

Network Working Group
Request for Comments: 2567
Category: Experimental

F.D. Wright
Lexmark International
April 1999

Design Goals for an Internet Printing Protocol

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

IESG Note

This document defines an Experimental protocol for the Internet community. The IESG expects that a revised version of this protocol will be published as Proposed Standard protocol. The Proposed Standard, when published, is expected to change from the protocol defined in this memo. In particular, it is expected that the standards-track version of the protocol will incorporate strong authentication and privacy features, and that an "ipp:" URL type will be defined which supports those security measures. Other changes to the protocol are also possible. Implementers are warned that future versions of this protocol may not interoperate with the version of IPP defined in this document, or if they do interoperate, that some protocol features may not be available.

The IESG encourages experimentation with this protocol, especially in combination with Transport Layer Security (TLS) [RFC2246], to help determine how TLS may effectively be used as a security layer for IPP.

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. This 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. 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 full set of IPP documents includes:

Design Goals for an Internet Printing Protocol (this document)
Rationale for the Structure and Model and Protocol for the
Internet Printing Protocol [RFC2568]
Internet Printing Protocol/1.0: Model and Semantics [RFC2568]
Internet Printing Protocol/1.0: Encoding and Transport [RFC2565]
Internet Printing Protocol/1.0: Implementer's Guide [ipp-iig]
Mapping between LPD and IPP Protocols [RFC2569]

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 specifications, and gives background and rationale for the IETF working group's major decisions.

The "Internet Printing Protocol/1.0: Model and Semantics" document describes a simplified model consisting of abstract objects, their attributes, and their operations that is independent of encoding and transport. The model consists of a Printer and a Job object. The Job optionally supports multiple documents. IPP 1.0 semantics allow end-users and operators to query printer capabilities, submit print jobs, inquire about the status of print jobs and printers, and cancel print jobs. This document also addresses security, internationalization, and directory issues.

The "Internet Printing Protocol/1.0: Encoding and Transport" document is a formal mapping of the abstract operations and attributes defined in the model document onto HTTP/1.1. It defines the encoding rules for a new Internet media type called "application/ipp".

The "Internet Printing Protocol/1.0: Implementer's Guide" document gives insight and advice to implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.0 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.

TABLE OF CONTENTS

| | |
|---|----|
| 1. INTRODUCTION..... | 4 |
| 2. TERMINOLOGY..... | 4 |
| 3. DESIGN GOALS..... | 6 |
| 3.1. End-user..... | 6 |
| 3.1.1. Finding or locating a printer..... | 6 |
| 3.1.2. Create an instance of the printer..... | 7 |
| 3.1.3. Viewing the status and capabilities of a printer..... | 7 |
| 3.1.4. Submitting a print job..... | 8 |
| 3.1.5. Viewing the status of a submitted print job..... | 9 |
| 3.1.6. Canceling a Print Job..... | 9 |
| 3.2. Operator (NOT REQUIRED FOR V1.0)..... | 9 |
| 3.2.1. Alerting..... | 9 |
| 3.2.2. Changing Print and Job Status..... | 10 |
| 3.3. Administrator (NOT REQUIRED FOR v1.0)..... | 10 |
| 4. OBJECTIVES OF THE PROTOCOL..... | 10 |
| 4.1. SECURITY CONSIDERATIONS..... | 11 |
| 4.2. Interaction with LPD (RFC1179)..... | 12 |
| 4.3. Extensibility..... | 12 |
| 4.4. Firewalls..... | 13 |
| 4.5. Internationalization..... | 13 |
| 5. IPP SCENARIOS..... | 13 |
| 5.1. Printer Discovery..... | 14 |
| 5.2. Driver Installation..... | 15 |
| 5.3. Submitting a Print Job..... | 15 |
| 5.4. Getting Status/Capabilities..... | 16 |
| 5.5. Asynchronous Notification..... | 17 |
| 5.6. Job Canceling..... | 17 |
| 6. Security Considerations..... | 18 |
| 7. REFERENCES..... | 18 |
| 8. ACKNOWLEDGMENTS..... | 19 |
| 9. AUTHOR'S ADDRESS..... | 19 |
| 10. APPENDIX - DETAILED SCENARIOS..... | 20 |
| 10.1. Printer discovery within an enterprise..... | 20 |
| 10.2. Printer discovery across enterprises..... | 21 |
| 10.3. Printer discovery on the Internet -logical operations..... | 21 |
| 10.4. Printer discovery on the Internet - authentication..... | 22 |
| 10.5. Driver Download..... | 23 |
| 10.6. Submitting a print job as a file..... | 24 |
| 10.7. Submitting a print job with two documents..... | 24 |
| 10.8. Submitting a print job as a file, printing fails..... | 25 |
| 10.9. Submitting a print job with authentication, PRIVACY and payment..... | 26 |
| 10.10. Submitting a print job with decryption error..... | 27 |
| 10.11. Submitting a print job with authentication..... | 28 |
| 10.12. Submitting a print job generated dynamically..... | 29 |
| 10.13. Submitting a print job with a Printer jam - CANCELED..... | 29 |

| | |
|---|----|
| 10.14. Submitting a print job with a Printer jam - recovered..... | 30 |
| 10.15. Submitting a print job with server pull..... | 31 |
| 10.16. Submitting a print job with referenced resources..... | 32 |
| 10.17. Getting Capabilities..... | 33 |
| 10.17.1. Submission Attributes..... | 33 |
| 10.17.2. Printer Capabilities..... | 33 |
| 10.18. Getting Status..... | 34 |
| 10.18.1. Printer State/Status..... | 34 |
| 10.18.2. Job Status..... | 34 |
| 10.18.3. Status of All My Jobs..... | 34 |
| 10.19. Asynchronous Notification..... | 35 |
| 10.19.1. Job Completion..... | 35 |
| 10.19.2. Job Complete with Data..... | 35 |
| 10.19.3. Print Job Fails..... | 35 |
| 10.20. Cancel a job..... | 36 |
| 10.21. End to end Scenario - within an enterprise..... | 36 |
| 10.22. End to end Scenario - across enterprises..... | 37 |
| 10.23. End to End Scenario - on the internet..... | 40 |
| 11. Full Copyright Statement..... | 43 |

1. INTRODUCTION

The IPP 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.

2. TERMINOLOGY

Internet Printing for the purposes of this document is the application of Internet tools, programs, servers and networks to allow end-users to print to a remote printer using, after initial setup or configuration, the same methods, operations and paradigms as would be used for a locally attached or a local area network attached printer. This could include the use of HTTP servers and browsers and other applications for providing static, dynamic and interactive printer locating services, user installation, selection, configuration, print job submission, printer capability inquiry and status inquiry of remote printers and jobs.

