Roger deBry
IBM Corporation
T. Hastings
Xerox Corporation
R. Herriot
Sun Microsystems
Scott Isaacson
Novell, Inc.

November 1996

Internet Printing Protocol - IPP/1.0
draft-isaacson-ipp-info-00.txt
Expires May 27, 1997

Status of this Memo

Abstract

This Internet-Draft specifies an Internet Printing Protocol (IPP)that is intended to be version 1.0. This protocol is heavily influence by the semantic operations and attributes defined in ISO/IEC 10175 Document Printing Application (DPA) parts 1 and 3. It also incorporates some of the implementation and interoperability lessons learned from other printing related standards such as POSIX System Administration - Part 4 (POSIX 1378.4) and X/Open A Printing System Interoperability Specification(PSIS).

IPP is defined as a set of abstract data types and operations. The operations are implemented using a simple request and response mechanism built on top of HTTP. The abstract data types are encoded as simple ASCII text strings.

The IPP protocol covers only end user operations on basic print service objects. Authentication is realized by mechanisms outside the scope of the protocol, but the protocol does introduce some access control functionality so that only authorized end users are allowed to submit print jobs to printers whose implementation and site policy support access control. Also, the Cancel Job operation requires some authentication so that jobs can only be canceled by the end user who submitted the job. Extended monitoring and management is possible through other protocols such as the SNMP Printer MIB. In the areas where there are no existing standards, some proposed and emerging standards are being worked (management, security, etc.). As these services become more stable, this document (and hence the protocol) can be updated to reflect the integration and relationships with these other standards.

Table of Contents

1. Introduction

The Internet Printing Protocol (IPP) is an application level protocol that can be used for distributed printing on the Internet. The protocol is heavily influenced by the printing model introduced in the Document Printing Application (ISO/IEC 10175 DPA) standard, which describes a distributed printing service. DPA identifies the end user and administrative roles associated with a distributed printing service, and defines the set of operations supported by the service. This IPP specification (version 1.0) deals only with the end user role. These ideas and concepts, when unified with other Internet protocols and services, realize a distributed print service for the Internet.

This specification uses the verbs: "shall", "should", "may", and "need not" to specify conformance requirements as follows:

- "shall": indicates an action that the subject of the sentence must implement in order to claim conformance to this specification

- "may": indicates an action that the subject of the sentence does not have to implement in order to claim conformance to this specification, in other words that action is an implementation option

- "need not": indicates an action that the subject of the sentence does not have to implement in order to claim conformance to this specification. The verb "need not" is used instead of "may not", since "may not" sounds like a prohibition.

- "should": indicates an action that is recommended for the subject of the sentence to implement, but is not required, in order to claim conformance to this specification.

2. Distributed Printing

This document assumes a distributed computing environment where requesters of print services (clients, applications, PC drivers, etc.) cooperate and interact with print service providers. Although the underlying configuration may be a complex n-tier client/server system, an important simplifying step in this protocol is that the only object the requester of the print service ever sees is a "printer". It is important, however, to understand that in a real system, other components of a print service exist.

2.1 Generic Print System Components

Every distributed print service, including those using the Internet Printing Protocol, includes elements from the following list.

- End Users: End Users are humans (or agents or applications who work on behalf of a human) who submit print jobs.

- Print clients: Print clients are computer network nodes with which humans interact in order to manipulate the distributed print service. A print client uses some protocol to invoke print service operations on another node. Each operation has arguments and results associated with it. The print client provides arguments which add information about the operation requested, and receives results which describe the status and outcome of the operation.

- Print servers: Print servers may be embedded in an output device or implemented in a separate system which is associated with an output device. The print server receives requests from the print client and sends back results which describe the status and outcome of the operation requested. A print server normally provides queuing, job management, and device management functions.

- Queues: Print jobs may be queued or stored on a spool prior to printing. This allows a print service provider to accept one or more print jobs while the printer (or printers) is busy processing another job. Queues, if present, may be implemented in the client, in the server, in the output device, or in some combination of the three.

- Output Devices: Output devices interpret the print data and generate some form of output. In the case of a laser printer, for example, this normally means rasterizing the print data and putting the resulting marks on paper. An output device may receive print data directly from a client or through a Print server.

A specific implementation of a print service may not include all of the elements described here, and the physical packaging of elements is up to the implementation. For example, an output device may include a queue or a print server may include a rasterizer.

2.2 IPP Components

The print model defined by the Internet Printing Protocol simplifies the user's view of the system components described in the previous section by encapsulating the important elements of the system into five simple objects:

- End Users (no specific object definition via attributes)
- Clients (no specific object definition via attributes)
- Printers (section 6.4)
- Print Jobs (section 6.2)
- Job Templates (section 6.5)

Clients use the following operations:

- Print (section 5.4.1)
- Cancel Job (section 5.4.2)
- Get Attributes (section 5.4.3)
- Get Jobs (section 5.4.4)

3. IPP Objects

This section describes the IPP objects.

3.1 Printer

One of the most significant objects in the IPP model is the Printer. To the end user, the Printer object represents the functionality of the actual output device along with the queuing, job management, and device management functions often associated with a print server. An IPP Printer object implements the Internet Printing Protocol. Using the protocol, end users may query the attributes of the Printer, submit jobs to the Printer, determine subsequent states of submitted and queued jobs and state of the Printer, and cancel their own print jobs. The realization of a Printer object may take on different forms for any given configuration of real components. However, the details of the configuration of real components must be transparent to the end user.

In addition, a Printer is an abstraction for any document Output Device. This means that a Printer could be used to represent any real or virtual device which can support the Printer operations and interfaces. For example, a Printer could be used to front end a fax-out device, any kind of imager, or even a CD writer.

Some examples of configurations containing IPP Printer object include:

- An output device, with no spooling capabilities, supporting IPP
- An output device, with a built-in spooler, supporting IPP
-
-
- A print server with one or more associated output devices with the print server supporting IPP.
    - The associated output devices may or may not be capable of spooling jobs
    - The associated output devices may or may not support IPP

See the following figures for some examples on how to view IPP Printer objects on top of other printing system models:

```
Legend:

##### indicates an IPP Printer object which is
      either embedded in an output device or is
      hosted in a server.  An IPP Printer object
      may or may not queue/spool.

any   indicates any network protocol or direct
      connect, including IPP


embedded printer:
                                          output device
                                        +---------------+
 O   +--------+                         |  ###########  |
/|\  | client |------------IPP------------># Printer #  |
/ \  +--------+                         |  ###########  |
                                        +---------------+


hosted printer:
                                        +---------------+
 O   +--------+        ###########      |               |
/|\  | client |--IPP--># Printer #-any->| output device |
/ \  +--------+        ###########      |               |
                                        +---------------+



                                        +---------------+
fan out:                                |               |
                                    +-->| output device |
                                any/    |               |
 O   +--------+      ###########  /     +---------------+
/|\  | client |-IPP-># Printer #-*
/ \  +--------+      ###########  \     +---------------+
                                any\    |               |
                                    +-->| output device |
                                        |               |
                                        +---------------+
```

3.2 Job

A Job object is used to model a job. A job can contain one or more documents. However, there are no separate document objects. The impact of this is that there are no attributes that pertain to one document in a job but not to others, except for a single attribute that specifies the document data, its location, and its format. Note: In future versions, documents may become separate objects with attributes whose scope and application are different from the corresponding job attributes.

Job attributes are broken up into the following groups:

- Job Informational (sections 6.2.1, 6.2.2)
- Job Status (section 6.2.3)
- Job Sheet (section 6.2.4)
- Notification (section 6.2.5)
- Job Scheduling (section 6.2.6)
- Job Production (section 6.2.7)
- Conversion of Text Files (section 6.2.8)
- Job Resources (section 6.2.9)
- Number of Documents (section 6.2.10)
- Document Attributes (6.2.11)

3.3 Job Template

A Job Template object is used to model job defaults. A Job Template is essentially a set of job attributes that initialize a newly created job object.

Issue: The notion of Job Template needs more work.

3.4 Object Relationships

Instances of objects within the system have relationships which must be maintained persistently along with the persistent storage of the objects themselves. A Printer can contain zero or more Job objects. Therefore, a job object is contained in exactly one Printer object. A Job object contains one or more Documents.

A Printer object is associated with zero or more Job Template objects.

3.5 Object Identity

All instances of all objects have an identifier attribute that makes them unique so that they can be unambiguously referenced.

The following objects have the following mandatory identifier attributes:

Object	Identifier	Containing Object
Printer	printer-name	None
Job	job-identifier	Printer
Job Template	job-template-name	None

4. Naming

Clients identify Printer objects by using an HTTP type URL. For example, a URL for a Printer object named "printer-1" whose network node's domain name is "some.domain.com", might look like:

http://some.domain.com/printer-1

In this case, the URL identifies the use of the HTTP protocol. The Printer is located at the node identified by the DNS name "some.domain.com" and "printer-1" is the name of the Printer.

Another example is the following URL:

http://1.2.3.4:nnn/printer-2

In this case, the URL identifies the use of the HTTP protocol. The Printer is located at the node identified by the IP address of "1.2.3.4" using port nnn for the HTTP server, and "printer-2" is the name of the Printer. (The actual value of nnn is to be assigned by IANA as part of this standards project).

It is not necessary to expose the Job Template objects that might be associated with a given printer as separate objects. They can be exposed in two ways through URL naming.

- The Job Template can be hidden from the end user by a URL that represents just the Job Template name (but does not expose the Printer object name) as the two URLS

    1) http://some.domain.com/two-sided-printer, and
    2) http://some.domain.com/draft-printer.

  These look like two different Printers , but underneath they represent the same Printer object, but that Printer object has two associated Job Templates and each is exposed through a different URL for the same Printer object. Each one of the Job Templates specified by a URL would contain a different Job Template default attribute set. One Job Template would contain the defaults for two-sides printing and the other would contain the defaults for draft printing.

- The Job Template can be exposed along with the name of the Printer object directly in the URL as in:

    1) http://some.domain.com/hr-printer/resumes
    2) http://some.domain.com/hr-printer/1040forms

  In this case there are "resumes" and "1040forms" Job Templates associated with the "hr-printer" Printer.

This specification establishes, through IANA, a new well known port, port nnn, for the use of IPP over HTTP. The purpose of this new well known port would be to distinguish printing from non-printing content. While any acceptable HTTP content could be inter-mixed over HTTP well known port 80, only IPP printing would be acceptable on port nnn.

4.1 Directory Services

IPP does not require any specific directory service. However, this specification does define a generic schema that can be used for any specific instance of a directory service. That is, some of the attributes from the Printer object are called out as attributes that may be added to a directory entry which represents that Printer. This allows directory users to find and locate IPP Printers by either a simple name look up or by some filtered attribute search.

4.2 Directory Entry Schema

The following attributes define the generic directory entry schema. All directories entries for IPP Printers in all types of directories should support at least these attributes.

Issue: The use of "objective" attributes vs. "subjective" attributes still needs to be resolved. For example, for Maximum Print Quality is it better to have values like "high", "medium", "low" or to have explicit, quantified, measurable values? Some of the issues are: end users don't often know what explicit objective values are or what they really mean and they want to depend on an administrator to define what is "high" quality printing and what is "low" quality, especially since today's objective values that equate to "high" are tomorrow's objective values that equate to "medium". On the other hand, some end users demand the control and power explicit values can give them when they do filtered searching. For example, they know and appreciate the difference between 20 ppm printers and 23 ppm printers.

