INTERNET-DRAFT

P. Moore, B Jahromi
Microsoft Corporation
S. Butler
Hewlett-Packard Company

Version 1.1 June 6, 1997

Internet Printing Protocol - Level 1

Status of this Memo

1 Introduction

The Internet Printing Protocol is an industry effort to standardise printing operation in the Internet. This proposal describes a subset of that planned protocol that allows simple print job submission. Originally termed Simple Web Printing (SWP) is is now called IPP Level 1 (IPP1).

The purpose of IPP1 is to allow a user to submit, control and monitor print jobs whilst connected to a network via HTTP. This network may be the Internet, a corporate intranet or a mixture.

The overall aims in the design of SWP are:-

No changes to the HTTP protocols.

Provide the UI via HTML.

Not to compromise the security of the client nor the server.

Simplicity of implementation and use.

RATIONALE: The UI is presented via HTML so that as much of the existing client and server technology can be used. The alternative would have been to provide a management protocol that would have then necessitated a client-side UI.

RATIONALE: HTTP is used as the transport for job submission because it is certain that a web user will have an HTTP connection. Although HTTP implies the availability of a general-purpose TCP/IP transport, many Internet proxy servers only support the proxying of HTTP.

This project addresses many of the requirements identified by the IPP project of the IETF PWG [IPP]. That project is broader in scope, however. Where appropriate SWP has adopted the terminology, syntax and semantics of the IPP project. This has itself inherited much from the ISO 10175 DPA standard [ISO].

A large portion of SWP is left to the implementation of either the server or the configuration of a given implementation. This allows for the maximum flexibility in introducing this capability into rich web sites. It also allows for turnkey solutions to be developed.

NOTE: HTTP/1.1 is required in order to implement IPP1. The chunked data encoding of HTTP/1.1 is used to allow the transmission of data whose size is not known at the outset. In addition, persistent connections should be used whenever possible.

1.1 Terminology

User: The person wishing to submit print data for printing via IPP1.

Client: The sum of the Operating System and applications executing on the user’s computer.

Server: The sum of the operating system and applications executing on one or more computers that provide the service offered by SWP – i.e. where the printers are. Note that a physical printer could implement IPP1, in that case it would be the server.

Printer: The logical endpoint for data submitted via IPP1 – this may really be a printer. Alternatively it could be storage, an application, an email gatway, whatever is appropriate for a given implementation..

2 Job submission

The major part of IPP1 is the protocol that allows a client to submit a print job via HTTP. In keeping with its name this protocol is very simple:-

The print job is submitted in a single HTTP POST.

The entity part of the POST contains information about the job and the actual print data itself.

RATIONALE: Several possibilities exist for submitting the print job to the server:-

The model given above – the ‘atomic’ POST

A ‘start-job’ POST including the job attributes followed by one POST containing the print data

A ‘start-job’ POST including the job attributes followed by a set of print data POSTs. The job is terminated by an ‘end-job’ POST.

The ‘atomic’ POST is chosen because this leads to the simplest and most robust server implementation (important if the protocol is to be implmented in ‘real’ printers). It is also in keeping with the spirit of statelessness of HTTP.

NOTE: Only one print job may be submitted per POST request.

How this protocol is used or implemented on the client is not specified by IPP1. Several possibilities exist:-

The browser could be modified to perform the POST directly

A utility could be provided that performs the POST when given a file and a URL.

A print subsystem component could be produced that pipes the data from a user application to the server via the POST.

The last option is the one used by Microsoft and Hewlett Packard in their implementation. This in no way precludes others from providing one or more of the alternatives listed above.

RATIONALE: The most common print operation performed by users is to print using an application’s print option. This is processed via

the print provider / print driver / port monitor combination in Microsoft Windows

2.1 Obtaining the print URL

The server must provide URLs for the printers that it wishes to expose via IPP1. These URL can be supplied in one of two ways:-

Displayed on an HTML page – the user may then copy this and use it wherever the printer URL is requested by the client side code.

Automatically supplied as part of the client configuration – see later. This is the preferred option.

The server should provide both.

2.2 The data POST

The job attributes and the print data are sent as the entity part of a single HTTP POST using HTTP/1.1 chunked encoding if needed. The server must be able to accept both normal and chunked data.

2.3 HTTP Headers used

IPP1 does not mandate the use of any HTTP headers but there are some headers which have meaning to, or are associated with, IPP1.

2.3.1 Content-type

The data POST must have a content-type of “application/ippp”. This allows the server to verify that this is indeed a POST in IPP1 format.

2.3.2 Content-length

The content-length may be given if known in advance or HTTP/1.1 chunked encoding may be used.

2.4 Print by reference

Print by reference means rather than the client supplying the data to be printed it supplies a pointer (a URL for example) to the file and the server then ‘pulls’ the data from that location.