For the purposes of this document, a WEB Browser is software available from a number of sources including but not limited to the following: Microsoft Internet Explorer, NCSA Mosaic, Netscape Navigator, Sun Hot Java!. The major task of these products is to use the Hypertext Transport Protocol (HTTP) to retrieve, interpret and display Hypertext Markup Language (HTML). These products are often a part of a complete Internet Printing system because they are often

used as a means of obtaining the status of or more information about the printing system; however, they may not be present in all implementations.

Throughout this document, 'printer' shall be interpreted to include any device which is capable of marking on a piece of media using any available technology. These design goals do not include support for multi-tiered printing solutions involving servers (single or multiple) logically in front of the actual printing device yet all such configurations shall be supported but shall appear to the end-user as only a single device.

Throughout this document 'driver' refers to the code installed in some client operating system to generate the print data stream for the intended printer. Some computing environments may not include a separate printer driver. Rather, the generation of the proper print data stream is accomplished in an application on that computer. How such a computer environment or application is updated to support a new printer now made available using IPP is outside the scope of IPP. The actual details for installing a printer driver are operating system dependent and are also outside the scope of IPP. See also section 4.1 (SECURITY CONSIDERATIONS) for security implications of driver download and installation.

The IPP protocol will support the following physical configurations:

- An IPP client talking to an IPP Printer object imbedded in a single, physical output device.
- An IPP Client talking to a server containing one or more IPP Printer objects. Each Printer object is associated with exactly one physical output device supported by the server. The protocol between the server and the output devices is undefined.
- An IPP Client talking to an IPP Printer object in a server. The Printer object is associated with one or more physical output devices, but the client only sees the Printer object, which is an abstraction and represents all of the associated physical output devices. The protocol between the server and the physical output devices is undefined.

Throughout this document, certain design goals will be identified as not being a part of version 1.0 (or V1.0) of the protocol or as being satisfied by means outside of IPP. IPP is assumed to be one part, an enabler, of a complete Internet Printing solution. For example printer instance creation is not performed by but is enabled by the protocol. Globally, none of the operator or administrators wants and needs are included in the design goals for version 1.0. Some of the end-user wants and needs may also be excluded from version 1.0 and will be so noted in the description of them. Subsequent versions of

the protocol (e.g. V2.0) may include support for these initially excluded wants and needs.

3. DESIGN GOALS

The next three sections identify the design goals for an Internet printing protocol from three roles assumed by humans: end-user, operator, and administrator. The goals defined here are only those that need to be addressed by an Internet printing protocol. Other wants and needs, such as that the operator needs physical access to the printer (e.g. to be able to load paper or clear jams) are not covered by this document. Section 5 contains scenarios which provide more detailed examples of the entire process including discovery, status, printing and end-of-job reporting.

3.1. END-USER

An end-user of a printer accepting jobs through the Internet is one of the roles in which humans act. The end-user is the person that will submit a job to be printed on the printer.

The wants and needs of the end-user are broken down into six categories: finding/locating a printer, creating a local instance of a printer, viewing printer status, viewing printer capabilities, submitting a print job, viewing print job status, altering the attributes of a print job.

3.1.1. Finding or locating a printer.

End-users want to be able to find and locate printers to which they are authorized to print. They want to be able to perform this function using a standard WEB browser or other application. Multiple criteria can be applied to find the printers needed. These criteria include but are not limited to:

- by name (Printer 1, Joes-color-printer, etc.)
- by geographic location (bldg 1, Kentucky, etc.)
- by capability or attribute (color, duplex, legal paper, etc.)

Additionally, while it is outside of scope of IPP, end-users want to be able to limit the scope of their searching to:

- inside a functional sub-domain
- include only a particular domain (lexmark.com)
- exclude specified domains

While an Internet printing protocol may not of itself include this function, IPP must define and enable a directory schema which will provide the necessary information for a directory service implementation to consistently represent printers by their IPP attributes.

3.1.2. Create an instance of the printer.

After finding the desired printer, an end-user needs to be able to create a local instance of that printer within the end-user operating system or desktop. This local instance will vary depending upon the printing paradigm of the operating system. For example, some UNIX users will only want a queue or a reference to a remote printer created on their machine while other UNIX users and Windows NT users will want the queue and also the necessary icons and registry entries to be created and initialized. Where required, drivers may need to be downloaded from some repository and installed on the computer. All necessary decompressing, unpacking, and other installation actions should occur without end-user interaction or intervention excepting initial approval by the end-user. Once the local instance of the printer has been installed, it shall appear to the end-user of the operating system and to the applications running there as any other printer (local, local area network connected, or network operating system connected) on the end-user desktop or environment. IPP's role in this goal is simply to enable the creation of the printer instance providing information such as where to locate a printer driver for this printer, as an attribute of an IPP Printer.

3.1.3. Viewing the status and capabilities of a printer.

Before using a selected printer or, in fact at any time, the end-user needs the ability to verify the characteristics and status of both printers and jobs queued for that printer. When checking the characteristics of a printer, the end-user typically wants to be able to determine the capability of the device, e.g.:

- supported media, commonly paper, by size and type
- paper handling capability, e.g. duplex, collating, finishing
- color capability

When checking the status of the printer and its print jobs, the end-user typically wants to be able to determine:

- is the printer on-line?
- what are the defaults to be used for printing?
- how many jobs are queued for the printer?
- how are job priorities assigned? (outside the scope of IPP)

3.1.4. Submitting a print job.

Once the desired printer has been located and installed, the end-user wants to print to that printer from normal applications using standard methods. These normal applications include such programs as word processors, spreadsheets, data-base applications, WEB browsers, production printing applications, etc. Additionally, the end-user may want to print a file already existing on the end-user's computer -- "simple push". In addition to printing from an application and simple push, the end-user needs to have the ability to submit a print job by reference. Printing by reference is defined to mean as submitting a job by providing a reference to an existing document. The reference, a URI, will be resolved before the actual print process occurs. Submitting a job by reference relieves the user from downloading the document from the remote server and then sending it via IPP to the printer. This saves both time and network bandwidth.

Some means shall be provided to determine if the format of a job matches the capability of the printer. This can be done by one of the following (all of which are outside of scope of the IPP protocol):

- the end-user selects the correct printer driver
- the printer automatically selects the proper interpreter
- the end-user uses some other manual procedure.

A standard action shall be defined should the job's requirements not match the capabilities of the printer.

