-status = < implementation dependent status information >

job = line-1 LF line-2 LF LF

line-1 = owner ":" SP rank 1*SP "[job" job SP host "]"

line-2 = file-name 1*SP document-size "bytes"

; jobs are in order of oldest to newest

rank = "active" / "1st" / "2nd" / "3rd" / integer "th"

; job that is printing is "active"

; other values show position in the queue

owner = <user name of person who submitted the job>

job-number = 1*3DIGIT

file-name = <file name> / (1*DIGIT "copies of" SP <file name> ; truncated to 24 characters

document-size = < size of single copy of the document.>

10. Appendix C: Unsupported LPD functions

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

LPD command

name	**description**
C	Class for banner page
I	Indent Printing
S	Symbolic link data
T	Title for pr
~~U~~	~~Unlink data file~~
W	Width of output

1	troff R font
2	troff I font
3	troff B font
4	troff S font

The follow LPD functions specify document-formats which have no IPP equivalent, unless someone registers them. The LPD-to-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

ISSUE: we may move some of these to the supported list.