Issue: We must specify which attributes are "mandatory" and which are "optional". LDAP uses the terms "must" and "may" to identify attributes that "must" appear and attributes that "may" appear in a given entry in the directory.

4.2.1 Name

This directory attribute is the printers name. It is a URL so it contains sufficient information to not only name, but to address the printer using IPP as well.

4.2.2 Description

This directory attribute is a free form string that can contain any site-specific descriptive information about this printer.

4.2.3 Location

This directory attribute is a free form string that can contain any site specific location information.

In order for filtered searches to be more effective, a given site may use some regular structuring within the string values such as "SITE:USA-San Jose,BUILDING:A1,FLOOR:2,ROOM:555" or "department5-2ndFloor-A5-IndianHills-Chicago-IL-USA".

4.2.4 Maximum Print Quality

This directory attribute indicates a somewhat subjective evaluation of the overall printing quality. The syntax and values shall be the same as for the print-quality Job attribute.

4.2.5 Cost

This directory attribute indicates a somewhat subjective evaluation of the overall cost of printing at this printer: "high", "medium", or "low".

4.2.6 Resolution

This directory attribute is the maximum resolution of the Printer in dpi.

The syntax and semantics shall be the same as for the printer-resolution-select job attribute.

4.2.7 Color Supported

This directory attribute specifies whether the Printer supports color and, if so, what type. The values are a type2Enum (see section 6). Standard values are: "none", "highlight", "three color (CMY)", "four color (CMYK)", "monochromatic".

4.2.8 Fonts Supported

This directory attribute takes on a list of fonts that are supported by the printer. The syntax and values shall be the same as for the fonts-used job attribute..

4.2.9 Maximum Speed

This directory attribute is the maximum speed of the printer ppm, ipm, spm, lpm, or cps. The syntax and values shall be the same as for the maximum-printer-speed Printer attribute.

4.2.10 Device Id

This directory attribute can be used for automatic driver download, database access, or other automatic configuration tasks. It might be used to generate a platform specific id such as the Windows Plug-and-Play id.

Issue: Is this the IEEE 1284-1994 device id, the Object Identifier as used in the Host Resource MIB hrDeviceId object, or some other identifier?

4.2.11 Make and Model

This directory attribute is a simple text string defined by the manufacturer that contains some reference to the make and model of the entity being represented to the end-user by this Printer object. The syntax shall be:

```
vendor-name "/" model-name
```

where the vendor-name is the same as that registered with IANA for use in domain names.

For example: "vendor-x/super-duper-printer".

4.2.12 Marker Type

This directory attribute is the printing mechanism of the print device: electrophotographic-laser, inkjet-aqueous, thermal-transfer, etc. The syntax and values shall be the same as for the printer-types Printer attribute, except the value of the Marker Type directory attribute shall be single-valued

4.2.13 Document Formats Supported

This directory attribute is a list of all of the document formats that the printer and/or its interpreter(s) support. The syntax and values shall be the same as for the document-format Job attribute.

4.2.14 Sides Supported

This directory attribute specifies the capabilities of the Printer for marking on sides of the medium. The syntax and values shall be the same as the sides Job attribute.

4.2.15 Finishings Supported

This directory attribute identifies the finishing operations supported by the Printer. The syntax and values shall be the same as the finishing job attribute.

4.3 Directory Entries Using LDAP

To allow directory users to locate an IPP Printer, a corresponding entry must be defined within a directory. This section describes how this is done using the Lightweight Directory Access Protocol (LDAP).

The LDAP directory entry includes the name of the entry and the attributes as defined in "4.2 Directory Entry Schema". The following is an example of how to define a directory entry for a Printer object using LDAP. It is given to assist the reader's understanding of this specification.

To create a Printer object directory entry using LDAP:

1. An administrator uses a program to create an entry for the Printer object on a directory server that supports LDAP. The administrator defines the Distinguished Name (dn) and the default subjective attributes for the Printer object directory entry.

Issue: Should the administrator also define default objective attributes or wait for the Printer object itself to initialize these attributes?

2. The Printer object invokes the ldap_open API to open a connection to the directory server:

Example: ld=ldap_open ("dir.host.name", LDAP_PORT)

where ld is the connection handle for subsequent LDAP APIs.

3. The Printer object invokes an ldap "bind" API to authenticate with the directory server.

Example: ldap_simple_bind_s (ld, dn, NULL) (which does a simple authentication without a password).

4. The Printer object invokes the ldap_modify or ldap_modify_s API to define the objective attributes for the Printer object entry as identified by its Distinguished Name (dn).

Example: ldap_modify_s (ld, dn, mods) (where mods is a NULL-terminated array of objective attributes and values to add or modify in the directory entry)

5. The Printer object invokes the ldap_unbind API to close the connection to the directory server.

Example: ldap_unbind (ld)

When one or more objective attributes are modified for a Printer object, the Printer object repeats steps 2-5 to update the modified objective attributes in its directory entry.

To locate a Printer object entry using LDAP, a program can use the ldap_search or ldap_search APIs or a user can specify an LDAP URL.

For example, to locate all Printer objects that support duplex, a user can specify URL:

```
ldap:///dir.host.name???(&(objectClass=printer)
   (sides-supported=2-sided-long-edge))
```

Issue: Is it allowed to filter the search based on the object class itself, in this case the object class of Printer? We need to define this new object class. How do we do this? One proposal is to subclass the device class defined in X.500:

```
printer OBJECT-CLASS ::= {
   SUBCLASS OF {device}
   MUST CONTAIN {<list of mandatory attributes>}
   MAY CONTAIN {<list of optional attributes>}
```

5. IPP Operations

This section introduces the IPP operations. Since IPP specifies the use of HTTP as the underlying communication protocol, the mapping of IPP operations on top of HTTP methods is also shown.

5.1 HTTP Overview

IPP is based on the existing HTTP standard. IPP is a lightweight application-level protocol designed with the Internet in mind. It is a generic, stateless, object-oriented protocol which can be used for any task through extension of its request methods (commands).

HTTP allows an open-ended set of methods to be used to indicate the purpose of a request. It builds on the discipline of reference provided by the Uniform Resource Location (URL) and message formats similar to those used by Internet Mail and the Multipurpose Internet Mail Extensions (MIME).

HTTP is based on a request-response paradigm. A requesting program (a client) establishes a connection with a receiving program (a server) and sends a request to the server in the form of a request method, a URL, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possibly print data. The server responds with a status line, including its protocol version, and a success or failure code, followed by a MIME-like message containing server information, entity meta-information, and possibly some content.

Current practice requires that the connection be established by the client prior to each request and closed by the server after sending the response. Both clients and servers shall be capable of handling cases where either party closes the connection prematurely, due to user action, automated time out, or program failure.

5.2 IPP Operation Encoding

IPP messages consist of requests from client to server and responses from server to client.

```
   IPP MESSAGE = Request | Response
```

Requests and responses use the generic message format of RFC 822 for transferring entities. Both messages may include optional header fields and an entity body. The entity body is separated from the headers by a null line (a line with nothing preceding the CRLF).

```
   Request = Request-line
          * (General-Header
                |  Request-Header
                |  Entity-Header)
          CRLF
          [ Entity-Body ]

   Response = Status-line
          * (General-Header
                |  Request-Header
                |  Entity-Header)
          CRLF
          [ Entity-Body ]
```

All IPP headers conform to the syntax

```
   IPP-Header = field-name ":" [field-value] CRLF.
```

IPP/1.0 defines the octet sequence CRLF as the end-of-line marker for all protocol elements except the entity-body.

Note that HTTP 1.1 defines a slightly different syntax, allowing for dynamically generated messages to be transmitted. This would be required for cases such as PC driver generated Print Operations. HTTP 1.1 defines a message header which specifies a transfer encoding called "chunks".

IPP messages are contained within HTTP methods. The HTTP POST method is used for the Print operation and the Cancel Job operation. The HTTP GET method is used for the Get Attributes operation and the Get Jobs operation (section 5.4).

5.2.1 HTTP Request-Header Fields

HTTP request header fields allow the client to pass additional information about the request, and about the client itself, to the server. All header fields are optional and when used it is assumed that IPP would use these headers in a standard way. IPP requests will be completely encapsulated within the entity body of an HTTP request. The HTTP Entity-Header has the form

```
   HTTP-Entity-Header =      Content-Encoding
                             | Content-Length
                             | Content-Type
                             | extension-header
```

The Content-Length field must always be a valid length, This means that for any Print Operations based on HTTP 1.0, the entire content must be generated before this header can be built. HTTP 1.1 provides the notion of "chunks" which will allow the content to be generated dynamically as the data is sent.

Content-Type will always be "Application/IPP".

5.2.1.1 IPP Request-Line

The first line of the entity body in an IPP operation is the IPP Request-Line. The Request-Line defines the Operation and the IPP Version.

```
   IPP-Request-Line = Operation-token  IPP/1.0  CRLF

   Operation-token  = Print | Cancel-Job |
                      Get-Attributes | Get-Jobs
```

5.2.2 HTTP Response-Header Fields

HTTP response fields allow the server to pass additional information about the response back to the client. IPP will use these headers in a standard way. IPP responses will be completely encapsulated within the entity body of an HTTP response.

5.2.2.1 IPP Status-Line

The first line of the entity body in an IPP response is the IPP Status-Line. The status-line consists of a protocol version followed by a numeric status-code and an associated text message.

```
IPP-Status-Line = IPP/1.0 Status-Code  Reason-Phrase  CRLF
```

5.3 The Print Job

In section 5.4.1, the Print Operation is described. In order to understand that operation better, we first present the notion of a Print Job. The entity body of a print operation request will contain a Print Job, as defined below. The headers defined here are IPP headers, but follow the same syntax as the basic HTTP headers.

```
Print-Job = Print-Job-Object-Header  ;section (5.3.1)
            [Job-Attributes]           ;section (5.3.4)
            *(Documents)

   Document =     Document-Header       ;section (5.3.2)
                  [Document-attributes] ;section (5.3.5)
                  [Content-Header       ;section (5.3.3)
                   content]
```

5.3.1 Print Job Object Header

```
Print-Job-Object Header = Content-Encoding
             | Content-Length
             | Content-Type
             | exten sion-header
```

Content-Type is always "IPP Print Object". Other header fields are as defined for HTTP 1.0.

5.3.2 Document Header

The document header allows the insertion of multiple documents within a job. At this point only a limited number of document attributes are defined. However, this structure allows the addition of other attributes which can be specified on a document boundary.

```
Document-Header =  Content-Encoding
      | Content-Length
      | Content-Type
      | extension-header
```

Content type is always "IPP Document". Other header fields are as defined in HTTP 1.0.

5.3.3 Document-Content Header

The document-content-header provides additional meta-information about the document. The document content header is an optional field and would not be present if the document was pointed to by a document URL attribute. It is composed of a number of document header fields as follows:

```
Document-Content-Header =       Content-Encoding
                   | Content-Length
                   | Content-Type
                   | extension-header
```

Content-Type is defined as :

```
Content-Type = Data-Stream-Format "/" Version
```

Thus, for example, if the document to be printed was a Postscript Level 2 document, the Content-Type would be specified as:

```
     Content-Type: Postscript/2.0
```

Other header fields are as defined by HTTP 1.0.