Because the end-user does not want to know the details of the underlying printing process, the protocol must support job-to-printer capability matching (all implementations are not necessarily required to implement this function.) This matching capability requires knowing both the printer's capabilities and attributes and those capabilities and attributes required by the job. Actions taken when a print job requires capabilities or attributes that are not available on the printer vary and can include but are not limited to:

- rejecting the print job
- redirecting the print job to another printer (Not in V1.0)
- printing the job, accepting differences in the appearance

Print jobs will also be submitted by background or batch applications without human intervention.

End-users need the ability to set certain print job parameters at the time the job is submitted. These parameters include but are not limited to:

- number of copies
- single or two sided printing
- finishing
- job priority

3.1.5. Viewing the status of a submitted print job.

After a job has been submitted to a printer, the end-user needs a way to view the status of that job (i.e. job waiting, job printing, job done) and to determine where the job is in the print queue.

In addition to the need to inquire about the status of a print job, automatic notification of the completion of that job is also required.

Notification means are not defined by the protocol but the protocol must provide a means of enabling and disabling the notification.

3.1.6. Canceling a Print Job

While a job is waiting to be printed or has been started but not yet completed, the original creator/submitter of the print job (i.e. the end-user) shall be able to cancel the job entirely (job is waiting) or the remaining portion of it (job is printing.) Altering the print job itself is not a V1.0 design goal.

3.2. OPERATOR (NOT REQUIRED FOR V1.0)

An operator of a printer accepting jobs through the Internet is one of the roles in which humans act. The operator has the responsibility of monitoring the status of the printer as well as managing and controlling the jobs at the device. These responsibilities include but are not limited to the replenishing of supplies (ink, toner, paper, etc.), the clearing of minor errors (paper jams, etc.) and the re-prioritization of end-user jobs. Operator wants and needs will not be addressed by V1.0 of the protocol.

The wants and needs of the operator include all those of the end-user but may include additional privileges. For example, an operator may be able to view all print jobs on a printer while the end-user might only be able to see his own jobs.

3.2.1. Alerting.

One of the required operator functions is having the ability to discover or to be alerted to changes in the status of a printer particularly those changes that cause a printer to stop printing and

to be able to correct those problems. As such, an Internet printing protocol shall be able to alert a designated operator or operators to these conditions such as 'out of paper', 'out of ink', etc. Additionally, the operator shall be able to, asynchronous to other printer activity, inquire as to a printer's or a job's status.

3.2.2. Changing Print and Job Status.

Another of the required operator functions is the ability to affect changes to printer and job status remotely. For example, the operator will need to be able to re-prioritize or cancel any print jobs on a printer to which the operator has authority.

3.3. ADMINISTRATOR (NOT REQUIRED FOR V1.0)

An administrator of a printer accepting jobs through the Internet is one of the roles in which humans act. The administrator has the responsibility of creating the printer instances and controlling the authorization of other end-users and operators. Administrator wants and needs will not be addressed by V1.0 of the protocol.

The wants and needs of the administrator include all those of the end-user and, in some environments, some or all of those of the operator. Minimally, the administrator must also have the tools, programs, utilities and supporting protocols available to be able to:

- create an instance of a printer
- create, edit and maintain the list of authorized end-users
- create, edit and maintain the list of authorized operators
- create, edit and maintain the list of authorized administrators
- create, customize, change or otherwise alter the manner in which the status capabilities and other information about printers and jobs are presented
- create, customize, or change other printer or job features
- administrate billing or other charge-back mechanisms
- create sets of defaults
- create sets of capabilities

The administrator must have the capability to perform all the above tasks locally or remotely to the printer.

4. OBJECTIVES OF THE PROTOCOL

The protocol to be defined by an Internet printing working group will address the wants and needs of the end-user (V1.0). It will not, at least initially, address the operator or administrator wants and needs (V2.0).

The protocol defined shall be independent of the operating system of both the client and the server. Generally, any platform capable of supporting a WEB Browser should be capable of being a client. Generally, any platform providing a WEB/HTTP server and printing services should be capable of being a server. Usage of the WEB Browser and Server is not required for IPP; the operating system, operating system extensions or other applications may provide IPP functionality directly.

In many environments such as Windows 95, Windows NT and OS/2, the print data is created and transmitted to the printer on the fly rather than being created, spooled and then transmitted to the printer (a typical UNIX method.) The Internet Printing Protocol must properly handle either methodology and make this transparent to the end-user.

4.1. SECURITY CONSIDERATIONS

It is required that the Internet Printing Protocol be able to operate within a secure environment. Wherever reasonable, IPP ought to make use of existing security protocols and services. IPP will not invent new security features when the design goals described in this document can be met by existing protocols and services. Examples of such services include Secure Socket Layer Version 3 (SSL3) [SSL] and HTTP Digest Access Authentication [RFC2069]. Note: SSL3 is not on the IETF standards track.

Since we cannot anticipate the security levels or the specific threats that any given IPP print administrator may be concerned with, IPP must be capable of operating with different security mechanisms and policies as required by the individual installation. The initial security needs of IPP are derived from two primary considerations. First, the printing environments described in this document take into account that the client, the Printer, and the document to be printed may each exist in different security domains. When objects are in different security domains the design goals for authentication and message protection may be much stronger than when they are all in the same domain.

Secondly, the sensitivity and value of the content being printed will vary from one instance of a print job to another. For example, a publicly available document does not need the same level of protection as a payroll document does. Message protection design goals include data origin authentication, privacy, integrity, and non-repudiation.

In many environments (e.g. Windows, OS/2) a printer driver may be needed to create the proper datastream for printer. This document discusses downloading such a new driver from a variety of sources. Downloading and installing any software, including drivers) on a computer exposes that computer to a number of security risks including but not limited to:

- defective software
- malicious software (e.g. Trojan horses)
- inappropriate software (i.e. software doing something deemed unreasonable by the user.)

As such, proper security considerations and actions need to be taken by the user and/or a system administrator to prevent the compromising of the computer. Administrators should configure downloading mechanism for printer drivers in such a way as to be able to verify the source of driver software and encrypt or otherwise protect that software during download.

Examples including security considerations can be found in sections 5 (IPP SCENARIOS) and 10 (APPENDIX - DETAILED SCENARIOS) later in this document.

4.2. INTERACTION WITH LPD (RFC1179)

Many versions of UNIX and in fact other operating systems provide a means of printing as described in [RFC1179] (Line Printer Daemon Protocol.) This document describes the file formats for the control and data files as well as the messages used by the protocol. Because of the simplistic approach taken by this protocol, many manufacturers have include proprietary enhancements and extensions to 'lpd.' Because of this divergence and due to other design goals described in this document, there is no requirement for backward compatibility or interoperability with 'lpd'. However, a mapping of LPD functionality and IPP functionality shall be provided so as to enable a gateway between LPD and IPP.

