Internet-Draft

S. Isaacson
Novell, Inc.
D. Taylor
Novell, Inc.
M. MacKay
Novell, Inc.
P. Zehler
Xerox Corporation
T. Hastings
Xerox Corporation
C. Manros
Xerox Corporation
R. Herriot
Sun Microsystems
November 14, 1996

Internet Printing Protocol - IPP/1.0
Version 0.91, November 14, 1996

NOTE: This document still has many rough spots which will need further editing. At this time, the reader should read it for major concepts.

NOTE: This version (Ver. 0.91) contains changes (over Ver. 0.9) in the following sections: Abstract - minor wording changes; Intro. - minor wording changes; Section 2: Simplified, removed redundant paragraphs; Added Section 3: Old section 2 paragraphs about IPP objects, fixed old section 2.2.2 on Jobs and 2.2.4on Job Templates;Added Section 4: on naming and directory schemas; Deleted Old section 3 and 4; Section 5: Fixed User Operations (including deletion of empty table rows), Section 6: Major modifications on Object Attributes; All other sections: minor changes, some fixes to security section.

Status of this Memo

This document is an Internet -Draft. Internet -Drafts are working documents of the Internet Engineering Task Force (IETF), its

areas, and its working groups. Note that other groups may also distribute working documents as Internet -Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet -Drafts as reference material or to cite them other than as ``work in progress.''

To learn the current status of any Internet -Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet -Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

This Internet-Draft specifies an Internet Printing Protocol (IPP). 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 initially covers only end user operations on basic print service objects. Future versions of the protocol will cover operator and administrator operations. Authentication is relized 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 devices with access control. Also, the Cancel Job operation requires some authentication and authorization so that jobs can only be canceled by authorized end-users. Extended monitoring and management is possible through other protocols such as the SNMP Printer MIB [1].. 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 deals initially 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.

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 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: Printer 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 send 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 three simple objects:

- Printers (section xxx)
- Print Jobs (section xxx)
- Job Templates (section xxx)

These objects are not encapsulations of both data and behavior as in other object oriented models, but are simple collections of attribute/value pairs. [We may try to fix this in our new design, but it's not high priority.]

Clients interact with these using the following operations:

- Print (section xxx)
- Cancel Job (section xxx)
- Get Attributes(section xxx)
- Get Jobs (section xxx)

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.

Some examples of an IPP Printer object include:

- An output device, with a 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
- A print server with one or more downstream print servers and/or output devices where the upstream print server supports IPP

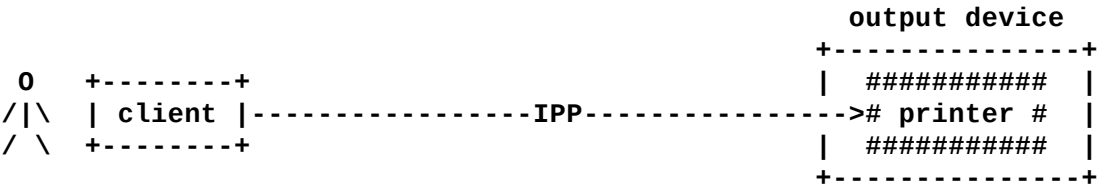
See the following figures for some examples on how to view IPP Printer objects on top of other printing system models:

Legend:

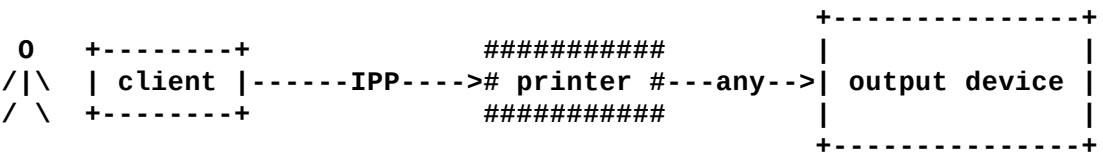
indicates an IPP printer object which is either embedded in an output device or is hosted in a server. An IPP printer object may or may not queue/spool.