5.3.4 Job Attributes

Job attributes are defined in section 6.2. Attributes will always be sent as

```
Job-Attribute = Attr-name ":" Attr-value     CRLF

Attr-value = 1#Value
```

In the above example, "1#Value" means one or more "," separated values.

5.3.5 Document Attributes

Document attributes are defined in section 6.2.11. The syntax for a document attribute is

```
Document-Attribute = Attr-Name ":" Attr-Value    CRLF

Attr-Value = 1#Value
```

In the above example, "1#Value" means one or more "," separated values.

5.4 Operation Semantics

In this section the four IPP operations are described in terms of their contents and semantics.

5.4.1 Print Operation

When an end user submits a job, the client submits a Print Request and receives a Print Response.

Note that the Printer name is not needed since it is the target of the entire operation. A Print Job contains the information needed by the Printer object to print a document or set of documents. When the print operation is invoked, the Entity-Body in the HTTP request includes an IPP Print Job. The concrete syntax of the Print Job is defined in section 5.3.

Each Printer object has an associated Job Template object assigned by the Administrator. When accepting a Print operation, the Printer shall use the corresponding value of an attribute from the Printer’s Job Template as the default value for any job attribute that the submitting client omits from the Print operation.

If neither the client nor the Printer's Job Template supplies a value for a job attribute, then the output device shall supply its own default value for that job attribute, if necessary, in order to produce output.

5.4.1.1 Print Request

The following abstract data types are part of the Print Request:

Job and Document Attributes — A set of Job object and Document attributes as defined in section 6.2

Requested Attributes — A set of attributes without values in whose values the requester is interested.

Document Contents — Document content is optional and shall not be included when a URL is provided in the document-URL attribute which points to the content.

5.4.1.2 Print Response

The following abstract data types are part of the Print Response:

Job-Identifier — A URL Used for all other operations on this Job.

Job Status — Current-job-state

Printer State — Printer-state

Result Attributes	The requested attributes with their current values, if the requester supplied any Requested Attributes
Message	Optional message
Errors	Optional Error Information

5.4.2 Cancel Job Operation

This operation allows a user to cancel one specific Print Job any time after the print job has been established on the Printer Object. Some pages may be printed before a job is terminated if printing has already started when the Cancel Job operation is received. Only the end-user who is also the job originator (job-originator Job attribute) can cancel the job.

The Cancel HTTP request will be sent to the URL identifying the job to be canceled.

5.4.2.1 Cancel-Job Request

The following abstract data types are part of the Cancel Job Request:

Message	Optional message to the operator.
job-retention-period	The number (cardinal) of minutes that that job is to be retained after the job has been canceled. This parameter updates the value of the job-retention-period that may have been submitted by the submitter in the Print operation.

5.4.2.2 Cancel-Job Response

The following abstract data types are part of the Cancel Job Response:

Job Status	Optional Job status information
Errors	Optional Error Information

5.4.3 Get Attributes Operation

This operation allows an end-user to obtain information from the Print object concerning jobs, printers, and print queues, based on ISO 10175. The entity-body of the Get Attributes operation contains the set of attributes that the requester is interested in. The requester should not supply values in the Requested Attributes input parameter; the Printer shall ignore the values of any supplied by the requester. The attribute list is returned in the response with the appropriate attribute values filled in. If no attribute list is supplied, then all attributes defined for that object are returned.

5.4.3.1 Get-Attributes Request

The following abstract data types are part of the Get Attributes Request:

Selector	Job-Identifier (URL) or Printer URL or Job Template URL
Requested Attributes	A set of attributes without values in whose values the requester is interested

5.4.3.2 Get-Attributes Response

The following abstract data types are part of the Get Attributes Response:

Result Attributes	The requested attributes of the object with their current values, if the requester supplied any Requested Attributes
Errors	Optional error information

5.4.4 Get Jobs Operation

This operation allows a client to retrieve a list of print jobs belonging to the target Printer object. A list of attributes the client is interested in seeing may be appended to the request. If no attributes are asked for the default set of job-name and total-job-octets is returned for each job along with the job-identifier. Jobs will be returned in the order in which they are scheduled to print.

5.4.4.1 Get-Jobs Request

The following abstract data types are part of the Get Jobs Request:

selector	Indicates which jobs the requester seeks. The values are type2Enum (see section 6). Standard values are: “ all-jobs” - including completed jobs

“pending” - all jobs which are pending and processing

"my-jobs" - my jobs that are pending or processing

Requested Attributes
A set of attributes without values in whose values the requester is interested.

5.4.4.2 Get-Jobs Response

The following abstract data types are part of the Get Jobs Response:

Jobs
A list of Job URLs is returned. The list is in "scheduled" order. The job-identifier attribute shall be returned as the first attribute of each job to mark the beginning of the set of attributes for the next job.

Result Attributes
In addition to the job-identifier attribute which is always returned, either the Requested Attributes are returned or the following attributes by default, if the requester did not supply any Requested Attributes: job-total-octets and number-of-intervening-job. This last attribute is necessary since an end user may request just their own jobs and they need some relative position indicator if there are other jobs interspersed in the waiting list which are not returned in the response or cannot be because of site security policy restrictions.

Errors
Optional Error Information

6. Object Attributes

This section describes the attributes, syntaxes, and values that are part of IPP. The sections below show the objects and their associated attributes which are included within the scope of this protocol. The text in these sections has been heavily influenced by the ISO/IEC 10175 DPA (Final, June 1996).

6.1 Attribute Syntaxes

The syntax for attribute values is specified using the notation of RFC 822.

The special syntax State is used to form other syntaxes for xxx-supported attributes of the Printer object that indicate job attributes that the Printer supports. Such support may include operator intervention, delivery of an order that the provider has previously placed, or may require that the provider place a special order. The syntax for State is itself a type2Enum. The standard values are: [":not-ready" / ":on-order" / ":special-order"]

An attribute value with an empty State means that the indicated value is ready to be used without human intervention.

An attribute value with a ":not-ready" State means that operator intervention is required.

An attribute value with a ":on-order" State means that the provider has placed an order for the indicated value and that the operator must wait until the resource is delivered before the job can be printed. However, an end-user may submit a job that requires such a resource and the Printer shall accept such a job.

An attribute value with a ":special-order" State means that the provider shall make a special order for the resource, when a job is submitted that needs such a resource. However, an end-user may submit a job that requires such a resource and the Printer shall accept such a job.

For example, the media-supported printer attribute might contain the following values:

```
   media-supported = na-letter-white, na-letter-transparent,
                     b:not-ready
```

Meaning that na-letter-white and na-letter-transparent are loaded into the two trays of the output device and that b is supported, but requires the operator to change the trays.

The sections below reference the following syntax items:

string	arbitrary ASCII strings, no control characters, except <SPACE>.
StringPair	string “:” string
stringState	string State
name	arbitrary ASCII strings, no control characters, and no <SPACE> characters.
Url	Universal Resource Locator
dateTime	date and time in RFC 822 format
deltaTime	[hours “:”] minutes
cardinal	0 .. n represented as ASCII digits
type1Enum	standard names, must revise the IPP standard to add a new name. No private names are allowed.

type2Enum	standard names, but an implementor can, at any time, add new values by proposing them to the PWG for registration (or an IANA-appointed registry advisor after the PWG is no longer certified) where they are reviewed for approvoal.. IANA keeps the registry. Implementors can support private (un-registered) with a suitable distinguishing prefix, such as -xxx- where xxx is the company name registered with IANA for use in domain names.
Type3Enum	standard names, but an implementor can add new values by submitting a registration request directly to IANA, no PWG or IANA-appointed registry advisor review is required. Implementors can support private (un-registered) names with a suitable distinguishing prefix, such as -xxx- where xxx is the company name registered with IANA for use in domain names.
type2EnumState	type2Enum State
type3EnumState	type3Enum State
boolean	tokens: yes, y, true, or t and no, n, false, or f.
positiveInteger	1 .. n represented as ASCII digits
positiveIntegerCross	positiveInteger [“x” positiveInteger]
positiveIntegerCrossState	positiveIntegerCross State
positiveIntegerRange	positiveInteger “:” positiveInteger
positiveIntegerUnits	positiveInteger units
positiveIntegerState	positiveInteger State
units	“ppm” | “ipm” | “spm” | “cps” | “lpm”
type3Locale	type3Country “:” type3Language “:” type3CodeSet
type3Country	type3Enum - Standard values are the two-character country codes from ISO 639.
type3Language	type3Enum - Standard values are the two-character language codes from ISO 3166.
type3CodeSet	type3Enum - Standard values are from the IANA Code Set registry.
type2Format	name [“/” version]
version	name
type3LocaleState	type3Locale State

887

Also, the following conventions (from RFC 822) are used:

"1#" in front of a data syntax	means one or more values separated by ",".

NOTE - For consistency, no Job (or Job Template) or Printer attribute has the syntax # meaning zero or more values separated by ",". Instead, a distinguished value, such as "none", is used to indicate no value. For the Printer Object, the omission of the attribute entirely, is also used to indicate no value. In all such cases for the Printer object where a conforming implementation may omit the attribute all together, an explicit sentence indicates the meaning of the Printer attribute when the attribute is unspecified.

6.2 Job Attributes

A job object contains a set of job attributes and one or more documents. A client shall create a job and send it to a server using the Print operation. When accepting a Print operation, the Printer shall use the corresponding value of an attribute from the Printer's Job Template as the default value for any job attribute that the submitting client omits from the Print operation.

A client may use a job template associated with the selected printer in order to initialize the job. To do so, the client uses the Get-Attributes operation to get the URLs of the Printer's Job Templates. Then the client may get the default attributes from the Printer's default Job Template in order to initialize a display to the end-user with the Printer's defaults. See the printer-job-templates Printer attribute. However, a client need not access the Job Template in order to issue a Print operation; the client can depend on the Printer to supply the default job object attribute values as part of the Print operation.

Each section heading below contains the name of an attribute and its syntax in parentheses using the rules of RFC 822.

6.2.1 Job Informational Attributes (Set by a Client/End User)

The client may specify these attributes in the Print operation to provide information to identify a print-job.

The client may also specify these attributes in the operations: Get-Attributes, and Get-Jobs.

6.2.1.1 job-name (string)

This attribute supplies a human readable string for naming the print-job.

This attribute is intended to be printed on a start sheet, returned in a Get-Jobs result, or used in notification messages.

If the client does not specify this attribute, a Printer shall set it to the value of the document-name attribute of the first document in the job.

6.2.2 Job Informational Attributes (Set by a Printer)

The Print shall add all of these attributes to a job to provide information to identify a print-job.

The client may specify these attributes in the operations: Get-Attributes and Get-Jobs, but not in Print.

6.2.2.1 job-identifier (url)

This attribute provides the job-identifier for this job on the Printer. The Printer shall generate a job-identifier value as a URL.

The value of the job-identifier attribute shall be returned by the Printer as part of the PrintResult in the Print operation.