4.3. EXTENSIBILITY

The Internet Printing Protocol shall be extensible by several means that facilitate interoperability and prevent implementation collisions:

- by providing a process whereby implementers can submit proposals for registration of new attributes and new enumerated values for existing attributes.

- * that require review and approval. The Internet Assigned Number Authority (IANA) will be the repository for such accepted registration proposals after review.

- * that do not require review and approval. IANA will be the repository for such registrations.

- by providing syntax in the protocol so that implementers may add private (i.e. unregistered) attributes and enumerated attribute values.

- by providing versioning and negotiation so as to enable future implementations of IPP to interoperate with implementations of version 1.0 of IPP.

4.4. FIREWALLS

As stated in section 3 Design Goals, Internet printing shall, by definition, support printing from one enterprise to another. As such, the Internet printing protocol must be capable of passing through firewalls and/or proxy servers (where enabled by the firewall administrator) preferably without modification to the existing firewall technology.

4.5. INTERNATIONALIZATION

Users of Internet printing will come from all over the world. As such, where appropriate, internationalization and localization will be enabled for the protocol.

5. IPP SCENARIOS

Each of the scenarios in this section describes a specific IPP operation, such as submitting a print job. Section 10 contains several detailed flows for each scenario to provide additional detail. The examples should not be considered exhaustive, but illustrative of the functions and features required in the protocol. Flows are intended to be protocol neutral. It is not assumed that all of the functions and features described in these scenarios will necessarily be supported directly by IPP or in version 1.0 of IPP.

See the IPP Model and Semantics document for details on configurations of clients, servers and firewalls.

5.1. PRINTER DISCOVERY

Client
Service

Directory Service

```
+----->
  give me information on printers with these characteristics
```

```
<-----+
  Information on Printers matching these characteristics
```

The objective of printer discovery is to locate printers that meet the client's wants and needs. The Directory Service should provide enough information for the client to make an initial choice. The client may have to connect to each individual Printer offered to get more detail. Not all information available from the Directory Service is obtained using IPP; some information may be administratively provided.

The actual protocol used between client and Directory or Name Service is considered outside the scope of IPP. Printer Discover is included in the scenarios to provide design goals for the directory schema for IPP Printers and to further define Printer attributes.

Characteristics that might be considered when locating a Printer include:

- capabilities of the Printer, e.g. PDLs supported
- physical location, e.g. in building 010
- driver required and location
- cost per page to print (outside the scope of IPP)
- whether or not printer is access controlled
- whether or not usage requires client authentication
- whether or not Printer can be authenticated
- whether or not payment is required for printing (outside the scope of IPP)
- maximum job size (spool size) (outside the scope of IPP)
- whether or not Printer support compression (outside the scope of IPP)
- whether or not Printer supports encryption
- administrative limits on this Printer
 - maximum number of copies per job
 - maximum number of pages per job

Responses could additionally include:

- how to get more information
 - web page
 - telephone number
 - help desk

5.2. DRIVER INSTALLATION

Client

Printer

```
+----->
  Where can I find a driver & software to install it?

<-----+
  URIs for drivers and install software
```

Driver here refers to the code installed in some client operating system to generate the print data stream for the intended printer. The actual details for installing a printer driver are operating system dependent and are also outside the scope of IPP. However, an IPP printer or a directory service advertising an IPP Printer should be capable of telling a client what drivers are available and/or required, where they can be found, and provide pointers to installation instructions, installation code or initialization strings required to install the driver. See section 4.1 (SECURITY CONSIDERATIONS) for security implications of driver download and installation.

5.3. SUBMITTING A PRINT JOB

Client

IPP Printer

```
+----->
  Here is a Print Job
    - Job attributes
    - Print data

<-----+
  Response
```

The protocol must support these sources of client data:

- Print data is a file submitted with the job
- Print data is generated on the fly by an application
- Print data is a file referenced by a URI

The protocol must handle overrun conditions in the printer and must support overlapped printing and downloading of the file in devices that are unable to spool files before printing them.

Every print request will have a response. Responses will indicate success or failure of the request and provide information on failures when they occur. Responses would include things like:

- Got the print job and queued it
- Got the print job and am printing it
- Got the print job, started to print it, but printing failed
 - why it failed (e.g. unrecoverable PostScript error)
 - state of the printer
 - how much printed
- Got the print job but couldn't print it
 - why it can't be printed
 - state of the printer
- Got the print job but don't know what to do with it
- Didn't get a complete print job (e.g. communication failure)

5.4. GETTING STATUS/CAPABILITIES

```

Client                                IPP Printer

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  Get status and/or capabilities of Printer

< -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  Status/Capabilities

```

Clients will need to get information about

- Static capabilities of the device
- Dynamic state of the Printer (e.g. out of paper)
- State of a specific job owned by this client
- State of all jobs owned by this client
 - queued
 - printing
 - completed

- Job submission attributes supported/required
 - scheduling attributes (e.g. priority)
 - production attributes (e.g. number of copies)

5.5. ASYNCHRONOUS NOTIFICATION

| | |
|---|-------------|
| Client | IPP Printer |
| +-----> | |
| Use the following method to notify me of Printer events | |
| . | |
| . | |
| . | |
| <-----+ | |
| Asynchronous notification of Printer event | |

Clients must be able to request asynchronous notification for Printer events such as

- job completion
- a fatal error that requires the job to be resubmitted
- a condition that severely impacts a queued job for this client
 - e.g. printer is out of paper

Note: end-user notification is a V1.0 design goal while operator notification is for V2.0.

5.6. JOB CANCELING

| | |
|-----------------------------------|-------------|
| Client | IPP Printer |
| +-----> | |
| Cancel the named job as indicated | |
| <-----+ | |
| Response (did it or not) | |

Similarly clients must be able to make changes to jobs which have been submitted and are queued for printing. Changing of job attributes should also be supported. Job modifications, holding and releasing of jobs are not included in the design goals for IPP v1.0.

6. SECURITY CONSIDERATIONS

The security considerations for IPP are described in Section 4.1 above.

7. REFERENCES

- [ipp-iig] Hastings, T. and C. Manros, "Internet Printing Protocol/1.0: Implementer's Guide", Work in Progress.
- [RFC2569] Herriot, R., Hastings, T., Jacobs, N. and J. Martin, "Mapping between LPD and IPP Protocols", RFC 2569, April 1999.
- [RFC2566] deBry, R., Hastings, T., Herriot, R., Isaacson, S. and P. Powell, "Internet Printing Protocol/1.0: Model and Semantics", RFC 2568, April 1999.
- [RFC2565] Herriot, R., Butler, S., Moore, P. and R. Tuner, "Internet Printing Protocol/1.0: Encoding and Transport", RFC 2565, April 1999.
- [RFC2568] Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol", RFC 2568, April 1999.
- [ISO10175] ISO/IEC 10175, Document Printing Application, June 1996.
- [RFC1179] McLaughlin, L., "Line Printer Daemon Protocol" RFC 1179, August 1990.
- [SSL] Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.