any indicates any network protocol or direct connect, including IPP

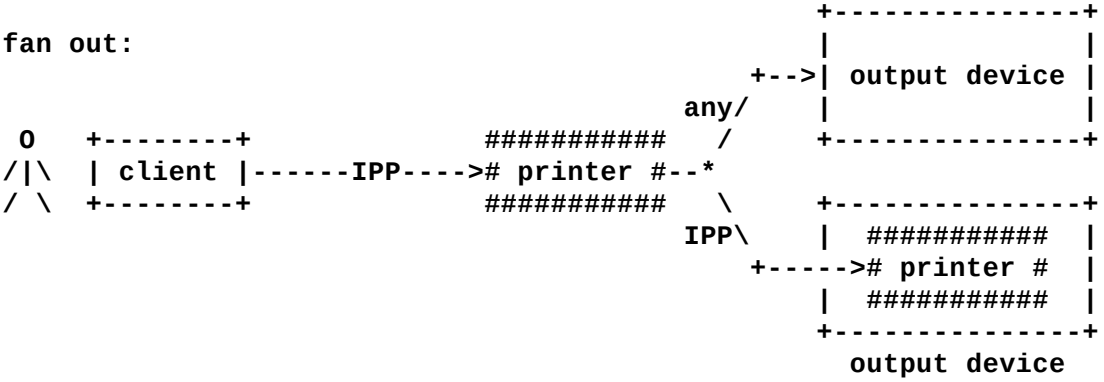
embedded printer:



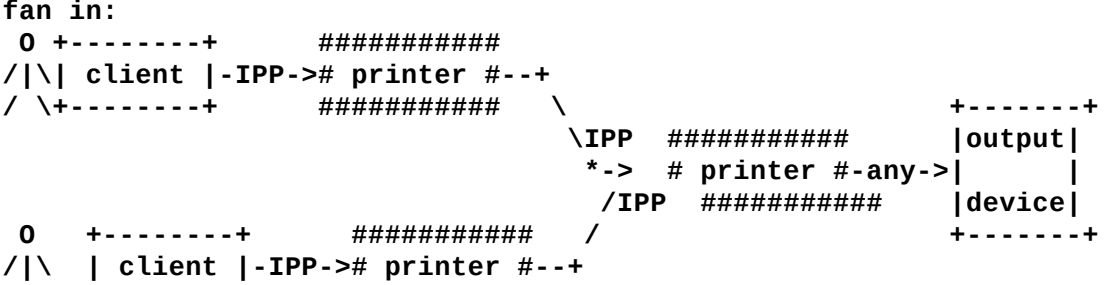
hosted printer:



fan out:



fan in:



3.2 Job

A Job object is used to model a job. A job can consist of 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 provide information to

- identify the print job(section xxx)
- assist in selecting the Printer (section xxx)
- report job status (section xxx)
- assist in scheduling and processing (section xxx)
- describe the documents in the job (section xxx)
- produce the document (section xxx)

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 a client references to initialize a newly created job object. Attributes which are sent along with the job at the time the job is submitted override the attributes in the Job Template object.

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. A Job object contains one or more Documents. A Printer object is associated with one 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. In the object-oriented model, these are the globally unique object references which are created by factories or constructors.

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://1.2.3.4:3042/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 3042 for the HTTP server, and "printer-2" is the name of the Printer.

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 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 locations, but underneath they represent the same Printer object but using two different Job Template default attribute sets.

- The Job Template can be exposed along with the name of the Printer object directly in the URL as in: http://some.domain.com/hr-printer/resumes. In this case there is a "resumes" Job Template associated with the "hr-printer" Printer.

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.

4.2.1 Status

[Such a dynamic value seem like it could be a problem in some name service entries.]

The printer status field in the directory entry is really a "summary" attribute of the true printer state. The following mapping takes place between the Printer Status attribute in the directory entry and the printer-state attribute in the Printer object:

"Not Connected"
STATE_NOT_CONNECTED
STATE_PAUSED_NOT_CONNECTED
"Shutdown"
STATE_SHUTDOWN
"Active"
STATE_IDLE
STATE_PAUSED
STATE_PRINTING
"Stopped"
STATE_STOPPED
STATE_PAUSED_STOPPED

Even though the Printer may not be up and running, the directory entry still exists in the directory. In this case, the directory entry represents the fact that it may begin running at some future time.