6.2.2.2 job-originator (name)

This attribute specifies the name of the person submitting the print job. The Printer shall set this attribute to the most authentic name that it can obtain from the client. The operation-user-name attribute is intended to be a source of the most authentic name.

6.2.2.3 job-originating-host (name)

This attribute identifies the originating host of the job. The Printer shall set this attribute to the value of the operation-host-name which is intended to be the most authentic host name of the client.

6.2.2.4 job-locale (type3Locale)

This attribute identifies the locale of the job, i.e, the country, language, and coded character set. The Printer sets this attribute from the value of the operation-locale.

The Printer shall use this attribute to determine the locale for notification messages that it sends.

Issue: Is there a more standard syntax for locale?

6.2.3 Job Status Attributes (Set by Printer)

The Printer shall add these attributes to a job when a client submits a job, and the Printer shall assign appropriate values to each such job-status attribute.

The Printer uses these attributes to specify the job status before, during and after the processing of the print-job by the Printer.

The client may specify job-status attributes in: Get-Attributes and Get-Jobs, but not Print.

6.2.3.1 current-job-state (type1Enum)

This attribute identifies the current state of the job. Standard values are:

Unknown — The job state is not known, or is indeterminate.

held — The job is waiting to be released for scheduling for any number of reasons as specified by the value of the job's job-state-reasons attribute.

pending — The job is waiting to start processing on a printer.

processing — The server is processing the job, or has made the job ready for printing, but the output device is not yet printing it, either because the job hasn't reached the output device or because the job is queued in the output device or some other spooler, awaiting the output device to print it.

Or

The server has completed processing the job and the output device is currently printing the job. That is, an output device is either printing pages of the job, or failing in its attempt to print pages of the job because of some wait state, such as, start-wait, end-wait, needs-attention, etc. The complete job state includes the detailed status represented in the printer's printer-state attribute.

paused — The job has been paused

Interrupted — The job has been interrupted by some intervening job, and shall resume processing automatically once the intervening job has completed.

Terminating The job has been canceled by a Cancel-Job request or aborted by the server and is in the process of terminating. The job's job-state-reasons attribute contains the reasons that the job is being terminated.

Retained The job is being retained at the server as a result of the job's job-retention-period being non-zero. The job has (1) completed successfully or with warnings or errors, (2) been aborted while printing by the server, or (3) been canceled by the Cancel-Job request before or during processing. The job's job-state-reasons attribute contains the reasons that the job has been retained.

While in the retained state, all of the job's document data (and resources, if any) shall be retained by the server; thus a job in the retained state could be reprinted, using some means outside the scope of IPP V1.0.

Completed The job has:

(1) completed successfully or with warnings or errors,
(2) been aborted by the server while printing, or
(3) been canceled by the Cancel-Job request,

AND the job's:
(1) job-retention-period was zero or has expired, or
(2) job-discard-time has arrived.

The job's job-state-reasons attribute contains the reason(s) that the job has been completed.

While in the completed state, a job's document data (and resources if any) need not be retained by the server; thus a job in the completed state could not be reprinted. The length of time that a job may be in this state, before transitioning to unknown, is implementation-dependent. However, servers that implement the completed job-state shall retain, as a minimum, the following attributes for any job in the completed state: job-identifier, job-originator, job-name, current-job-state, output-device-assigned, and job-state-reasons.

The IPP protocol supports all values for job states, but Printers need only support those states which are appropriate for the particular implementation.

6.2.3.2 output-device-assigned (name)

This attribute identifies the Output Device to which the Printer has assigned this job.

If an Output Device implements a Printer, the Printer need not set this attribute.

If a Print Server implements a Printer, the value shall be empty until the Printer assigns an Output Device to the job.

The value of the job's output-device-assigned attribute shall remain after the job has completed, so that end users can determine the Output Device on which the job was printed.

6.2.3.3 submission-time (dateTime)

This attribute indicates the time at which this job was accepted by the Printer. If the Printer does not support the notion of time, the attribute need not be stored as part of the job object.

6.2.3.4 number-of-intervening-jobs (cardinal)

This attribute indicates the number of jobs that are "ahead" of this job in the current scheduled order. For efficiency, it is only necessary to calculate this value when an operation if performed that requests this attribute.

NOTE - This attribute is necessary since an end user may request just their own jobs and they need some relative position indicator if there are other jobs interspersed in the waiting list which are not returned in the response or cannot be because of site security policy restrictions.

6.2.3.5 job-message-from-operator (string)

This attribute provides a message from an operator, system administrator or "intelligent" process to indicate to the end user the reasons for modification or other management action taken on a job.

6.2.3.6 completion-time (dateTime)

This attribute indicates the time at which this job completed. This time is useful for jobs which are retained after printing. If the Printer does not support the notion of time, the attribute is not stored as part of the Job object.

6.2.3.7 job-state-reasons (1#type2Enum)

This attribute identifies the reason or reasons that the job is in the state that it is in (e.g., held, terminating, retained, completed, etc.). The printer shall indicate the particular reason(s) by setting the value of the job-state-reasons attribute.

The following standard values are defined:

none	There are not reasons associated with the job's current state.
documents-needed	The complete job has been accepted by the server, but the server is waiting for its files to be transferred before the job can be scheduled to be printed.
job-hold-set	The value of the job's job-hold attribute is TRUE.
job-print-after-specified	The value of the job's job-print-after or print-off-peak attributes have specified a time specification that has not yet occurred.
Required-resources-not-ready	At least one of the resources needed by the job, such as media, fonts, resource objects, etc., is not ready on any of the physical printer's for which the job is a candidate.
Successful completion	The job completed successfully.
Completed-with-warnings	The job completed with warnings.
Completed-with-errors	The job completed with errors (and possibly warnings too).
Cancelled-by-user	The job was cancelled by the user using the CancelJob request.
Cancelled-by-operator	The job was cancelled by the operator using the CancelJob request.
Aborted-by-system	The job was aborted by the system.
Logfile-pending	The job's logfile is pending file transfer.
Logfile-transferring	The job's logfile is being transferred.

6.2.3.8 impressions-completed (cardinal)

This attribute contains the number of impressions that the Printer has completed printing. If the Printer cannot report this number, the Printer leaves this attribute unspecified.

6.2.3.9 media-sheets-completed (cardinal)

This attribute contains the number of media-sheets that the Printer has completed printing. If the Printer cannot report this number, the Printer leaves this attribute unspecified.

6.2.4 Job Sheet Attributes (Set by Client/End User)

The client shall specify these attributes to control the printing of job sheets.

The client may also specify job sheet attributes in: Get-Attributes and Get-Jobs.

6.2.4.1 job-sheets (type3Enum)

This attribute determines what type of job-sheets the Printer shall print with the job.

The standard values are: none, and default-sheet.

The value "none" means that the Printer shall print no job sheets. The value "default-sheet" means that the Printer shall print the job sheets defined by an administrator. If the administrator's policy is not to support none, the Printer shall use the default-sheet value if the client supplies the "none" value.

NOTE - The effect of this attribute on jobs and documents is controlled by the files-are-one-document and files-are-interleaved job attributes.

6.2.5 Notification Attributes (Set by a Client/End User)

The client shall specify these attributes to indicate events that the client is interested in, along with the notification address and method for performing the notification.

The client may also specify notification attributes in: Get-Attributes and Get-Jobs.

6.2.5.1 notification-events (1#type2Enum)

This attribute specifies the events about which the end user want to be notified.

Standard values are: none, job-completion, job-problems and printer-problems.

If this attribute contains the event none, the Printer shall not notify. This value is useful if an administrator has set up a notification Printer default but the end user does not want notification. If the none value and other values are supplied, the Printer shall ignore the none value.

If this attribute contains the value: job-completion, the Printer shall notify the client when the job containing this attribute completes with or without errors or is cancelled by the end-user or the operator.

If this attribute contains the value: job-problems, the Printer shall notify the client when this job has a problem while this job is printing. Problems include: paper jam and out-of-paper.

If this attribute contains the value: printer-problems, the Printer shall notify the client when any job, including this job, has a problem while this job is waiting to print or printing. Problems include: paper jam and out-of-paper.

6.2.5.2 notification-address (url)

This address specifies both the address and mechanism for delivery of notification events to the client. The client specifies this attribute in the operation-notification-address attribute which the Printer in turn uses to set this attribute.

The Printer shall use this attribute as the address for sending messages to a job submitter when an event occurs that the end user has registered an interest in or when certain other events occur, such as Cancel-Job.

If the URL has a "mailto:" scheme, then email is used and the rest of the URL is used as the email address. If the URL has a "http:" scheme, then an HTTP method is used to add HTML formatted events to the end of the specified HTML file.

6.2.6 Job Scheduling Attributes (Set by Client/End User)

The client shall specify these attributes to provide the Printer with information for the scheduling a print-job.

The client may also specify these attributes in: Get-Attributes and Get-Jobs.

6.2.6.1 job-priority (type1Enum)

This attribute specifies a priority for scheduling the print-job. Printers that employ a priority-based scheduling algorithm use this attribute.

There are three standard values: high, default, and low. Among those jobs that are ready to print, a Printer shall print all such jobs with a high priority before printing those with a default or low priority, and a Printer shall print all such jobs with a default priority before printing those with a low priority.

If the client does not specify this attribute, the Printer assumes that the end user places no constraints concerning priority on the scheduling of the print-job, and it has a priority value of default.

An operator can modify a job to have any priority. An end-user is restricted by the value of the maximum-end-user-priority Printer attribute.

6.2.6.2 job-print-after (dateTime)

This attribute specifies the calendar date and time of day after which the print-job shall become a candidate for printing.

If the value of this attribute is in the future, the Printer shall set the value of the job's current-job-state to held and add the job-print-after-specified value to the job's job-state-reasons attribute and shall not schedule the print-job for printing until the specified date and time has passed. When the specified date and time arrives, the Printer shall remove the job-print-after-specified value from the job's job-state-reason attribute and, if no other reasons remain, shall change the job's current-job-state to pending so that the job becomes a candidate for being scheduled to print.

If this attribute is unspecified or the value is in the past, the job shall be a candidate for scheduling immediately.

6.2.6.3 job-print-off-peak (type3Enum)

This attribute specifies the off-peak period during which the print-job shall become a candidate for printing.

Standard values are: "evening", "night", "weekend", "second-shift", "third-shift".

If this attribute is specified, it contains a value with which an administrator has associated allowable print times. An administrator is encouraged to pick names that suggest the type of off-peak period.

If the value of this attribute is in the future, the Printer shall set the value of the job's current-job-state to held and add the job-print-after-specified value to the job's job-state-reasons attribute and shall not schedule the print-job for printing until the specified date and time has passed. When the specified date and time arrives, the Printer shall remove the job-print-after-specified value from the job's job-state-reason attribute and, if no other reasons remain, shall change the job's current-job-state to pending so that the job becomes a candidate for being scheduled to print.

If this attribute is unspecified, the job shall be a candidate for scheduling immediately.

6.2.6.4 job-retention-period (deltaTime)

The retention time is expressed in hours and minutes, e.g. 6:00 (6 hours), or 20 (20 minutes).

This attribute specifies the minimum period of time following the completion of job processing and printing that the server shall keep job attributes and document data. The Printer may keep these attributes and data longer than the value of the job-retention-period attribute.