IPP1 print job submission does not support print by reference.

RATIONALE: Print by reference can be implemented in other ways on the server without any interaction with the client, other than the submission of the request. For example one could have an ISAPI extension on the server that accepted the URL of a file to be printed.

3 Server Responses

The normal HTTP/1.1 mechanisms are used to return results to the client from the server during a print POST operation.

The following responses are explicitly used by IPP1. Other responses may be used and have their normal meaning (‘401 Unauthorized’ for example)

3.1 200 OK

This means that the print request has been accepted by the server. The server must include an entity containing the URI that the client may use to access the job in the server’s queue. This URI points to a resource that can be browsed directly by a web browser – i.e. this is a web page that the user can open directly.

The URI is encoded in UTF8.

3.2 400 Bad Request

The print POST request has incorrect syntax. In particular this response is used when the server cannot parse the IPP1 headers.

3.3 403 Forbidden

The server has detected a syntactically correct POST for a resource that does not correspond to a printer.

3.4 413 Request entity too large

The server may implement a scheme to limit the size of print jobs that it will accept. If a print request exceeds those limits then this response must be returned.

3.5 415 Unsupported media Type

The IPP1 headers indicate that the print data is in a format not supported by the server.

4 Client Configuration

In order for a user to make use of IPP1 facilities it is expected that their client system will need to be configured in some way. This may include installing of executable components, print drivers, utilities, etc. as well as configuration data. This configuration data may include information about the printer to be used and that is needed by the client (installed options, memory size for example).

IPP1 does not specify what configuration needs to be done nor how it should be performed. This is specific to the client platform. The server should provide a mechanism for the client to be automatically configured, this is best done via URLs that point to client platform specific downloads.

There is a potential threat to the client from these downloads given that these may include executable code to be loaded via the Web. These downloads should therefore make use of whatever client protection schemes may be available (Microsoft's Authenticode for example). In the absence of these schemes it is hoped that the browser will at least alert the user to the potential threat.

4.1 Deinstalling

The implementation should provide a mechanism for removing the configuration performed by the above steps.

5 Monitoring and management

In this case monitoring and management means:-

Allowing the user to follow the progress of their print job in the server

Allowing the user to cancel their job, whilst queued and once delivery has started

IPP1 includes no specific features for monitoring or management of the submitted print jobs, printers or queues. Instead the server implementation should provide HTML forms and pages that allow these actions to be performed. These pages will need to be dynamically generated by whatever tools the HTTP server supports, ISAPI, ASP, CGI, etc.

The management pages should be easily discoverable; IPP1 provides no explicit support for these pages so it must be clear to the user where and how to reach them. The same page(s) that provided access to the client configuration and print submission should contain pointers to the relevant management pages.

6 Security

The following issues should be addressed by any implementation.

6.1 Server

The service provider (the organization running an instance of a particular IPP1 server implementation) will expect certain behaviors:-

6.1.1 Authentication

In order to control access to its resources the provider needs to know the identity of the end-user. This can also provide a mechanism for billing the user.

6.1.2 Access Control

To prevent malicious or accidental tampering with other print jobs the provider needs to be able to restrict access to only those jobs submitted by the current user.

It should be possible to control access to devices depending on the identification of the user. For example color printers may only be available to certain users.

6.1.3 Denial-of-service

The provider needs to have a mechanism for protecting itself from denial of service attacks. Example of this would be transmissions of large numbers of jobs, transmission of very large jobs, etc.

6.2 User

The user will have certain expectations for security:-

6.2.1 Privacy

The user should be able to request that the print job be transmitted in a secure manner.

6.2.2 Authentication

The user needs to be sure that IPP1 server is really who it claims to be.

6.2.3 Access Control

The user has the same expectation as the service provider that their jobs should be protected from other users whilst being processed.

6.3 Expected Implementation.

L1 does not define any protocols for security. The implementation must use the standard HTTP authentication methods to establish the identity of the client. The server must accept at least anonymous access and basic authentication. Any other schemes are allowed, this depends on the schemes supported by the client and / or server. The normal HTTP authentication scheme negotiation must be used.

6.3.1 Anonymous use

The anonymous case will probably be used for ‘Internet Fax’. In this case a company will announce on its web pages (or elsewhere) the URL of one or more public printers that anybody may send print jobs to.

There is a challenge for the server implementation in this case: prevention of denial-of-service attacks. The protection schemes used by commercial fax machines could be used, keeping lists of recent callers and only allowing a limited number of connections within a certain period for example.

The server may choose to not allow any management of the print jobs in this case given that there would be no control over which user could get at which jobs. Again, taking ‘real’ fax as the example, once a fax has been sent there is no way that it can be recalled.