4.2.2 Resolution

This is a single valued, maximum resolution in either the horizontal or vertical direction of the print device in dpi.

4.2.3 Color Supported

This is a BOOLEAN for either yes, color printing is supported, or no color printing is not supported.

4.2.4 Maximum Speed

This is the maximum speed of the printer in the units defined in Maximum Speed Units

4.2.5 Maximum Speed Units

This is the units of the maximum speed rating of the print device. This can be: pages per minute, sheets per minutes, characters per second, etc.

4.2.6 Plug and Play Device Id

This attribute can be used for automatic driver download and other automatic configuration tasks.

4.2.7 Model

This is a simple text string defined by the manufacturer.

4.2.8 Manufacturer

This is a simple text string defined by the manufacturer. There is no registration, and there is a possibility of overlap, but the goal is to keep this simple, not too complex.

4.2.9 Type

This is the printing mechanism of the print device: laser, ink jet, thermal, etc.

4.2.10 PDLs Supported

This is a list of all of the page description languages (PDLs) that the printer and/or its interpreter(s) support.

4.2.11 Sides Supported

This is either a 1 or a 2 to indicate the maximum number of sides on which the printer can automatically print.

5. IPP Operations

IPP defines the following end user operations:

The following symbols are used in the tables below:

P perform the operation directly
PF perform the operation; forward to Output Device sometimes
UA unsupported in an Output Device unless it supports queuing
U unsupported operation

Operation	Print Server	Output Device
Print	PF	P
Cancel Job	PF	P
Get Attributes	PF	P
Get Jobs	PF	P

5.1 Print Operation

When an end-user uses GUI to submit a job, the GUI client gets an HTML form from the default printer. If the end-user changes the selected printer, the GUI client gets the HTML form from that printer. The HTML form comes with the values supported by the printer and it is initialized by the values from the job template associated with the named printer.

[Further work needs to done to define the above concept.]

5.1.1 Print Request

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

Printer Name	Note: I don't think that this is needed
Job and Document Attributes	A set of Job object and Document attributes as defined in section xxx
Document Contents	Note: What if there are multiple documents and each has a different size? How does this map on top of the HTTP header that has one size? Does it require multiple HTTP operations?

5.1.2 Print Response

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

Job Id	Used for all other operations on this Job.
Job Status	Job state information
Printer State	Optional Printer state information
Message	Optional message Note: Is this needed?
Errors	Optional Error Information

5.2 Cancel Job Operation

5.2.1 Cancel Job Request

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

Job Id	The identifier of the job to be canceled
Document Number	Optional document number of the document to
Message	Optional message to the operator.

5.2.2 Cancel Job Reponse

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

Job Status	Optional Job status information
Errors	Optional Error Information

5.3 Get Attributes Operations

5.3.1 Get Attributes Request

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

Selector	A Job Id or Printer Name URL
Requested Attributes	A set of attributes in which the requestor is interested

5.3.2 Get Attributes Response

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

Result Attributes	The requested attributes of the object
Errors	Optional error information

5.4 Get Jobs Operation

Get Jobs Request
5.4.1 Get Jobs Request

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

Filtering	A lightweight filtering mechanism, such as all jobs versus a particular user’s jobs.
Requested Attributes	A set of job attributes in which the requestor is interested

5.4.2 Get Jobs ResponseGet Jobs Response

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

Result Attributes	Attribute set containing the returned results.
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

NOTE: This is what Tom has submitted:

Each attribute shall be in one of the following data syntaxes:

```
644       string       - arbitrary ASCII strings, no control characters,
645                      except <SPACE>.
646       string pair  - strings separated by ":"
647       name         - arbitrary ASCII strings, no control characters, and
648                      no <SPACE> characters.
649       type 1 enum  - standard names, must revise the standard to add
650                      a new name.  No private names are allowed.
651       type 2 enum  - standard names, but an implementor can add new
652                      by proposing them to the PWG for registration
653                      (or an IANA-appointed registry advisor after the
654                      PWG is no longer certified) anytime.  IANA keeps
655                      the registry.
656                      Implementors can add private (un-registered)
657                      with a suitable distinguishing prefix, such as
658                      -xxx- where xxx is the company name
659                      registered with IANA.
660       type 3 enum  - standard names, but an implementor can add new
661                      names by submitting a registration request directly
662                      to IANA, no PWG or IANA-appointed registry advisor
663                      review is required.
664                      Implementors can add private (un-registered) names
665                      with a suitable distinguishing prefix, such as
666                      -xxx- where xxx is the company name registered with
667                      IANA.
668       type 3 pair  - two type 3 enum names separated by ":".
669       cardinal     - 0 .. n represented as ASCII digits
670       ordinal      - 1 .. n represented as ASCII digits
671       ordinal pair - two ordinals separated by ":"
672       boolean      - tokens: yes, y, true, or t and no, n, false, or f.
673       date/time    - date/time in ??? format
674       url          - Universal Resource Locator
675       octet string - arbitrary binary octets
676       string units - ordinal followed by type 2 enum units
677
678
```

679 NOTE: This is what Bob has 680 submitted:

```
681       The sections below reference the following syntax items:

682       string: TBD
683       stringPair: string “:” string
684       stringState: string state
685       name: TBD
686       URL: TBD
687       dateTime: TBD
688       deltaTime: [hours “:”] minutes
689       cardinal: TBD
690       type1Enum: TBD
691       type2Enum: TBD
692       type3Enum: TBD
693       type2EnumState: type2Enum state
694       type3EnumState: type3Enum state
695       state: TBD
696       Boolean: TBD
697       positiveInteger: TBD
```

```
698       positiveIntegerCross: positiveInteger [ “x” positiveInteger ]
699       positiveIntegerCrossState: positiveIntegerCross state
700       positiveIntegerRange: positiveInteger  “:” positiveInteger
701       positiveIntegerUnits: positiveInteger units
702       positiveIntegerState: positiveInteger state
703       units: “ppm” | “ipm” | “spm” | “cps” | “lpm”
704       type3Locale: type3Country “:” type3Language “:” type3CodeSet
705       type3Country: type3Enum
706       type3Language: type3Enum
707       type3CodeSet: type3Enum
708       type2Format: name [ “/” version ]
709       version: name
710       type3LocaleState: type3Locale state
711
712    6.2 Job Attributes
```

713 A job object contains a set of job attributes and one or more
714 documents. A client shall create a job and send it to a server
715 using the Print operation. A client may use a job template
716 associated with the selected printer in order to initialize the
717 job.

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

```
720
721
722    6.2.1 Job Informational Attributes Set by a Client
723
```

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

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

```
728    6.2.1.1 job-name (string)
729
```

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

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

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

```
736    6.2.2 Job Informational Attributes Set by a Printer
737
```

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

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

6.2.2.1 job-identifier (string)

This attribute provides the job -identifier for this job on the Printer. The Printer shall generate a job -identifier value that is unique on that Printer, but need not be unique across the distributed environment.

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 notification-address (name)

This address specifies the email address of the client. The client specifies this attribute in the operation-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 user has registered an interest in or when certain other events occur, such as Cancel-Job.

Note: The only type of notification is email.

ISSUE: can the email address be inferred with job-originator and the originating-host.

6.2.2.5 job-locale (type3Locale)

This attribute identifies the locale of the job. 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.

The type3EnumTrip consists of 3 colon separated type 3 enums. The first is the country. The second is the language. The third is the code-set.

ISSUE: is there a more standard syntax for locale?

6.2.3 Printer Selection Attributes

The client shall specify this attribute to select a particular Printer.

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

Issue: this attribute may be implicit in the specified URL in the Print operation.

6.2.3.1 printer-name-requested (URL)

This attribute identifies the printer that the client requests for printing the job.

6.2.4 Job Status Attributes

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.

6.2.4.1 current-job-state (type1Enum)

This attribute identifies the current state of the job with the following values: unknown, pre-processing, pending, processing, printing, held, terminating, retained, completed.

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

6.2.4.2 printer-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 does 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..