NOTE - the requester may change this job attribute using the input parameter to the Cancel-Job operation.

6.2.7 Job Production Attributes (Set by Client/End User)

The client shall specify these attributes to affect the rendering, production and finishing of the documents in the job. Similar types of instructions may also be contained in the document to be printed.

If there is a conflict between the value of one of these attributes, and a corresponding instruction in the document (either implicit or explicit), the value of the attribute shall take precedence over the document instruction.

Job Production and Resource Attributes each address a similar set of features but they have different uses.

A job production attribute provides a client with a way to request some feature at print time that may not have been embedded within the document data when the document was created. A job production attribute also provides a client with a way to override a feature at print time that was embedded within the document data when the document was created.

Note: until companies that supply interpreters for PDL's, such as PostScript and PCL allow a way to specify overrides for internal job production instructions, a Printer may not be able to implement these attributes for some PDL's.

A job resource attribute tells a Printer what features the job needs. A program that translates document data to a Printer's PDL, and/or merges production attributes into the document data should add job resource attributes to a job.

For example, a job production attribute medium-select with the value of "letter" requests that a job be printed on letter paper, but gives no information about what resources the job needs. For example, a job resource attribute media-used with the values of "letter" and "ledger" tell a Printer that the job needs letter and ledger paper, but gives no information about which pages use each medium.

The client may also specify job production-instruction attributes in: Get-Attributes and GetJobs.

6.2.7.1 medium-select (type2Enum)

This attribute identifies the medium that the Printer shall use for all pages of the document regardless of what media are specified within the document.

The values for medium include medium-names, medium-sizes, input-trays and electronic forms so that one attribute specifies the media.

Standard values are (taken from ISO DPA and the Printer MIB):

default	The default medium for the output device
iso-a4-white	Specifies the ISO A4 white medium
iso-a4-colored	Specifies the ISO A4 coloured medium
iso-a4-transparent	Specifies the ISO A4 transparent medium
iso-a3-white	Specifies the ISO A3 white medium
iso-a3-colored	Specifies the ISO A3 coloured medium
iso-a5-white	Specifies the ISO A5 white medium
iso-a5-colored	Specifies the ISO A5 coloured medium
iso-b4-white	Specifies the ISO B4 white medium
iso-b4-colored	Specifies the ISO B4 coloured medium
iso-b5-white	Specifies the ISO B5 white medium
iso-b5-colored	Specifies the ISO B5 coloured medium
jis-b4-white	Specifies the JIS B4 white medium
jis-b4-colored	Specifies the JIS B4 coloured medium
jis-b5-white	Specifies the JIS B5 white medium
jis-b5-colored	Specifies the JIS B5 coloured medium

The following standard values are defined for North American media:

na-letter-white	Specifies the North American letter white medium
na-letter-colored	Specifies the North American letter coloured medium
na-letter-transparent	Specifies the North American letter transparent medium
na-legal-white	Specifies the North American legal white medium
na-legal-colored	Specifies the North American legal coloured medium

The following standard values are defined for envelopes:

iso-b4-envelope	Specifies the ISO B4 envelope medium
iso-b5-envelope	Specifies the ISO B5 envelope medium

iso-c3-envelope	Specifies the ISO C3 envelope medium
iso-c4-envelope	Specifies the ISO C4 envelope medium
iso-c5-envelope	Specifies the ISO C5 envelope medium
iso-c6-envelope	Specifies the ISO C6 envelope medium
iso-designated-long-envelope	Specifies the ISO Designated Long envelope medium
na-10x13-envelope	Specifies the North American 10x13 envelope medium
na-9x12-envelope	Specifies the North American 9x12 envelope medium
monarch-envelope	Specifies the Monarch envelope
na-number-10-envelope	Specifies the North American number 10 business envelope medium
na-7x9-envelope	Specifies the North American 7x9 inch envelope
na-9x11-envelope	Specifies the North American 9x11 inch envelope
na-10x14-envelope	Specifies the North American 10x14 inch envelope
na-number-9-envelope	Specifies the North American number 9 business envelope
na-6x9-envelope	Specifies the North American 6x9 inch envelope
na-10x15-envelope	Specifies the North American 10x15 inch envelope

The following standard values are defined for the less commonly used media (white-only):

executive-white	Specifies the white executive medium
folio-white	Specifies the folio white medium
invoice-white	Specifies the white invoice medium
ledger-white	Specifies the white ledger medium
quarto-white	Specified the white quarto medium
iso-a0-white	Specifies the ISO A0 white medium
iso-a1-white	Specifies the ISO A1 white medium
iso-a2-white	Specifies the ISO A2 white medium
iso-a6-white	Specifies the ISO A6 white medium
iso-a7-white	Specifies the ISO A7 white medium
iso-a8-white	Specifies the ISO A8 white medium
iso-a9-white	Specifies the ISO A9 white medium
iso-10-white	Specifies the ISO A10 white medium
iso-b0-white	Specifies the ISO B0 white medium
iso-b1-white	Specifies the ISO B1 white medium
iso-b2-white	Specifies the ISO B2 white medium
iso-b3-white	Specifies the ISO B3 white medium
iso-b6-white	Specifies the ISO B6 white medium
iso-b7-white	Specifies the ISO B7 white medium
iso-b8-white	Specifies the ISO B8 white medium
iso-b9-white	Specifies the ISO B9 white medium
iso-b10-white	Specifies the ISO B10 white medium
jis-b0-white	Specifies the JIS B0 white medium

jis-b1-white	Specifies the JIS B1 white medium
jis-b2-white	Specifies the JIS B2 white medium
jis-b3-white	Specifies the JIS B3 white medium
jis-b6-white	Specifies the JIS B6 white medium
jis-b7-white	Specifies the JIS B7 white medium
jis-b8-white	Specifies the JIS B8 white medium
jis-b9-white	Specifies the JIS B9 white medium
jis-b10-white	Specifies the JIS B10 white medium

The following standard values are defined for engineering media:

a	Specifies the engineering A size medium
b	Specifies the engineering B size medium
c	Specifies the engineering C size medium
d	Specifies the engineering D size medium
e	Specifies the engineering E size medium

The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

top	The top input tray in the printer.
middle	The middle input tray in the printer.
bottom	The bottom input tray in the printer.
envelope	The envelope input tray in the printer.
manual	The manual feed input tray in the printer.
large-capacity	The large capacity input tray in the printer.
Main	The main input tray
side	The side input tray

The following standard values are defined for media sizes (from ISO dPA):

iso-a0	Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216
iso-a1	Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216
iso-a2	Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216
iso-a3	Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216
iso-a4	Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216

iso-a5 Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216
iso-a6 Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216
iso-a7 Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216
iso-a8 Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216
iso-a9 Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216
iso-a10 Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216

1195
iso-b0 Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216
iso-b1 Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216
iso-b2 Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216
iso-b3 Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216
iso-b4 Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216
iso-b5 Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216
iso-b6 Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216
iso-b7 Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216
iso-b8 Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216
iso-b9 Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216
iso-b10 Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216

1196
na-letter Specifies the North American letter size: 8.5 inches by 11 inches
na-legal Specifies the North American legal size: 8.5 inches by 14 inches
executive Specifies the executive size (7.25 X 10.5 in)
folio Specifies the folio size (8.5 X 13 in)
invoice Specifies the invoice size (5.5 X 8.5 in)
ledger Specifies the ledger size (11 X 17 in)
quarto Specifies the quarto size (8.5 X 10.83 in)

1197
iso-c3 Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269
iso-c4 Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269
iso-c5 Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269

iso-c6	Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269
iso-designated-long	Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO 269

1198

na-10x13-envelope	Specifies the North American 10x13 size: 10 inches by 13 inches
na-9x12-envelope	Specifies the North American 9x12 size: 9 inches by 12 inches
na-number-10-envelope	Specifies the North American number 10 business envelope size: 4.125 inches by 9.5 inches
na-7x9-envelope	Specifies the North American 7x9 inch envelope size
na-9x11-envelope	Specifies the North American 9x11 inch envelope size
na-10x14-envelope	Specifies the North American 10x14 inch envelope size
na-number-9-envelope	Specifies the North American number 9 business envelope size
na-6x9-envelope	Specifies the North American 6x9 envelope size
na-10x15-envelope	Specifies the North American 10x15 envelope size
monarch-envelope	Specifies the Monarch envelope size (3.87 x 7.5 in)

1199

a	Specifies the engineering A size: 8.5 inches by 11 inches
b	Specifies the engineering B size: 11 inches by 17 inches
c	Specifies the engineering C size: 17 inches by 22 inches
d	Specifies the engineering D size: 22 inches by 34 inches
e	Specifies the engineering E size: 34 inches by 44 inches

1200

jis-b0	Specifies the JIS B0 size: 1030mm x 1456mm
jis-b1	Specifies the JIS B1 size: 728mm x 1030mm
jis-b2	Specifies the JIS B2 size: 515mm x 728mm
jis-b3	Specifies the JIS B3 size: 364mm x 515mm
jis-b4	Specifies the JIS B4 size: 257mm x 364mm
jis-b5	Specifies the JIS B5 size: 182mm x 257mm
jis-b6	Specifies the JIS B6 size: 128mm x 182mm
jis-b7	Specifies the JIS B7 size: 91mm x 128mm
jis-b8	Specifies the JIS B8 size: 64mm x 91mm
jis-b9	Specifies the JIS B9 size: 45mm x 64mm
jis-b10	Specifies the JIS B10 size: 32mm x 45mm

1201

6.2.7.2 finishing (type2Enum)

This attribute identifies the finishing operation that the Printer should apply to each copy of the printed document.

NOTE - The effect of this atttribute on jobs and documents is controlled by the files-are-one-document and files-are-interleaved job attributes.

Standard values for this attribute are:

none	Perform no finishing.
staple	This indicates that staples are to be used to bind the document. The exact number and placement of the staples is site-defined; other finishing object attributes may be included to provide this information.
staple-top-left	This indicates that one or more staples should be placed on the top left corner of the document
staple-bottom-left	This indicates that one or more staples should be placed on the bottom left corner of the document
staple-top-right	This indicates that one or more staples should be placed on the top right corner of the document
staple-bottom-right	This indicates that one or more staples should be placed on the bottom right corner of the document
saddle-stitch	This indicates that one or more staples (wire stitches) are to be used to bind the document along the middle fold. The exact number and placement of the stitches is site-defined.
edge-stitch	This indicates that one or more staples (wire stitches) are to be used to bind the document along one edge. The exact number and placement of the staples is site-defined.
punch	This indicates that holes are required in the finished document. The exact number and placement of the holes is site-defined The punch specification may be satisfied (in a site- and implementation-specific manner) either by drilling/punching, or by substituting predrilled media.
cover	This value is specified when it is desired to select a non-printed (or pre-printed) cover for the document. This does not supplant the specification of a printed cover (on cover stock medium) by the document itself.

```
    bind            This indicates that a binding is to be
                    applied to the document; the type and
                    placement of the binding is site-defined.
```

6.2.7.3 number-up (type3Enum)

This attribute specifies the number of source page-images to impose upon a single side of an instance of a selected medium.

In general, only certain numeric values are valid for this attribute and the value "none", depending upon the Printer implementation to which the print-request is directed. Standard values are: "none", "1", "2", "4".