6.3.2 Authenticated use

Authentication will be used in the case where a service bureau is making its printers available to paying customers, or where a business users wants to submit print to private printers on their corporate intranet.

Once the client identity is established then it is the responsibility of the server to maintain the association between the submitted job and the supplied user identification. It is also the responsibility of the server implementation to control access to the various server resources (printer, jobs, etc.) based on that user identification.

User identification can be used to protect against denial of service attacks. Given that anonymous printing will not, in general, be available then it is unlikely that a known user would do this.

The use of persistent HTTP connections is encouraged in order to avoid having the server challenge the user on every access to a resource.

Optionally, privacy and mutual authentication should be provided by SSL3 where this is available. The server implementation can control and / or announce its support of SSL3 by changing the URLs published on the page(s) that grant access to the printers. Two URLs could be provided for each device, one using SSL the other not, this would allow for clients that do not support SSL.

7 HTTP Entity format

The format of the entity sent in the job POST command is as follows:-

Protocol Version

Header-length

Operation

Attribute-1

Attribute-2

Attribute-3

Attribute-n

Print data

This is transmitted as a contiguous sequence of octets. There is no alignment and no padding.

All binary values in the header are transmitted in network byte order – MSB first.

All strings (both attribute names and value – see later) are in UTF8 format.

The encoding of the print data is not specified – it should be the native format of the PDL

7.1 Protocol Version

This is a 16 bit value giving the version number of the protocol encoing of this job. For version 1 it must be

X'0100'

7.2 Header-length

This is a count of the octets contained in the header. The header is defined as being all of the octets from the end of the header-length up to the start of the print data.

It is a 32 bit integer

The minimum value for header-length is therefore 0.

7.3 Operation

This specifies the operation to be perfromed by the server on this POST data.

L1 defines 2 operations

PrintJob – Print the job

Validate – No print data is sent,just verify access authorization and attributes.

Operation is encoded as an attribute called 'Operation' (see next section), it is distinguished by the fact that it must be the first attribute.

7.4 Attribute-n

Each Attribute is of the form:-

Name-Length Name Value-Length value

Where:-

Name-Length is a 16 bit integer giving the length of the name that follows

Name is a US-ASCII string identifying the Attribute. The string is case insensitive. The protocol permits the use of any characters, however it is recommended that specialbe avoided ('+','_',' ', etc.)

Length is a 16-bit binary value identifying the length of the value that follows. The Type and Length fields are not included in this length. The length specifies the number of octets that the value occupies, not its logical length. For example a text string may have 5 characters in it but occupy 10 octets (and hence have a length of 10) if the entity is using non ASCII characters.

Value is the value of the Attribute itself. The representation of the value being defined by the Attribute – see later

RATIONALE: This format is used because it makes it easy to parse and removes ambiguities that could arise from alternative schemes. For example a scheme using zero terminated strings may have problems with double-byte characters sets; a scheme based on CRLF termination is also ambiguous in those cases and in cases where there is a need to include CRLF in the value itself (a multi-line comment for example).

There may be 0 or more attributes (the operation pseudo-attribute must be present). No maximum limit is defined.

7.5 Print data

This is the print data itself. Unless otherwise indicated in the document-format attribute it is assumed by the server to be in a format that can be sent directly to the printer.

If the document-format is specified and is not directly compatible with the printer then the server must either convert the data into a form suitable for delivery or must raise an error (‘415 Unsupported Media Type’).

The print data must be absent in a Validate operation.

8 Attributes

The attributes used in IPP1 follow the model of [IPP], which is itself based on ISO 10175 [ISO].

[IPP] does not (at the time of writing) define any protocol encoding for these attributes. IPP1 therefore defines an encoding for them:-

Each attribute is identified by a unique name.

The respresentation of an attribute value depends on the type of the attribute.

‘SetOfX’ attributes (ones that have more than one value – example ‘finishing’) are represented by repeating the attribute triplet sequence for as many occurrences as required. The occurrences must be contiguous in the header. In the case where there is a default value then it must be the first value.

8.1 Attribute Value syntax

This section refers to [IPP] 5.1 for the names of value syntaxes. It provides a mapping between the syntax specified there and that used in IPP1.

8.1.1 Text

All attribute values specified as being composed of a sequence of characters ('Text', 'Name', 'fileName' and 'url') are represented by characters in UTF8 format. The value-length field of the attribute must contain the count of the number of octets used to store the text (i.e this may be greater than the number of characters in the case where non ASCII characters are used)

8.1.2 Octet String

A sequence of octets

8.1.3 Boolean

Shall be represented by one octet. X'00' meaning 'false' and X'FF' meaning 'true'.

8.1.4 Integer

Shall be represented by a 32-bit signed integer . Includes IntegerSeconds, IntegerPoints, IntegerUnits.

8.1.5 DateTime

Shall be encoded in RFC1123 format (e.g. 'Mon, 28 Apr 1997 09:00:00).