ISSUE: Is this attribute appropriate for a model in which we are hiding the downstream Printer.The printers -assigned value shall not be the same as the printer requested by the user.

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

6.2.4.3 submission-time (dateTime)

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

6.2.4.4 job-message-from-administrator (string)

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

6.2.4.5 completion-time (dateTime)

This attribute indicates the time at which this job completed. This time is useful for jobs which are retained after printing.

6.2.4.6 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. It is valid for the printer to set the value of the job-state-reasons attribute to the empty set.

6.2.4.7 The following standard values are defined: documents -needed, job-hold-set, job-print-after-specified, job-off-peak-specified, required-resources-not-ready, successful completion, completed-with-warnings, completed -with-errors, cancelled -by-user, cancelled-by-operator, aborted -by-system, logfile -pending, logfile-transferringimpressions-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.

6.2.4.8 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.

6.2.5 Job Sheet Attributes

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

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

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' means that the Printer shall print the job sheets defined by an administrator.

6.2.6 Notification Attributes

The client shall specify these attributes to indicate events that the client is interested in.

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

6.2.6.1 notification-events (#type1Enum)

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

This attribute will support three events classes: job-completion, job-problems and printer-problems., If attribute contains no values, then the client is requesting no notification.

This attribute will support only one delivery method, namely email. The attribute notification-address specifies the email address.

If this attribute contains the event job-completion, the Printer shall notify the client when the job containing this attribute completes.

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

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

ISSUE: is there a problem with an attribute with an empty value being different from no attribute. Otherwise, we need a special value of 'none'. 'none', unlike other values should not be combined with other values. This particular case does not seem like a case where an administrator wants to prevent the use of 'none', so empty seems like a good solution.

ISSUE: Email is quite deficient for timely notification to an end-user who receives a lot of email, but there are no other choices. The internet community needs to solve this problem, perhaps with an extremely-urgent email.

6.2.7 Job Scheduling Attributes

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.

job-hold (Boolean) This attribute specifies whether the print -job is a candidate for scheduling for printing or not, when the Printer would otherwise place the job in the pending or processing states

When the value is FALSE, the Printer shall not hold the job from being scheduled for printing, unless there are other reasons (see the current -job-state and the job -state-reasons job -status attributes).

When the value is TRUE, the Printer shall place the job in the held state and add the job -hold-set value to the job's job-state-reasons attribute and shall not schedule the print -job for printing. If the job enters the held state because its job-hold attribute was TRUE, a client shall reset the job's job-hold attribute to FALSE by me ans of the ModifyJob operation before the printer can schedule the job for printing. When the value is set to FALSE as a result of the ModifyJob operation, the printer shall remove the job -hold-set value from the job-state-reasons 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 on printer(s).

ISSUE: the above few sentences assume a ModifyJob operation, which is not in version 1.0. But without this operation, the job-hold operation is not very useful. **Perhaps we should remove job-hold.**

6.2.7.1 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, medium, 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 medium or low priority, and a Printer shall print all such jobs with a medium priority before printing those with a low priority.

If the client does not specify this attribute, the Printer assumes that the user places no constraints concerning priority on the scheduling of the print -job, and it has a priority value of the value of the Printer’s maximum-end-user-priority attribute. If that attribute is unspecified, the Printer uses the value of ‘high’.

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

6.2.7.2 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.

6.2.7.3 job-print-off-peak (type3Enum)

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

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, such as 'night', 'weekend', 'thirdShift'.

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

6.2.7.4 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.

6.2.8 Job Production Attributes

The client shall specify these attributes to affect the rendering, production and finishing of the document.. 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.

After the information from these attributes has been folded into the document data (possibly during a translation process of the document data), these attributes are no longer relevant and can be discarded from a job. Instead, the resource attributes specify the resources needed to print the job as modifieed by the job production attributes.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.The client may also specify document production -instruction attributes in: Get-Attributes and GetJobs.

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 defined: TBD

6.2.8.1 number-up (positiveInteger)

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

In general, only certain numeric values are valid for this attribute, depending upon the Printer implementation to which the print-request is directed. Typical supported values are 2 and 4. If this attribute is unspecified or has a value of 1, then the Printer does not apply any number-up transformation to the pages.