This attribute primarily controls the translation, scaling and rotation of page images, but a site may choose to add embellishments, such as borders to each logical page. The value "none" shall not include any embellishments and shall place one logical page on a single side of an instance of the selected medium without any translation, scaling, or rotation.

6.2.7.4 sides (type2Enum)

This attribute specifies how source page-images are to be imposed upon the sides of an instance of a selected medium.

The standard values are: 1-sided, 2-sided-long-edge, 2-sided-short-edge.

1-sided imposes each consecutive source page-image upon the same side of consecutive media sheets.

2-sided-long-edge imposes each consecutive pair of source page-image upon front and back sides of consecutive media sheets, such that the orientation of each pair of source-pages on the medium would be correct for the reader as if for binding on the long edge. This imposition is sometimes called "duplex".

2-sided-short-edge imposes each consecutive pair of source page-image upon front and back sides of consecutive media sheets, such that the orientation of each pair of source-pages on the medium would be correct for the reader as if for binding on the short edge. This imposition is sometimes called "tumble" or "head-to-toe".

Issue: How does sides interact with portrait vs. landscape and reverse-landscape documents?

6.2.7.5 copies (positiveInteger)

This attribute specifies the number of copies of the job to be printed. If this attribute is unspecified by both the client and the Printer's Job Template, its default value shall be 1.

NOTE - The effect of this atttribute on jobs and documents is controlled by the files-are-one-document and files-are-interleaved job attributes.

6.2.7.6 printer-resolution-select (positiveIntegerCross)

This attribute specifies the resolution that the Printer should use.

The syntax allows a single integer to specify the resolution or a pair of integers to specify the resolution when the x and y dimensions differ. When two integers are specified, the first is in the x direction, ie., in the direction of the shortest dimension of the medium, so that the value is independent of whether the printer feeds long edge or short edge first.

6.2.7.7 print-quality (type2Enum)

This attribute specifies the print quality that the Printer should use.

The standard values are:

draft	Lowest quality available on the printer
normal	Normal or intermediate quality on the printer
high	Highest quality available on the printer

6.2.7.8 page-select (positiveIntegerRange)

This attribute specifies the pages in the document that the Printer shall use. This attribute is unlikely to be useful for jobs with more than one document or in Job Templates. If this attribute is unspecified, then the Printer shall print all pages in a document.

6.2.7.9 files-are-one-document (boolean)

This attribute is relevant only if a job consists of two or more documents. It controls finishing operations, job-sheet placement, and the order of documents when the copies attribute exceeds 1.

If the files for the job are a and b and this attribute is true, then files a and b are treated as a single document for finishing operations. Also, there will be no slip sheets between files a and b. If more than one copy is made, the ordering must be a, b, a, b, ... The attribute files-are-interleaved is ignored.

If the files for the job are a and b and this attribute is false or unspecified by both the client and the Printer's Job Template, then each file is treated as a single document for finishing operations. Also, a client may specify that a slip sheet be between files a and b. If more than one copy is made, and the attribute files-are-interleaved false or unspecified, the ordering is a, a, b, b, ... If more than one copy is made, and the attribute files-are-interleaved true, the ordering is a, b, a, b, ...

6.2.7.10 files-are-interleaved (boolean)

This attribute is used in conjunction with files-are-one-document (q.v.).

6.2.8 Attributes for Conversion of Text and HTML Files (Set by Client/End User)

The client shall specify these attributes to control formatting for text documents or HTML documents.

A client need not specify these attributes for other types of documents, such as PostScript or PCL.

6.2.8.1 width (cardinalUnits)

This attribute specifies the media width for the document in characters.

6.2.8.2 length (cardinalUnits)

This attribute specifies the media length for the document in characters.

6.2.8.3 left-margin (cardinalUnits

This attribute specifies the left-margin for the document in characters.

6.2.8.4 right-margin (cardinalUnits)

This attribute specifies the right-margin for the document in characters.

6.2.8.5 top-margin (cardinalUnits)

This attribute specifies the top-margin for the document in lines.

6.2.8.6 bottom-margin (cardinalUnits)

This attribute specifies the bottom-margin for the document in lines.

6.2.8.7 repeated-tab-stops (cardinalUnits)

This attribute specifies the tab stops for the document in characters.

6.2.8.8 header-text (string)

This attribute specifies the header text for the document.

6.2.8.9 footer-text (string)

This attribute specifies the footer text for the document.

6.2.8.10 number-pages (boolean)

This attribute specifies that the pages should be numbered in the document.

6.2.8.11 default-font (string)

This attribute specifies the font to use for all text in the document.

6.2.8.12 font-size (cardinalUnits)

This attribute specifies the font-size in points for text in the document. The value of this attribute affects the size of the other text attributes.

If this attribute is omitted and the Printer's default Job Template does not contain this attribute, the Printer shall assume a value of 10. A value of 10 with a fixed pitch font, shall produce 12 characters per inch in the horizontal direction and with 6 lines per inch in the vertical direction.

6.2.8.13 default-code-set (type3Enum)

This attribute specifies the code-set in which the document is encoded.

6.2.8.14 content-orientation (type2Enum)

This attribute specifies the orientation of the document.

The standard values are:

portrait The page orientation such that the sides are longer than the top when the page is held in the intended human reading orientation

landscape The page orientation such that the sides are shorter than the top when the page is held in the intended human readable orientation. Landscape is defined to be a rotation of the page by +90 degrees with respect to the medium (i.e. anti-clockwise) from the portrait orientation
NOTE – The +90 direction was chosen because simple finishing on the long edge is the same edge whether portrait or landscape

reverse-portrait The page orientation defined to be a rotation of 180 degrees with respect to portrait

reverse-landscape The page orientation defined to be a rotation of 180 degrees with respect to landscape. Landscape is defined to be a rotation of the page by -90 degrees with respect to the medium (i.e. clockwise) from the portrait orientation
NOTE – Reverse-landscape was added because some applications rotate landscape -90 degrees from portrait, rather than +90 degrees.

6.2.9 Job Resource Attributes (Set by the program that produces or senses the PDL)

A program (described below) shall add these attributes, which describe the resources needed to print the job.

A Printer may use these attributes to validate and schedule the print-job without interpreting the contents of the document. This provides the opportunity for a Printer to support a broad set of document formats yet still support fast efficient scheduling and validation of each job.

The client/end user shall not specify these attributes. Instead, it is the duty of the program that translates the document to the printer's PDL (or analyzes it) to add these attributes and their values to the job. Such a program may execute at a number of different points in time:

1. The program produces a final form document and stores these resource attributes in a file before the end-user submits the print job.

2. The program produces a final form document data stream when the end-user specifies "Print" to the application program (e.g., Windows GDI driver).

3. The program running in the context of the Printer or server translates a revisable or final form document into a PDL that the output device understands.

If any of these attributes is unspecified, the Printer shall assume that the all resources required by the document of the type specified by the missing attributes are ready, ie., are available to the Printer and/or output device without human intervention.

These attributes may be unspecified if the translation program fails to provides such values, or if no translation occurs (e.g. the document is a PostScript document.

Note: The Printer does not use these attributes during the actual printing of a document.

Note: these attributes allow more than one value wherever it is possible for a job to specify more than one value of the corresponding job attribute, possibly by embedded instructions.

The client may specify these attributes in: Get-Attributes and Get-Jobs.

See the section on job production attributes for an explanation of how the job resource attributes differ from the job production attributes.

6.2.9.1 document-formats-used (1#type2Format)

This attribute identifies the document formats needed to print the document(s) in this job.

A format consists of two elements, a name and a version. The latter element is optional.

The syntax is for type2Format:

```
     name [ "/" version ]
```

Examples include: PostScript, PostScript/2.0 and PCL/5e

Note: The version component is optional.

The names shall be registered with IANA as "printer languages" following the procedures established by the Printer MIB (currently proposed as an ITEF standard by RFC 1759).

6.2.9.2 fonts-used (1#string)

This attribute identifies the font resources used in the document(s) in the job.

6.2.9.3 code-sets-used (1#type3Enum)

This attribute identifies the code-sets used in the document(s) in the Job. This attribute is relevant only for files that are not in ASCII, such as text files and possibly PCL files. PostScript files are always ASCII. Normally there is at most 1 code-set.

Standard values are defined in the section specifying the default-code-set attribute.

6.2.9.4 media-used (1#type2Enum)

This attribute identifies the media, media-sizes, input-trays or electronic forms needed to print the document(s) in the job.

Standard values for this attribute are defined in the section specifying the medium-select attribute.

6.2.9.5 sides-used (type2Enum)

This attribute specifies whether a job needs 1-sided, 2-sided-long-edge, or 2-sided-short-edge printing.

Standard values for this attribute are defined in the section specifying the sides Job attribute.

6.2.9.6 print-quality-used (type2Enum)

This attribute specifies what print quality the job needs.

Standard values for this attribute are defined in the section specifying the print-quality attribute.

6.2.9.7 finishing-used (type2Enum)

This attribute specifies what finishing the job needs.

Standard values for this attribute are defined in the section specifying the finishing attribute.

6.2.9.8 printer-resolution-used (positiveIntegerCrossState)

This attribute specifies what resolution the job needs.

The interpretation of the values for this attribute are defined in the section on printer-resolution-select Job attribute.

6.2.9.9 total-job-octets (positiveInteger)

This attribute specifies the total size of the job in octets. This attribute is the first of three that a translation program can use to specify the size of a job.

6.2.9.10 job-impression-count (positiveInteger)

This attribute specifies the total size of the job in impressions.

6.2.9.11 job-media-sheet-count (positiveInteger)

This attribute specifies the total size of the job in media-sheets.

6.2.10 Number of Documents (Set by Printer)

This group contains a single attribute which specifies the number of documents in the job.

The Printer sets the value of this attribute depending on the number of documents that the client supplies in the Print operation. The client shall not specify this attribute (directly) in Print, but may specify this attribute in: Get-Attributes and Get-Jobs.

6.2.10.1 number-of-documents (positiveInteger)

This attribute specifies the number of documents in the job. Each document shall contain its own set of document content attributes described below.

6.2.11 Document Data (Set by a Client/End User)

This group of attributes describes the document data for the job. These attributes also include the document data or reference it.

All job attributes in other sections of this document occur only once per job and apply to all documents in a job.

The client may specify document-data attributes in Print. The client must specify either the document-URL or document-content in Print.

Except for document-content, the client may specify document-data attributes in: Get-Attributes, and Get-Jobs.

6.2.11.1 document-format (type2Format)

This attribute identifies the document format of this document.

If the client does not specify this attribute, then the Printer shall attempt to determine the format in order to decide if the document data needs to be translated. The version component is optional.

6.2.11.2 document-name (string)

This attribute contains the name of the document used by the client to initially identify the document.

6.2.11.3 document-URL (url)

This attribute contains the URL of the document if the client specified the document with a URL.

If this attribute is specified, then document-content shall be unspecified.

6.2.11.4 document-content (octetString)

This attribute contains the actual contents of the document.

If this attribute is specified, then document-URL shall be unspecified.

This attribute shall be used during the transmission of the Print operation over a network. A Printer shall save the document data to a file and reference it with the document-URL. A Get-Attribute or Get-Jobs operation shall always find that this attribute is unspecified.