8. ACKNOWLEDGMENTS

This document draws heavily from preliminary work done by others especially in the Printer Working Group (PWG). The author gratefully acknowledges the specific contributions of:

Scott Isaacson
Novell
sisaacson@novell.com

Roger deBry
Utah Valley State College
debryro@uvsc.edu

Carl-Uno Manros
Xerox
manros@cpl0.es.xerox.com

Robert Herriot
Sun
Robert.Herrior@pahv.xerox.xom

Tom Hastings
Xerox
hastings@cpl0.es.xerox.com

Peter Zehler
Xerox
Peter.Zehler@usa.xerox.com

9. AUTHOR'S ADDRESS

F.D. (Don) Wright
Lexmark International
C14/035-3
740 New Circle Rd
Lexington, KY 40550

Phone: 606-232-4808
Fax: 606-232-6740
EMail: don@lexmark.com

10. APPENDIX - DETAILED SCENARIOS

The following are more detailed scenarios illustrating how the Internet Printing Protocol is expected to be used as a part of a complete Internet Printing system. Some parts of the scenarios include concepts, functions and information that may be outside of the scope of version 1.0 of IPP (e.g. cost per page, payments means available, etc.) The information contained herein is meant to be generic. There may not be an exact wording or terminology match between these scenarios and the implementation documents.

10.1. PRINTER DISCOVERY WITHIN AN ENTERPRISE

A user wants to find a color Postscript printer in his/her enterprise which will print transparencies. The client, directory service, and printer are all behind the same corporate firewall. Because color foils are expensive, printers of this type are access controlled and require an account to be established so that printing can be billed back to the using department. Note the request to find a printer usable by Dept. J15. Drivers for all supported printers are available from the server they are associated with. A help desk is provided for end user support. The printer is unattended.

Client

Directory Service

```

+-----+
Find a printer with these characteristics
- prints color, prints transparencies
- prints Postscript
- is in building 003
- accessible by the client

< -----+
Printer "Color-A"
- prints color, prints transparencies
- prints Postscript
- in room H-6, building 003
- driver ABC-Postscript-V1.3 required, here is URI
- cost is $.45 per page for color transparencies
- limit is 10 pages per job
- authentication required to use printer
- printer is unattended
- help desk at x5001

Printer "Color-B"
- prints color, prints transparencies
- prints Postscript
- in room J-10, building 003

```

- driver XYZ-Postscript-V2.4 required, here is URI
- cost is \$1.25 page for color transparencies
- limit is 5 pages per job
- authentication is required to use printer
- printer is unattended
- help desk at x5001

10.2. PRINTER DISCOVERY ACROSS ENTERPRISES

A user in Company A wants to find a public printer in a business partner's enterprise (Company B) on which to print a purchase order. The client is behind one corporate firewall and the directory service and the printer are behind a different corporate firewall. Drivers for all supported printers are available from the server they are associated with. A web page is provided for end user support for public printers.

Client

Company B Directory Service

```
+----->
Find a printer with these characteristics
- prints black and white
- is in El Segundo, building A
- is a public printer

<-----+
Printer "Public-A"
- prints black and white
- prints Postscript
- in El Segundo, room H-6, building A
- driver ABC-Postscript-V1.3 required, here is URI
- printer is public
- help available at http://xerox/elSegundo/publicPrinters

Printer "Public-B"
- prints black and white
- prints PCL/5e
- is in El Segundo, room J-10, building A
- driver XYZ-PCL-V2.4 required, here is URI
- printer is public
- help available at http://xerox/elSegundo/publicPrinters
```

10.3. PRINTER DISCOVERY ON THE INTERNET -LOGICAL OPERATIONS

A student wants to print a paper on a printer at his neighborhood Ink-o's print shop. The report was written using Microsoft Word. The student is interested in the cost of printing since his budget is limited. Note the use of logical operators to find this information.

Client

Ink-o's Directory Service

```
+----- >
Find a Printer with these characteristics
- prints color or black and white
- costs less than $.50 per page
- tell me about resolution and marking technology

< -----+
Printer "Color-A"
- prints color
- 600 dpi laser printer
- prints Postscript
- driver ABC-Postscript-V1.3 required, here is URI
- cost is $.50 per page for color
- payment required prior to submitting print job
- here is URI for more information on Ink-o's

Printer "Mono-B"
- prints black and white
- 300 dpi inkjet printer
- prints Postscript
- driver XYZ-Postscript-V2.4 required, here is URI
- cost is $0.35 page for black and white
- payment required prior to submitting print job
- here is URI for more information on Ink-o's
```

10.4. PRINTER DISCOVERY ON THE INTERNET - AUTHENTICATION

An executive in her hotel room is finishing an important presentation on her laptop computer. She connects to a local print shop through the web to get a copy of her charts printed for tomorrow's presentation. She must find a print shop that is convenient to her hotel and can print color transparencies. She wants to be sure that the printer can be authenticated and can accept encrypted data.

Client

SirZippy Directory Service

```
+----- >
Find a Printer with these characteristics
- prints color transparencies
- is in Boulder, Colorado
- Printer can be authenticated
- Printer supports encryption
```

Tell me when you are open for business

< -----+

Printer "Color-A"

- prints color transparencies
- prints Postscript
- driver ABC-Postscript-V1.3 required, here is URI
- payment required prior to submitting print job
- Printer can be authenticated
- Data can be encrypted
- Located at 1670 Pearl Street, Boulder, CO
- This Branch is open 24 hours a day

Printer "Color-B"

- prints color transparencies
- prints Postscript
- driver ABC-Postscript-V1.3 required, here is URI
- payment required prior to submitting print job
- Printer can be authenticated
- Data can be encrypted
- Located at 1220 Arapahoe, Boulder, CO
- This Branch is open from 9:00 am to 6:30 pm

10.5. DRIVER DOWNLOAD

An end user in an enterprise wants to print a lengthy report on a newly installed high speed PostScript printer. Since she will likely use this printer often, she would like to download a driver and install it on her workstation. She is running Windows 95. Note: Driver download is not a V1.0 design goal.

Client

IPP Printer

+----->

Tell me where to find print drivers for you

< -----+

Driver install file is at
<http://www.ibm.com/drivers/NP12a/Win95>

10.6. SUBMITTING A PRINT JOB AS A FILE

An end-user wants to submit a print job. The print file already exists on his workstation. The client and printer are behind the same corporate firewall. The printer is available to anyone behind the firewall and no authorization or authentication is required. The data is pushed to the printer. The printer is capable of spooling the output. No errors occur.