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.ISSUE: should there be a separate attribute to control embellishments, especially for the 1-up case. finishing (type2Enum) This attribute identifies the finishing operation that the Printer should apply to each copy of the printed document. Examples include stapling, saddle -stitching, hole-drilling, binding with tape, etc.
Standard values for this attribute are include:TBD.

6.2.8.2 sides (type2Enum)

This attribute specifies whether the document should be printed in one of three ways: 1-sided (simplex), 2-sided- long-edge (duplex), 2-sided- short-edge (tumble).
The standard values are: 1-sided, 2-sided-long-edge, 2-sided-short-edge.

6.2.8.3 copies (positiveInteger)

This attribute specifies the number of copies of the job to be printed. If this attribute is unspecified, its default value is 1 copy.

6.2.8.4 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 paper feed direction.

6.2.8.5 print-quality (type2Enum)

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

The standard values are: TBD.

6.2.8.6 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 prints all pages in a document.

6.2.8.7 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**, 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, ….

6.2.8.8 files-are-interleaved (Boolean)

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

6.2.9 Attributes for Conversion of Text Files

The client shall specify these attributes to control formatting for text documents or HTML documents. If the client does not specify any of these attributes, a Printer shall uses its own defaults.

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

6.2.9.1 width (cardinalUnits)

This attribute specifies the media width for the document.

6.2.9.2 length (cardinalUnits)

This attribute specifies the media length for the document.

6.2.9.3 left-margin (cardinalUnits

This attribute specifies the left-margin for the document.

6.2.9.4 right-margin (cardinalUnits)

This attribute specifies the right-margin for the document.

6.2.9.5 top-margin (cardinalUnits)

This attribute specifies the top-margin for the document.

6.2.9.6 bottom-margin (cardinalUnits)

This attribute specifies the bottom-margin for the document.

6.2.9.7 repeated-tab-stops (cardinalUnits)

This attribute specifies the tab stops for the document.

6.2.9.8 header-text (string)

This attribute specifies the header text for the document.

6.2.9.9 footer-text (string)

This attribute specifies the footer text for the document.

6.2.9.10 number-pages (Boolean)

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

default-font (string) This attribute specifies the font to use for all text in the document.

6.2.9.11 default-code-set (type3Enum)

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

6.2.9.12 content-orientation (type2Enum)

This attribute specifies the orientation of the document.

The standard values are: landscape or portrait.

6.2.10 Job Resource Attributes

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 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 attributess and their values to the job. 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. 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.

document-format-used (1#type2Format)This attribute identifies the document format needed to print 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

ISSUE: do we want the version to be optional?

1157 X
1158
1159 6.2.10.1 fonts-used (1#string)

1160 This attribute identifies the font resources used in the job.

1161 6.2.10.2 code-sets-used (1#type3Enum)

1162 This attribute identifies the code-sets used in the document. This
1163 attribute is relevant only for files that are not in ASCII, such
1164 as text files and possibly PCL files. PostScript files are always
1165 ASCII. Normally there is at most 1 code-set.

1166 Standard values are defined in the section on default-fonts.

1167 6.2.10.3 media-used (1#type2Enum)

1168 This attribute identifies the media, media-sizes, input-trays or
1169 electronic forms needed to print the job.

1170 Standard values for this attribute are defined in the section on
1171 medium-select.

1172
1173 6.2.10.4 sides-used (type2Enum)

1174 This attribute specifies whether a job needs one-sided, two-sided-
1175 long-edge, or two-sided-short-edge printing.

1176 Standard values for this attribute are defined in the section on
1177 sides.

1178
1179 6.2.10.5 print-quality-used (type2Enum)

1180 This attribute specifies what print quality the job needs.

1181
1182 Standard values for this attribute are defined in the section on
1183 print-quality.

1184 6.2.10.6 finishing-used (type2Enum)

1185 This attribute specifies what finishing the job needs.

1186 Standard values for this attribute are defined in the section on
1187 finishing.

1188 6.2.10.7 printer-resolution-used (positiveIntegerCrossState)

1189 This attribute specifies what resolution the job needs.

1190 6.2.10.8 total-job-octets (positiveInteger)

1191 This attribute specifies the total size of the job in octets. This
1192 attribute is the first of three that a translation program can use
1193 to specify the size of a job.