6.3 Operation Attributes (Set by Client)

NOTE: These attributes have just been introduced and they are not as stable as the attributes in the other sections. Some work is still needed to show the relationship between these attributes, job attributes, printer attributes, and authentication and authorization.

The client shall set these attributes and associate them with an operation rather than an object.

It is intended that a client program rather than an end-user has control over the setting of these values so that they cannot be easily forged.

6.3.1 operation-locale (type3Locale)

This attribute identifies the locale of the client. The Printer uses this attribute to determine the locale of (1) messages in the result of the operation, (2) in errors returned by the operation or (3) notification events sent to the submitter.

The standard values are defined in the section on the job-locale attribute.

If an operation does not specify this attribute, the Printer shall assume that the operation has the same locale as the Printer.

6.3.2 operation-notification-address (url)

This attribute specifies both the address and mechanism for delivery of events. If the URL has a "mailto:" scheme, then email is used and the rest of the URL is used as the email address. If the URL has a "http:" scheme, then an HTTP APPEND method is used to add HTML formatted events to the end of the specified HTML file.

6.3.3 operation-user-name (name)

This attribute identifies the most authenticated end-user name that the client can supply. This name identifies the end-user performing the operation.

This value shall be set by the system rather than the end-user in order to minimize the chance of forgery.

6.3.4 operation-host-name (name)

This attribute identifies the most authenticated host name that the client can supply. This name identifies the host from which the operation comes.

This value shall be set by the system rather than the end-user in order to minimize the chance of forgery.

6.4 Printer Attributes (Set by the Administrator)

A printer object may be realized in either a Print Server or Output Device. Note: How these attribute are set by an Administrator is outside the scope of this specification.

A Printer Object in an Output Device contains a set of printer object attributes that represent an Output Device capable of rendering a document in visible form. Examples include electronic and electro-mechanical printers such as laser printers, ink-jet printers, and various kinds of impact printers, but may include other types of output devices such as microfiche imagers and plotters as well.

A Printer Object in a Print Server may supply queuing, spooling, and scheduling for an Output device that does not queue or spool.

A Print Server, in the most common case, controls exactly one downstream Output Device. The Print Server's Printer object has attributes whose values are the same as those of the Printer object in the downstream Output Device.

A Printer Object in a Print Server may contain a set of printer object attributes that are the union of the Printer objects in the downstream Output Devices. This object extends the capabilities of an Output Device. For example, an administrator might define a single Print Server to represent all of the Output Devices of the same type and capability in a single location, associated with a particular server. A end user would normally send a print-job to a Print Server, and allow the Print Server to assign the job to a particular Output Device based on the relative load and availability of the printers under its control, thus providing a load balancing service. However, nothing precludes an administrator from configuring a print system so that an end user can send a print-job directly to an Output Device.

The attributes defined in this section provide information about a particular Printer.

6.4.1 printer-name (name)

This attribute uniquely identifies the printer on its host.

6.4.2 printer-location (string)

This attribute identifies the location of this printer.

6.4.3 printer-model (string)

This attribute identifies the make and model of the printer.

6.4.4 printer-type (type2Enum)

This attribute identifies the marking technology of the printer.

The standard values for this attribute are the descriptive names specified by ISO DPA which have corresponding enum symbolic and numeric values assigned by the Printer MIB (RFC 1759).. These standard values are:

other	Other than the standard values
unknown	Unknown printer type
electrophotographic-LED	electrophotographic LED
electrophotographic-laser	electrophotographic laser
electrophotographic-other	other electrophotographic
impact-moving-head-dot-matrix-9-pin	9-pin impact moving head dot matrix
impact-moving-head-dot-matrix-24-pin	24-pin impact moving head dot matrix
impact-moving-head-dot-matrix-other	neither 9-pin nor 24-pin moving head dot matrix
impact-moving-head-fully-formed	fully formed impact moving head
impact-band	impact band
impact-other	impact other
inkjet-aqueous	aqueous inkjet

inkjet-solid	solid inkjet
inkjet-other	other inkjet
pen	pen
thermal-transfer	thermal transfer
thermal-sensitive	thermal sensitive
thermal-diffusion	thermal diffusion
thermal-other	other thermal
electro-erosion	electro-erosion
electro-static	electro-static
photographic-microfiche	photographic microfiche
photographic-imagesetter	photographic imagesetter
photographic-other	other photographic
ion-deposition	ion deposition
E-beam	E-beam
typesetter	typesetter

6.4.5 printer-state (type1Enum)

This attribute identifies the current state of the printer and shall be set by the Printer. The protocol support all values for printer states, however a Printer shall only generate the printer states which are appropriate for the particular implementation.

The following standard values are defined:

unknown	The printer state is not known, or is indeterminate, or is not returned by the operation
idle	The printer is ready to accept jobs, but none have been scheduled on it.
printing	The printer is currently printing a job
needs-attention	The printer needs human attention (no special skills required). This state typically includes adding paper, clearing a jam, changing the medium, etc.
paused	The operator has (temporarily) paused the printer, by means outside the scope of IPP V1.0.
shutdown	The printer has been taken out of service, (for a long time), whether for repairs or others reasons. The printer's message generic attribute may be used to record a reason and estimated time for return to service
job-start-wait	The currently processing job was started with the job-start-wait attribute set, and is awaiting operator intervention or time-out.

job-end-wait The currently processing job was started with the job-end-wait attribute set, and is awaiting operator intervention or time-out.

job-password-wait The currently processing job was started with the job-password attribute set, and is awaiting the operator or user to enter the password supplied by the job-password attribute.

needs-key-operator The printer needs the attention of a key operator. Key operator functions are printer-specific, but typically include adding toner or developer, or attending to a hardware fault.

connecting-to-printer The server has scheduled a job on the printer and is in the process of connecting to a shared network printer (and may not be able to actually start printing the job for an arbitrarily long time depending on the usage of the printer by other servers).

timed-out The server was able to connect to the printer (or is always connected), but was unable to get a response from the printer in the time specified by the printer's printer-timeout-period attribute.

6.4.6 printer-state-message (string)

This attributes specifies a message that gives further information about the current printer state and shall be set by the Printer.

6.4.7 message (string)

This attribute provides a message from an operator, system administrator or "intelligent" process to indicate to the end user information or status of the printer, such as why it is unavailable or when it is expected to be available.

6.4.8 printer-job-templates (1#urlDefault)

This attribute identifies the URL of each of the Job Templates that this Printer is associated with and the one Job Template this Printer uses as its default for supply job attributes that the client omits. There shall be only one value with the default qualifier. Other Printers can be associated with the same Job Templates.

The syntax is:

```
    url [ ":" default ]
```

6.4.9 locale (type3Locale)

This attribute specifies the locale that the Printer operates in.

The standard values are defined in the section on the job-locale attribute.

6.4.10 notification-events (1#type2Enum)

This attribute specifies the events on whose occurrence the Printer should notify those addresses specified by the notification-addresses attribute.

If the attribute is unspecified, the Printer does not perform notification, though the Printer still checks the job's notification-events attribute.

In this attribute, job-problem and printer-problem have the same meaning.

The standard values are defined in the section on the job's notification-events attribute.

NOTE - This attribute is intended to notify operators, not end-users.

6.4.11 notification-addresses (1#url)

This attribute specifies the method and addresses to which the Printer should send messages when events specified by the notification-events attribute occur.

If the attribute is unspecified, the Printer does not perform notification, though the Printer still checks the job's notification-events attribute.

NOTE - This attribute is intended to notify operators, not end-users.

6.4.12 end-user-acl (1#name)

This attribute specifies the end users who are allowed to print on the Printer.

If the attribute is unspecified, the Printer allows anyone to print.

6.4.13 maximum-printer-speed (positiveIntegerUnits)

This attribute indicates the maximum printer speed of the Printer in units of pages per minute, impressions per minute, lines per minute, and characters per minute. A job cannot control a Printer's speed, but a Printer Browser can use printer speed as a criteria.

The standard units are a type2Enum and are: ppm, ipm, spm, lpm, cps.

6.4.14 fonts-substitutions (1#stringPair)

This attribute specifies an appropriate substitute for a font that is advertised as supported in the fonts-supported attribute, even though the Printer doesn't actually have the font available.

This attribute consists of a set of font pairs: a font name and the font to use instead.

If this attribute is unspecified, the Printer does not perform any font substitutions.

6.4.15 fonts-supported (1#stringState)

This attribute identifies the font resources supported by this printer and indicates the state of readiness for each font.

The standard names are defined in the section on default-font.

Each item in the list contains the pair consisting of a font name and a state indicating the font's readiness state.

6.4.16 media-supported (1#nameState)

This attribute identifies the media, media-sizes, input trays, and electronic forms supported by this printer, and indicates the state of readiness for each medium resource.

The standard names are defined in the section on the section on the medium-select.

Standard states are: not-ready, on-order, and special-order. The omission of a state shall indicate that the medium is ready, i.e., can be used without human intervention..

6.4.17 document-formats-supported (1#type2FormatState)

This attribute identifies the document-formats, including the document-format-versions, supported by the Printer. This set includes both the formats that are native to the Printer and those formats that the Printer can translate to one that is native to the Printer. From the client's point of view, this set contains all formats in which documents can be submitted to this Printer.

Proprietary document format identifiers, and versions are assigned by the owners of those formats.

The state of readiness for each format is also included, though all formats should normally always be ready.

6.4.18 numbers-up-supported (1#type3EnumState)

This attribute identifies the number-up values supported by this printer..

The state of readiness for each number-up value is also included, though all number-up conversions should always be ready.

6.4.19 finishings-supported (1#type2EnumState)

This attribute identifies the finishing operations supported by this Printer and states of readiness for each finishing.

The standard finishing objects are defined in the section on the finishing Job attribute.

6.4.20 sides-supported (1#type2EnumState)

This attribute indicates the values of the sides attribute supported by this printer and the states of readiness of each value.

The standard values are defined in the section on the sides attribute.

6.4.21 print-qualities-supported (1#type2EnumState)

This attribute indicates the values of the printer-quality attribute supported by this printer and the states of readiness for each print-quality value.

The standard values are defined in the printer-quality attribute.

6.4.22 printer-resolutions-supported (1#positiveIntegerCrossState)

This attribute indicates the values of the printer-resolution-select attribute supported by this printer and their states of readiness.

The state of readiness for each printer resolution is also included, though normally all printer-resolutions should always be ready.

The syntax is discussed in the section on the printer-resolution-select attribute.

6.4.23 code-sets-supported (1#type3EnumState)

This attribute indicates the values of the default-code-set attribute supported by this printer and the states of readiness for each code-set.

The standard values are defined in the default-code-set attribute.

6.4.24 off-peak-times-supported (1#type3EnumState)

This attribute indicates the values of the job-print-off-peak attribute supported by this printer and the states of readiness for each value.

If this attribute is unspecified, then the Printer has no off-peak periods.

The standard values are defined in the section on the job-print-off-peak Job attribute.

Note: this document does not define how an administrator associates the off-peak names with actual time periods.

6.4.25 events-supported (1#type2EnumState)

This attribute indicates the values of the job and printer notification-events attribute supported by this Printer and the states of readiness for each value.

If this attribute is unspecified, then the Printer does not support notification.