Client

IPP Printer

```
+----->
Here is a print job
- job name = MyJob
- notify me by email when done printing
- print on iso-a4-white paper
- print on both sides of the paper
- return status of the printer in response
- document is in Postscript format
- here is the document to print

<-----+
Print job accepted and spooled
- job id = #12345
- current state of print job = spooled
- submission time = 02/12/97, 15:35
- printer state = printing
```

10.7. SUBMITTING A PRINT JOB WITH TWO DOCUMENTS

An end-user wants to submit a print job. The print file already exists on his workstation. The client and printer are behind the same corporate firewall. The printer is available to anyone behind the firewall and no authorization or authentication is required. The data is pushed to the printer. The job consists of two separate documents. The printer is capable of spooling the output. No errors occur.

Client

IPP Printer

```
+----->
Here is a print job
- job name = MyJob
- notify me by email when done printing
- print on iso-a4-white paper
- print on both sides of the paper
- return status of the printer in response

<-----+
```



```

Print job accepted and spooled
- job id = #12345
- submission time = 02/12/97, 15:35
+----- >
- here is the document to print

< -----+
- OK

+----- >
- here is the document to print, it is the last document.

< -----+
- OK

```

10.8. SUBMITTING A PRINT JOB AS A FILE, PRINTING FAILS

An end-user wants to submit a print job. The print file already exists on his workstation. The client and printer are behind the same corporate firewall. The printer is available to anyone behind the firewall and no authorization or authentication is required. The data is pushed to the printer. The printer is not capable of spooling the output so it begins printing while still receiving the file. An error occurs and the printer cannot complete printing (in this case the user requires A4 paper and that paper size is not available on the printer.)

Client

IPP Printer

```

+----- >
Here is a print job
- job name = MyJob
- notify me by email when done printing
- print on iso-a4-white paper
- print on both sides of the paper
- return status of the printer in response
- document is in Postscript format
- here is the document to print

< -----+
Print job accepted

- printing failed
- current state of print job = canceled (A4 not available)
- submission time = 02/12/97, 15:35
- printer state = ready

```

10.9. SUBMITTING A PRINT JOB WITH AUTHENTICATION, PRIVACY AND PAYMENT

A traveling executive needs to print a set of transparencies for an important business meeting. The charts are in Lotus Freelance format on his notebook computer. He has located a SirZippy print shop near his hotel that will print color transparencies. Because the information on the charts is sensitive, he wants to be sure that his data is sent to the Printer in an encrypted format. He also wants to authenticate the Printer. The Printer also authenticates the user. Payment occurs across the Internet.

Client

IPP Printer

```
+----->
<-----+
```

Mutual authentication and exchange of secret keys

```
+----->
```

```
Here is a print job (encrypted)
- job name = MyJob
- notify me by email when done printing
- print on iso-a4-white paper
- print on both sides of the paper
- return status of the printer in response
- tell me where to pick up output
- document is in Postscript format
- here is the document to print
```

```
<-----+
```

```
Print job accepted and spooled (encrypted)
- job id = #12345
- current state of print job = spooled
- submission time = 02/12/97, 15:35
- printer state = printing
- payment required to proceed with job
- pick up at 230 East Main after 3:30 pm today
```

```
+----->
```

```
<-----+
```

Payment transaction

10.10. SUBMITTING A PRINT JOB WITH DECRYPTION ERROR

A traveling executive needs to print a set of transparencies for an important business meeting. The charts are in Lotus Freelance format on his notebook computer. He has located a SirZippy print shop near his hotel that will print color transparencies. Because the information on the charts is sensitive, he wants to be sure that his data is sent to the printer in an encrypted format. He also wants to authenticate the printer. The printer also authenticates the user. Payment occurs across the Internet. An error occurs during decryption.

Client

IPP Printer

```

+----- >
< -----+
  Mutual authentication and exchange of secret keys

+----- >
  Here is a print job (encrypted)
  - job name = MyJob
  - notify me by email when done printing
  - print on iso-a4-white paper
  - print on both sides of the paper
  - return status of the printer in response
  - tell me where to pick up output
  - document is in Postscript format
  - here is the document to print

< -----+
  Print job accepted and spooled (encrypted)
  - job id = #12345
  - current state of print job = spooled
  - submission time = 02/12/97, 15:35
  - printer state = printing
  - payment required to proceed with job
  - pick up at 230 East Main after 3:30 pm today

+----- >
< -----+
  Payment transaction
  .
  .
  .

< -----+
  Asynchronous response (email in this case)
  - decryption failed on job #12345

```

- no pages printed
- current state of job = aborted

10.11. SUBMITTING A PRINT JOB WITH AUTHENTICATION

An end-user wants to submit a print job. The print file already exists on his workstation. The client and printer are behind the same corporate firewall. The printer is available to anyone behind the firewall but authentication and authorization is required.

Authorization takes place using the authenticated end-user's name. The data is pushed to the printer. The printer is capable of spooling the output.

Client

IPP Printer

```
+----->
<-----+
Authentication
```

Note: An authentication failure would end the transaction at this point.

```
+----->
Here is a print job
- job name = MyJob
- notify me by email when done printing
- print on iso-a4-white paper
- print on both sides of the paper
- return status of the printer in response
- tell me where to pick up output
- document is in Postscript format
- here is the document to print

<-----+
Print job accepted and spooled
- job id = #12345
- current state of print job = spooled
- submission time = 02/12/97, 15:35
- printer state = printing
```

10.12. SUBMITTING A PRINT JOB GENERATED DYNAMICALLY

An end-user wants to submit a print job. The print data is generated dynamically and is being transmitted by a printer driver on the client workstation as available. The client and printer are behind the same corporate firewall. The printer is available to anyone behind the firewall and no authentication and authorization is required. The data is pushed to the printer. The printer is capable of spooling the output. No error occurs.

Client

IPP Printer

```
+----->
Here is a print job
- job name = MyJob
- notify me by email when done printing
- print on iso-a4-white paper
- print on both sides of the paper
- return status of the printer in response
- document is in Postscript format
- here is the print job

<-----+
Print data accepted and spooling started
- job id = #12345
- current job state = spooled
- submission time = 02/12/97, 15:35
- printer state = printing
```

10.13. SUBMITTING A PRINT JOB WITH A PRINTER JAM - CANCELED

An end-user wants to submit a print job. The print data is generated dynamically and is being transmitted by a printer driver on the client workstation as available. The client and printer are behind the same corporate firewall. The printer is available to anyone behind the firewall and no authentication and authorization is required. The data is pushed to the printer. The printer is not capable of spooling the output. The printer jams notifies the user and the user chooses to cancel the job.