6.2.10.9 job-impression-count (positiveInteger)

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

6.2.10.10 job-media-sheet-count (positiveInteger)

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

6.2.11 Number of Documents

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

The client shall specify this attribute in Print and may specify this attribute in: Get-Attributes and Get-Jobs.

6.2.11.1 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.

6.2.12 Document Data

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-path in Print.

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

6.2.12.1 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.

ISSUE: do we want the version to be optional?

6.2.12.2 document-name (name)

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

6.2.12.3 document-URL (name)

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 and document-path shall be unspecified.

6.2.12.4 document-content (octetString)

This attribute contains the actual contents of the document.

If this attribute is specified, then document-path and 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 or document-path attribute. A Get-Attribute or Get-Jobs operation shall always find that this attribute is unspecified.

6.2.12.5 document-path (name)

This attribute contains a path which references a file containing the document.

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

This attribute shall not be used during the transmission of the Print operation over a network. It is intended to reference the file when document data is on the printer.

ISSUE: is this attribute necessary or is document-URL sufficient?

6.3 Operation Attributes

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 messages in the result of the operation or in errors returned by the operation.

The type3EnumTrip consists of 3 colon separated type 3 enums. The first is the country. The second is the language. The third is the code-set.

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 (name)

This attribute identifies the email-address of the client. The Printer uses this attribute to determine the email address for any notifications that occur in the Printer.

ISSUE: can this address be determined from the next two attributes: operation-user-name and operation-host-name?

6.3.3 operation-user-name (name)

This attribute identifies the most authenticated user name that the client can supply. This name identifies the 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

A printer object may be realized in either a Print Server or Output Device .

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 contains 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 user/client 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 a user/client can send a print -job direc tly to an Output Device .

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.

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 text (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-types (type2Enum)

This attribute identifies the marking technology of the printer.

The value for this attribute are the descriptive names specified by ISO DPA. These values are: other, unknown. electrophotographic-LED, electrophotographic-laser, electrophotographic-other, impact-moving-head-dot-matrix-9-pin, impact-moving-head-dot-matrix-24-pin, impact-moving-head, dot-matrix-other, impact-moving-head-fully-formed, impact-band, impact-other, inkjet-aqueous, inkjet-solid, inkjet-other, pen, thermal-transfer, thermal-sensitive, thermal-diffusion, thermal-other, electro, erosion, electro-static, photographic-microfiche, photographic-imagesetter, photographic-other, ion-deposition, E-beam, typesetter.

ISSUE: Should they be from the printer MIB instead. In the printer MIB hyphens do not exist. Instead the first letter after a hyphen is upper case.

6.4.5 printer-state (type1Enum)

This attribute identifies the current state of 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, idle, printing, needs -attention, paused, shutdown, job -start -wait, job -end -wait, job -password -wait, needs -key -operator,connecting -to -printer,state -timed -out

6.4.6 printer-state-message (string)

This attributes specifies a message that gives further information about the current printer state. .

6.4.7 message (string)

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

6.4.8 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.

6.4.9 notification-events (#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 or empty, the Printer does not perform notification, though the Printer still checks the jobs’ 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.

6.4.10 notification-addresses (#name)

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

If the attribute is unspecified or empty, the Printer does not perform notification, though the Printer still checks the jobs’ notification-events attribute.

6.4.11 end-user-acl (#name)

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

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

ISSUE: this does not fully solve the internet authorization problem because of authentication issues.

6.4.12 maximum-printer-speed (positiveIntegerUnits)

This attribute indicates the maximum printer speed of the Printer. 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.

6.4.13 fonts-substitutions (#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.

6.4.14 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.

6.4.15 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.

There may be just two states: ready and needs-installing, or there may be a third state: needs-purchasing.

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

6.4.16 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.

6.4.17 numbers-up-supported (1#positiveIntegerState)

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.

6.4.18 finishings-supported (#type2EnumState)

This attribute identifies the per -document 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 attribute.