The standard values are defined in the section on the notification-events attribute.

6.4.26 locales-supported (1#type3LocaleState)

This attribute indicates the values of the job-locale attribute supported by this Printer and the states of readiness for each value.

The standard values are defined in the section on the job-locale attribute.

6.4.27 job-sheets-supported (1#type3EnumState)

This attribute identifies the job-sheet values supported by this printer, and the state of readiness for each job-sheet.

To allow no job sheets, the system administrator shall include the value "none" as a value for this attribute. The client specifies that there are no job sheets by using the value "none" as the value of the job-sheets attribute.

If the job-sheets attribute is not specified or contains a value which the Printer does not support, then the server shall select from among the values of this attribute. The server shall not select the value "none" unless it is the only value specified for the job-sheets-supported attribute.

NOTE – When the client supplies a value other than "none", it is preferable for the server to produce some job jobsheet, even if not the desired one, rather than produce none at all or reject the job.

6.4.28 maximum-copies (positiveInteger)

This attribute indicates the maximum number of copies of a document that can be rendered by this printer in a single print-job.

If the attribute is unspecified, there is no limit on the maximum number of copies for this Printer.

6.4.29 maximum-job-octets (positiveInteger)

This attribute indicates that the Printer shall accept a job only if its size in octets is less that the value specified by this attribute.

If the attribute is unspecified, there is no limit on the size of a job in octets.

6.4.30 maximum-impressions (positiveInteger)

This attribute indicates that the Printer shall accept a job only if its size in impression is less that the value specified by this attribute.

If the attribute is unspecified, there is no limit on the size of a job in impressions.

6.4.31 maximum-media-sheets (positiveInteger)

This attribute indicates that the Printer shall accept a job only if its size in media-sheets is less that the value specified by this attribute.

If the attribute is unspecified, there is no limit on the size of a job in media-sheets.

6.4.32 maximum-job-retention-period (deltaTime)

This attribute indicates that when the Printer accepts a job, the retention period must not exceed the value of this attribute. Otherwise, the Printer sets the job's retention-period to the value of this attribute.

If this attribute is unspecified, then the Printer places no limit on the retention time.

6.4.33 maximum-end-user-priority (type1Enum)

This attribute indicates that when the Printer accepts a job, the job-priority must not exceed the value of this attribute. Otherwise, the Printer sets the job's job-priority to the value of this attribute.

If this attribute is unspecified, then the Printer places no limit on the job-priority.

The standard values are defined in the section on the job-priority attribute.

6.4.34 queued-job-count (cardinal)

This attribute contains a count of the number of jobs that are either pending and/or processing and shall be set by the Printer.

6.4.35 scheduling-algorithm (type3Enum)

This attribute indicates the current scheduling algorithm for this Printer. Standard values are: "none", "smallest-job-first", "time-received".

6.5 Job Templates

The attributes for a Job Template can be any of the Job object attributes defined in the sections:

Job Sheet Attributes
Notification Attributes
Job Scheduling Attributes
   (except job-print-after)
Job Production Attributes
   (except page-select)
Attributes for Conversion of Text and HTML Files

6.6 Conformance

A conforming implementation shall implement all operations, objects and attributes defined in this document.

Also, for the core set of attributes listed in this specification, it is not required that a conforming server support all (standard) values of all supported attributes. For example, it is not required that a printer implement all finishing methods indicated by the standard values.

The explicit requirement of the term "supported", with respect to one of the attributes that deal with printer functions or resources, is that the server shall recognize the attribute and those values that are supported, and shall be able to respond to a query about which values that printer does, in fact, support.

IPP is explicitly designed to be extensible. Additional attributes can be proposed to be registered by going through the type 2 enum process which will register their specification after approval with IANA. In addition specific implementation instances may support not only the basic protocol as defined in this specification, but may add vendor-specific private extensions by prefixing attribute-names with their company name registered with IANA for use in domains. See attribute syntax section. However, such private extensions shall not duplicate attribute semantics already in this specification.

7. Security Considerations

This protocol does not identify any new authentication mechanisms. The authentication mechanisms built into HTTP (such as SSL and SHTTP) are recommended.

This protocol does define a simple authorization mechanism by introducing the "end-user-acl" attribute as part of the Printer object. This ACL attribute is a multi-valued list of all of the authenticated names of end-users. This protocol does not specify what the domain is for names in this ACL attribute.

Issue: Will it always be possible for a Printer to obtain a meaningful authenticated name that the Printer can match against the end-user-acl, or will some other mechanism be necessary, such as a password?

8. References

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

[2] Berners-Lee, T, Fielding, R., and Nielsen, H., "Hypertext Transfer Protocol - HTTP/1.0", RFC 1945, August 1995.

[3] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", RFC 822, August 1982.

[4] Postel, J., "Instructions to RFC Authors", RFC 1543, October 1993.

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

[6] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August 1995.

[7] Kirk, M. (editor), POSIX System Administration - Part 4: Printing Interfaces, POSIX 1387.4 D8, 1994.

[8] Borenstein, N., and Freed, N., "MIME (Multi-purpose Internet Mail Extensions) Part One: Mechanism for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, September, 1993.

[9] Braden, S., "Requirements for Internet Hosts - Application and Support", RFC 1123, October, 1989,

[10] McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.

[11] Berners-Lee, T., Masinter, L ., McCahill, M. , "Uniform Resource Locators (URL)", RFC 1738, December, 1994.

9. Author's Address

Scott A. Isaacson
Novell, Inc.
122 E 1700 S
Provo, UT 84606

Phone: 801-861-7366
Fax: 801-861-4025
EMail: scott_isaacson@novell.com

Tom Hastings
Xerox Corporation
701 S. Aviation Blvd.
El Segundo, CA 90245

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

Robert Herriot
Sun Microsystems Inc.
2550 Garcia Ave., MPK-17
Mountain View, CA 94043

Phone: 415-786-8995
Fax: 415-786-7077
Email: robert.herriot@eng.sun.com

Roger deBry
HUC/003G
IBM Corporation
P.O. Box 1900
Boulder, CO 80301-9191

Phone: (303) 924-4080
Fax: (303) 924-9889
Email: debry@vnet.ibm.com

Other Participants:

Devon Taylor, Novell, Inc.
Mike MacKay, Novell, Inc.
Peter Zehler, Xerox, Corp.
Keith Carter, IBM Corporation
Carl-Uno Manros, Xerox, Corp.
Don Wright - Lexmark
Steve Gebert - IBM
Ray Lutz - Cognisys
Mabry Dozier - QMS
Lee Ferrel - Canon
Hiro Sato - Canon
Pat Nogay - IBM
Jim Walker - Dazel
Jay Martin - Underscore
Bill Wagner - DPI
Stan McConnell - Xerox
Bob Setterbo - Adobe
Randy Turner - Sharp
Rob Whittle - Novell
Ron Bergman - Data Products
Lloyd Young - Lexmark
Andy Davidson - Tektronix
Rick Landau - Digital
David Kellerman - Northlake Software
David Roach - Unisys
Mike Timperman - Lexmark
Chris Wellens - Interworking Labs
Pete Loya - HP
Bob Pentecost - HP
Harry Lewis - IBM
William Wagner - Digital Products
Atsushi Yuki - Kyocera
Rob Rhoads - Intel
Jeff Barnett - IBM
Jim Walker - Dazel
Jeff Copeland - QMS
Chuck Adams - Tektronix

10. Appendix A: Sample IPP Operations

The following examples illustrate typical flows using the IPP protocol. In these examples, the IPP Printer object named "printer-1" is located at the node identified by the DNS name "some.domain.com". A Job Template has been defined for printer-1 which establishes the print defaults.

For brevity in the following flows, none of the HTTP headers are shown. CRLF sequences are not shown.

10.1 Querying the printer

```
Client                                  some.domain.com

-------------------------------------------------->
Post http://some.domain.com/printer-1   http/1.0
Get-Attributes  IPP/1.0
   printer-state :
   sides-supported :
   media-supported :
   document-formats-supported :

<--------------------------------------------------
http/1.0  201 "Created" (a response)
   IPP/1.0 xxx "attribute list returned"
   printer-state : idle
   sides-supported : 1-sided
   media-supported : iso-a4-white, iso-b4-white
   document-formats-supported : Postscript/2.0
```

10.2 Print Operation - with print data included

```
Client                                 some.domain.com

-------------------------------------------------->
```

```
Post http://some.domain.com/printer-1  http/1.0
   Print IPP/1.0
   Print-Job-Object Header
      job-name : My Job
      medium : iso-a4-white
      notification-events : Job-completion
      notification-address : joe@pc.domain.com
   Document Header
      document-name : Letter to Mom
   Document-Content Header (content type = Postscript/2.0)
      <Document in Postscript level 2 format>


<-------------------------------------------------
http/1.0  200 "accepted"
   IPP/1.0 xxx "print job accepted and queued"
      job-identifier : some.domain.com/printer-1/0037
      current-job-state : pending
      printer-state : needs-sttention
```

10.3 Print Operation - with no data included

```
Client                                   some.domain.com

------------------------------------------------->
Post http://some.domain.com/printer-1  http/1.0
   Print IPP/1.0
   Print-Job-Object Header
      job-name : My Job
      medium : iso-a4-white
      notification-events : Job-completion
      notification-address : joe@some.domain.com
   Document Header
      document-name : Letter to Mom
      document-URL : joe@pc.domain.com/Docs/To-mom.ps

<------------------------------------------------
http/1.0  200 "accepted"
   IPP/1.0 xxx "print job accepted and queued"
      job-identifier : some.domain.com/printer-1/0037
      current-job-state : pending
      printer-state : processing
```

10.4 Querying the state of the job

In this example, no attributes are specified, so all job attributes are returned.

```
Client                                some.domain.com

------------------------------------------------->
Post http://some.domain.com/printer-1/0037  http/1.0
   Get-Attributes  IPP/1.0
```

```
<-------------------------------------------------
http/1.0 201 "Created" (a response)
   IPP/1.0  xxx "atribute list returned"
   job-Name : My Job
   job-Originator : Joe@some.domain.com
   job-originating-host : pc.domain.com
   notification-address : joe@pc.domain.com
   job-locale : xx:xx:xx
   current-job-status : printing
   submission-time : 1996 Nov 22 1214
   media-sheets-completed : 2
```

10.5 Canceling a Job

```
Client                                some.domain.com

-------------------------------------------------->
Post: http://some.domain.com/printer-1/0037
   Cancel-Job  IPP/1.0



<-----------------------------------------------
http/1.0  200 "okay"
Current-job-state : terminating
```

10.6 Listing jobs on a Printer

List jobs on printer-1, only return job sizes. Jobs are returned in the order they are scheduled for printing. A Job-identifier attribute precedes the attributes returned for each job to delimit job boundaries.

```
Client                                some.domain.com

-------------------------------------------------->
Post http/1.0 some.domain.com/printer-1
   Get-Jobs  IPP/1.0
      total-job-octets :
```

```
<-------------------------------------------------
http/1.0  201 "Created" (a response)
   IPP/1.0 xxx "created an attribute list"
   job-identifier : 0033
   total-job-octets : 4567
   job-identifier : 0034
   total-job-octets : 12345
   job-identifier : 0035
   total-job-octets : 12356
```