Client

IPP Printer

```
+----->
Here is a print job
- job name = MyJob
- notify me by email when done printing
- print on iso-a4-white paper
- print on both sides of the paper
```

```

- return status of the printer in response
- document is in Postscript format
- here is the document to print

< -----+
  Print data accepted and printing started
  - job id = #12345

+----->
  - What is the status of print job #12345?

< -----+
  - Job #12345 accepted but printer jammed, cannot continue

+----->
  - Cancel job #12345

* Printer flushes remaining data
< -----+
  Print job terminated
  - current job state = canceled
  - submission time = 02/12/97, 15:35
  - printer state = jammed

```

10.14. SUBMITTING A PRINT JOB WITH A PRINTER JAM - RECOVERED

An end-user wants to submit a print job. The print data is generated dynamically and is being transmitted by a printer driver on the client workstation as available. The client and printer are behind the same corporate firewall. The printer is available to anyone behind the firewall and no authentication and authorization is required. The data is pushed to the printer. The printer is not capable of spooling the output. The printer jams, notifies the user and the user clears the jam and elects to continue.

Client

IPP Printer

```

+----->
  Here is a print job
  - job name = MyJob
  - notify me by email when done printing
  - print on iso-a4-white paper
  - print on both sides of the paper
  - return status of the printer in response
  - document is in Postscript format
  - here is the document to print

< -----+

```

```

Print data accepted and printing started
- job id = #12345

< ----- +
- Notification: printer jammed, cannot continue

* Jam is clear by human intervention, printing continues

+----- >
  Here is the last part of the document to print

< -----+
  Print job received
  - current job state = printing
  - submission time = 02/12/97, 15:35
  - printer state = printing

```

10.15. SUBMITTING A PRINT JOB WITH SERVER PULL

An end-user wants to submit a print job. The print data is in a file and is publicly available. It is pulled by the printer. The client and printer are behind the same corporate firewall. The printer is available to anyone behind the firewall and no authentication and authorization is required. The printer is capable of spooling the output. Printing may start before the entire job has been pulled.

Client

IPP Printer

```

+----- >
  Here is a print job

  - job name = MyJob
  - notify me by email when done printing
  - print on iso-a4-white paper
  - print on both sides of the paper
  - return status of the printer in response
  - here is a reference to the data to be printed

< -----+
  Print data accepted and printing started
  - job id = #12345
  - current state of job = spooled
  - submission time = 02/12/97, 13:15
  - printer state = printing

  .
  .
< -----+

```

Get the file to be printed

+----->

Here it is

Note: Failure to find the file, would end the transaction
with an error at this point and an asynchronous
notification would be send to the Client.

<-----+

Data received

10.16. SUBMITTING A PRINT JOB WITH REFERENCED RESOURCES

An end-user wants to submit a print job. Part of the print data is on a file on the user's workstation. It is pushed by the client, but the print job requires some resource not included in the print file. The client and printer are behind the same corporate firewall. The printer is available to anyone behind the firewall and no authentication and authorization is required. The printer is capable of spooling the output. No errors occur.

Client

IPP Printer

+----->

Here is a print job

- job name = MyJob
- notify me by email when done printing
- print on iso-a4-white paper
- print on both sides of the paper
- return status of the printer in response

<-----+

Print job accepted and spooled

- job id = #12345
- submission time = 02/12/97, 15:35

+----->

- here is the document to print

<-----+

- OK

+----->

- here is the URI to print, it is the last document.

<-----+

- OK


```

< -----+
  Get the external resource

+----->
  Here it is

```

10.17. GETTING CAPABILITIES

10.17.1. Submission Attributes

An end-user wants to get the production and scheduling attributes that are supported or required when submitting jobs to this printer. The client will use these attributes when forming the subsequent print request.

```

Client                                     IPP Printer
+----->
  I'm going to submit a Postscript job
  give me your job submission attributes

< -----+
  Postscript production attributes for this Printer are:
  - medium-select = us-letter-white, us-legal-white
    - default is us-letter-white
    - copies = 1,2,3,4,5
    - default is 1
      - print-quality = draft, normal, high
      - default is draft
      - sides = 1-sided, 2-sided-long-edge
      - default is 2-sided-long-edge
  - Job scheduling attributes for this Printer are:
    - job-priority = 1,2,3
    - default = 3

```

10.17.2. Printer Capabilities

An end-user wants to determine the resolution, marking technology, and PDLs supported by the printer.

```

Client                                     IPP Printer
+----->
  Please tell me the
  - resolution of the printer
  - the marking technology of the printer
  - PDLs supported

< -----+
  Printer resolution = 600 dpi
  Marking Technology = laser

```

PDLs supported = Postscript level 2, PCL/6

10.18. GETTING STATUS

10.18.1. Printer State/Status

An end-user wants to determine the state or status of the printer.

```
Client                                IPP Printer

+----->
  What is the state of the printer?

<-----+
  Printer state = out-of-paper
```

10.18.2. Job Status

An end user wants to get the status of a job he has submitted.

```
Client                                IPP Printer

+----->
  Please tell me the status of job #12345

<-----+
  Job #12345 is queued
  it is number 3 in the queue
  printer state = printing
```

10.18.3. Status of All My Jobs

An end user wants to get a list of all of the jobs he has submitted to this Printer.

```
Client                                IPP Printer

+----->
  Please tell me the status of my jobs

<-----+
  Job #00012 is complete
  Printed at 12:35 on 01/23/97

  Job #09876 is printing

  Job #12345 is queued
  it is number 3 in the queue
```

Job #34567 is queued
it is number 7 in the queue

10.19. ASYNCHRONOUS NOTIFICATION

10.19.1. Job Completion

An end-user wants to get notification of events that affect his print jobs. Print job completes without error.

Client

IPP Printer

```
< -----+
  Print job #123 completed
```

10.19.2. Job Complete with Data

An end-user wants to get notification of events that affect his print jobs. Print job completes, users asked for all end of job information.

Client

IPP Printer

```
< -----+
  Print job #123 completed
  - total pages printed = 15
  - number of copies printed = 3
  - total cost to print = $7.45
  - pick up copies in room H-6, building 005
```

10.19.3. Print Job Fails

An end-user wants to get notification of events that affect his print jobs. Print job fails. Printer is unattended.