6.4.19 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.

6.4.20 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.

6.4.21 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.

6.4.22 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.

6.4.23 off-peak-times-supported (#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 attribute.

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

6.4.24 events-supported (#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.

6.4.25 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.

6.4.26 job-sheets-supported (#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 attribu te 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 – 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.

6.4.27 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 or has a value of 0, there is no limit on the maximum number of copies for this Printer.

6.4.28 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 or has a value of 0, there is no limit on the size of a job in octets.

6.4.29 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 or has a value of 0, there is no limit on the size of a job in impressions.

6.4.30 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 or has a value of 0, there is no limit on the size of a job in media-sheets.

6.4.31 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.

6.4.32 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 time.

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

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 Files

6.6 Conformance

A conforming implementation shall implement all operations, objects and attributes defined in this document.IPP is explicitly designed to be extensible. This means that in addition to the attributes defined in this specification, specific implementation instances may support not only the basic protocol as defined in this specification, but might add vendor specific extentions.

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.

7. Security Considerations

This protocol does not identify any new authentication mechanisms. The authentication mechanisms built into HTTP (such as SSL and HTTPS) 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 specifiy what the domain is for names in this ACL attribute

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", FRC 1123, October, 1989,

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

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

F. Devon Taylor
Novell, Inc.
122 E 1700 S
Provo, UT 84606

Phone: 801-861-7179
Fax: 801 -861 -4025
EMail: devon_taylor@novell.com

Mike MacKay
Novell, Inc.
2180 Fortune Dr.
San Jose, CA 95131

Phone: 408-577-6368
Fax: 408-577-5151
EMail: mmackay@novell.com

Peter Zehler
Xerox Corporation

800 Phillips Rd. M/S 111-01X
Webster, NY 14580

Phone: 716-265-8755
Fax: 716-265-8792
EMail: peter_zehler@wb.xerox.com

Carl-Uno Manros
Xerox Corporation
701 S. Aviation Blvd.
El Segundo, CA 90245

Phone: 310-333-8273
Fax: 310-333-5514
EMail: cmanros@cp10.es.xerox.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

10. Appendix A: Extended Operations

The following symbols are used in the tables below:

P perform the operation directly
PF perform the operation; forward to Output Device sometimes
UA unsupported in an Outp ut Device unless it supports queuing
U unsupported operation

Lower priority end user operations are:

Operation	Print Server	Output Device
- Modify Job	P	UA
- Resubmit Job	P	UA

Management operations are:

Operation	Print Server	Output Device
- Clean Queue	PF	UA
- Disallow Queuing	P	UA
- Allow Queuing	P	UA
- Pause Printing	P	P
- Resume Printing	P	P
- Promote Job	PF	UA
- Shutdown Printer	P	P
- Startup Printer	P	P
- Create Printer	P	U
- Delete Printer	P	U
- Set Attribute	P	P

10.1 Modify Job Operation

10.1.1 Modify Job Argument

The following abstract data types are part of the Modify Job Argument (the attributes that can be modified may be severely restricted):

Job Id	Which job to modify.
	[There are no document attributes to modify.]
Job Attributes	Attribute set for Job attributes. Only replacement is possible; the GUI fetches the value and then sets a new one.
Message	Optional Message.
Common Arguments	

10.1.2 Modify Job Result

The following abstract data types are part of the Modify Job Result:

Modify Status	Modify result attributes.
Errors	Optional Error Information

10.2 Resubmit Job Operation

10.2.1 Resubmit Job Argument **Error! Bookmark not defined.**

The following abstract data types are part of the Resubmit Argument:

	.
Destination Printer Name	Optional name of the destination printer.
Operation	MOVE or COPY
Job Set	A set of jobs to move or copy. Each entry in the set has: Job Id, Document Number, Job attributes, and Document attributes.
Message	Optional Message
Common Arguments	

10.2.2 Resubmit Job Result

The following abstract data types are part of the Resubmit Job Result:

Resubmit Job Set	A set of jobs that were resubmitted. Each element in the set has: Old Job Id, New Job Id, and an attribute set with info about the results of the move or copy.
Errors	Optional Error Information