8.1.6 Keywords

Shall be represented by a 16-bit unsigned integer. Each set of allowable value for a keyword shall be assigned its own enumeration. This is specified for each Attribute that takes a keyword value. Where [ISO] defines a set of values for an attribute ('medium' for example) then the OIDs specified are used.

8.2 Attributes

All job and document attributes defined in [IPP] may be included in the IPP1 header.

A server may ignore those attributes that it does not support.

There are no mandatory attributes.

The attributes currently used by IPP1 are specified below.

8.2.1 Job-name

Name-Length : 8

Name : Job-Name

Syntax: Text (1 to 4096 characters)

Meaning: A client assigned name for the job. NFS for the server – treated as a comment

8.2.2 Job-originator

Name-Length : 14

Name : Job-Originator

Syntax: Name (1 to 255 characters)

Meaning: A client assigned user name. NFS for the server – treated as a comment. NOTE: The user identification needed for access control must be obtained from the HTTP authentication negotiation, not from this attribute.

8.2.3 Document-format

Name-Length : 15

Name : Document-Format

Syntax: Keyword – values taken from [RFC 1759]

Meaning: Specifies the type of data included; PCL, PostScript, etc.

8.2.4 Operation

Name-Length : 9

Name : Operation

Syntax : Keyword

PrintJob = 1

Validate = 2

8.2.5 Examples

Note that the hexadecimal values are shown in the network octet order. Also note that there is no break between the header, the attributes and the print data, they are transmitted a continuous sequence of octets. The line breaks are included for readability only.

Strings are shown as characters to improve readability, it should be understood that ‘a’ means hex 21

A print job with no attributes given:-

```
01 00                          ; version 1.0
00 00 00 0F                    ; length = 15
00 09 O p e r a t i o n 00 02 00 01
<print data>
```

A print job including name (“job”) and document-format of PCL:-

```
01 00
00 00 00 34                                  ; Header length = 52
00 09 O p e r a t i o n 00 02 00 01
00 0F D o c u m e n t – f o r m a t 00 02 00 03       ; PCL
00 08 J o b – n a m e 00 03 j o b
<Print Data>
```

9 Internationalization

This topic covers several aspects.

9.1 Coding of the IPP1 header.

The IPP header is in UTF8 format. This allows for any UNICODE characters.

9.2 Encoding of the print data.

The character set(s) used in the print data itself is not specified by IPP1.

9.3 Service HTML pages.

The HTML pages used to access and manage the service should be in the native language of the client and should therefore also use the appropriate character set.

The service implementation should use the normal HTTP/1.1 methods for discovering the client’s requirements (Accept-Charset, Accept-Language) and should honor them.

10 Location of server providing IPP1

IPP1 does not cover the location of service providers on the Web. The most likely scenario is that providers will announce this on their web pages and that standard web search engines will therefore be able to locate them.

11 Example of use

A user wishes to submit a print job to a local print shop that is on the Web and offers IPP1 services. The user already has an account with the shop. The user is using Microsoft Windows and Microsoft Word

The user accesses their home page www.pshop.com

On the home page there is a link to ‘online printing service’. The user clicks that link.

That leads to a page that asks him to select a printer. There are three shown: B&W, color and transparency along with the rates for each. The web server has used the HTTP

headers to identify the client platform and so offers links only to the relevant platform specific configuration downloads.

The user clicks on the color printer link. This initiates a client configuration. This transfers the necessary print drivers and configuration information (including the printer URL). This printer now appears as a normal Windows printer

The user now opens Word, selects the print shop's color printer in the print dialog and then starts the print.

The print provider spools the data to the web server.

The web page that offered links to the printers also has links that allow you to view the queue for that printer. The user clicks on the link for the color printer's queue and sees his job spooling in and then being printed.

12 References

[IPP] deBry R., Hastings T., Herriot B., Isaacson S., "Internet Printing Protocol/1.0 Model and Semantics", IBM, Xerox, Sun, Novell, March 1997

[RFC2068] Fielding R., Gettys J., Mogul J., Frystyk H., Berners-Lee T., "Hypertext Transfer Protocol - HTTP 1.1", UC-Irvine, DEC, MIT/LCS, January 1997

[RFC2069] Frank J., Hallam-Baker P., Hostetler J., Leach P.,Luotonen A., Sink E., Stewart L., "An Extension to HTTP – Digest Access Authentication", Northwestern Univ, CERN, Spyglass, Microsoft, Netscape, Open Market, January 1997

[ISO] ISO/IEC 10175 Document Printing Application (DPA), Final, June 1996

[RFC1759] Smith, R. Wright, F. Hastings, T,Zilles, S., and Gyllenskog, J. "Printer MIB" RFC 1759, March 1995

___