Client

IPP Printer

```
< -----+
  Print job #123 failed
  - total pages printed = 15
  - number of pages submitted = 25
  - printer-state = jammed
```

10.20. CANCEL A JOB

The end-user submits a print job and later decides to cancel it.

```

Client                                IPP Printer

+----->
<-----+
  Authentication.

+----->
  Cancel job #1234

<-----+
  Job #1234 Canceled

```

10.21. END TO END SCENARIO - WITHIN AN ENTERPRISE

An office worker prints on shared departmental printers. All printers in the office are public, that is, no authentication or authorization is required. Printers are protected from external access by a firewall. No billing or accounting is required. Most printing is done from desktop applications. A help desk is provided for printing problems. Standard operating systems and applications are used. Drivers are available, but are installed manually by support personnel. This scenario assumes that drivers have been installed and that drivers are not IPP aware, that is, they cannot communicate across an IPP connection to obtain status and capabilities. IPP printers appear in application pull-down menus. Printer configuration data is hard wired into the driver.

End-user selects print from the application pull down menu. An IPP printer is selected from the list of Printers offered

The driver puts up a dialogue with hard-wired set of options for this printer. The end-user makes choices and submits job.

```

Client                                IPP Printer

+----->
  Here is a print job
  - job-name = memo-to-boss
  - notify me by email when job is complete
  - print on us-letter-white paper
  - print 1 copy
  - print at normal quality
  - print on 1 side

```

- give me the state of the printer in response

The driver generates the print data and passes it to the IPP driver a piece at a time as it is generated.

```
+----->
  Here is the print data
```

```
<-----+
  Print data received, file is spooled
  - printer state = printing
  - time submitted = 2/12/97, 15:35
  - current job state = spooled
```

Client adds this job to list of current jobs. List of jobs and state of each is available on a pull-down menu on the client.

End-user selects job #1234 from list and clicks on it to see its status.

```
+----->
  Give me the state of job #1234
  and the state of the Printer
```

```
<-----+
  Job #1234 state = spooled
  - it is number 3 in the queue
  - printer state = printing
```

The job completes without error

```
<-----+
  Job #1234 completed
  12 of 12 pages printed
```

10.22. END TO END SCENARIO - ACROSS ENTERPRISES

An office worker in Company A needs to print an office document on a "public" printer at Company B, a business partner. Both companies have corporate firewalls so the print request must flow out of A's firewall and into B's firewall. The office worker can look at public printers in Company B's directory service. The document is generated by a desktop application. Since the printer is "public" no authentication or authorization is required. A driver is downloaded. The driver is IPP aware, that is, it can communicate dynamically through the IPP protocol layer to obtain information about the printer.

Job scheduling attributes for this Printer are:

- job-priority = 1,2,3
 default = 3

Driver puts up dialogue with available options and fills in the defaults.

End-user makes choices and submits job

```
+----- >
Here is a print job
- job-name = memo-to-Don-Wright
- notify me by email when job is complete
- print on us-letter-white paper
- print 1 copy
- print at normal quality
- print on 1 side
- give me the state of the printer in response
```

The driver generates the print data and passes it to the IPP driver a piece at a time.

```
+----- >
Here is the print data

< -----+
Print data received, and spooling started
print job id = #1234

Print data received, file is spooled

- printer state = printing
- time submitted = 2/12/97, 15:35
- current job state = spooled
```

Client adds this job to list of current jobs. List of jobs and state of each is available on a pull-down menu on the client.

End-user selects job #1234 from list and clicks on it to see its status.

```
+----- >
Give me the state of job #1234
and the state of the Printer

< -----+
Job #1234 state = spooled
```

- it is number 3 in the queue
- printer state = printing

* The job completes without error

```
< -----+
Job #1234 completed
12 of 12 pages printed
```

10.23. END TO END SCENARIO - ON THE INTERNET

An executive in her hotel room is finishing an important presentation on her laptop computer. She connects to a local print shop through the web to get a copy of her charts printed for tomorrow's presentation. She must find a print shop that is convenient and can print color transparencies. She must download and temporarily install a driver in order to generate the PDL required by the print shop. Mutual authentication is required by the print shop and payment must be made in advance. The job is encrypted on the wire to prevent eavesdropping.

End-user completes presentation. She goes to the web and connects to the SirZippy home page.

```
Client                               SirZippy Directory Service
+-----+----->

Find me a printer with these characteristics
- Near Market Street in San Jose
- Prints color transparencies
- drivers can be downloaded
- supports privacy (encryption)
-
```

Available Printers matching these characteristics are looked up in the Directory Service

```
< -----+

Printer "Color-A"
- located at 123 First Street in San Jose
- URI is http://www.SirZippy.com/FirstStreet/Color-A
- prints color transparencies
- 600 dpi laser
- driver ABC-Postscript-V1.3 available at this URI
- cost = $.75 per page
- authentication required to use printer
- payment required prior to printing
```


Printer "Color-B"

- located at 67 San Carlos Street, San Jose
- URI is <http://www.SirZippy.com/SanCarlos/Color-B>
- prints color transparencies
- 1200 dpi laser
- driver XYZ-PostScript-V4.3 available at this URI
- cost = \$1.25 per page
- authentication required to use printer
- payment required prior to printing
- more information at this URI

The user decides to use the first printer because it is closer. She connects to the URI given to get a driver.

Client

Driver URI

```
+----- >
I need a driver for "Color-A"

< -----+
Driver installer is at http://www.xerox.com/prtdrvrs

Driver is installed

User connects to
"Color-A"
```

Client

IPP Printer "Color-A"

```
+----- >
< -----+
Mutual authentication and exchange of secret keys

+----- >
I'm going to submit a print job
give me your job submission attributes

< -----+
Production attributes for this Printer are:
- medium-select = us-letter-white, us-legal-white
  - default is us-letter-white
- copies = 1,2,3,4,5
  - default is 1
- print-quality = draft, normal, high
  - default is draft
- sides = 1-sided, 2-sided-long-edge
  - default is 2-sided-long-edge
```

Job scheduling attributes for this Printer are:

- job-priority = 1,2,3
 default = 3

Driver puts up dialogue with available options and fills in the defaults.

End-user makes choices and submits job

```
+----- >
  Here is a print job

  - job-name = presentation
  - notify me by email when job is complete
  - print on us-letter-transparency
  - print 1 copy
  - print at high quality
  - print by 9:00 am tomorrow morning
  - give me the state of the printer in response
```

The driver generates the print data and passes it to the IPP driver a piece at a time.

```
+----- >
  Here is the print data

< -----+
Print data received, and spooling started
print job id = #1234

Print data received, file is spooled
  - printer state = printing
  - time submitted = 2/12/97, 15:35
  - current job state = held, waiting for payment

+----- >
< -----+
  Payment transaction

< -----+
  Job is scheduled to print, pick up after 9:00am tomorrow
  Thank you for using SirZippy
```

11. Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

