

Internet-Draft

Roger deBry
IBM Corporation
T. Hastings
Xerox Corporation
R. Herriot
Sun Microsystems
Scott Isaacson
Novell, Inc.
November 22, 1996

Internet Printing Protocol - IPP/1.0
Version 0.93, November 22, 1996

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 "lid-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) 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

53 site policy support access control. Also, the Cancel Job
 54 operation requires some authentication so that jobs can only be
 55 canceled by the end user who submitted the job. Extended
 56 monitoring and management is possible through other protocols such
 57 as the SNMP Printer MIB. In the areas where there are no existing
 58 standards, some proposed and emerging standards are being worked
 59 (management, security, etc.). As these services become more
 60 stable, this document (and hence the protocol) can be updated to
 61 reflect the integration and relationships with these other
 62 standards.

63 Table of Contents

64	1. INTRODUCTION	4
65	2. DISTRIBUTED PRINTING	5
66	2.1 Generic Print System Components	5
67	2.2 IPP Components	6
68	3. IPP OBJECTS	6
69	3.1 Printer	7
70	3.2 Job	8
71	3.3 Job Template	9
72	3.4 Object Relationships	10
73	3.5 Object Identity	10
74	4. NAMING	10
75	4.1 Directory Services	12
76	4.2 Directory Entry Schema	12
77	4.2.1 Name	12
78	4.2.2 Description	13
79	4.2.3 Location	13
80	4.2.4 Maximum Print Quality	13
81	4.2.5 Cost	13
82	4.2.6 Resolution	13
83	4.2.7 Color Supported	13
84	4.2.8 Fonts Supported	13
85	4.2.9 Maximum Speed	14
86	4.2.10 Device Id	14
87	4.2.11 Make and Model	14
88	4.2.12 Marker Type	14
89	4.2.13 Document Formats Supported	14
90	4.2.14 Sides Supported	14
91	4.2.15 Finishings Supported	15
92	4.3 Directory Entries Using LDAP	15

93	5. IPP OPERATIONS	16
94	5.1 HTTP Overview	16
95	5.2 IPP Operation Encoding	17
96	5.2.1 HTTP Request-Header Fields	17
97	5.2.2 HTTP Response-Header Fields	18
98	5.3 The Print Job	18
99	5.3.1 Print Job Object Header	18
100	5.3.2 Document Header	19
101	5.3.3 Document-Content Header	19
102	5.3.4 Job Attributes	19
103	5.3.5 Document Attributes	20
104	5.4 Operation Semantics	20
105	5.4.1 Print Operation	20
106	5.4.2 Cancel Job Operation	21
107	5.4.3 Get Attributes Operation	21
108	5.4.4 Get Jobs Operation	22
109	6. OBJECT ATTRIBUTES	23
110	6.1 Attribute Syntaxes	23
111	6.2 Job Attributes	25
112	6.2.1 Job Informational Attributes (Set by a Client/End User)	26
113	6.2.2 Job Informational Attributes (Set by a Printer)	26
114	6.2.3 Job Status Attributes (Set by Printer)	27
115	6.2.4 Job Sheet Attributes (Set by Client/End User)	31
116	6.2.5 Notification Attributes (Set by a Client/End User)	31
117	6.2.6 Job Scheduling Attributes (Set by Client/End User)	33
118	6.2.7 Job Production Attributes (Set by Client/End User)	34
119	6.2.8 Attributes for Conversion of Text and HTML Files (Set by	
120	Client/End User)	43
121	6.2.9 Job Resource Attributes (Set by the program that produces	
122	or senses the PDL)	45
123	6.2.10 Number of Documents (Set by Printer)	47
124	6.2.11 Document Data (Set by a Client/End User)	48
125	6.3 Operation Attributes (Set by Client)	49
126	6.3.1 operation-locale (type3Locale)	49
127	6.3.2 operation-notification-address (url)	49
128	6.3.3 operation-user-name (name)	49
129	6.3.4 operation-host-name (name)	49
130	6.4 Printer Attributes (Set by the Administrator)	50
131	6.4.1 printer-name (name)	50
132	6.4.2 printer-location (string)	50
133	6.4.3 printer-model (string)	51
134	6.4.4 printer-type (type2Enum)	51
135	6.4.5 printer-state (type1Enum)	51
136	6.4.6 printer-state-message (string)	52
137	6.4.7 message (string)	52
138	6.4.8 printer-job-templates (1#urlDefault)	53
139	6.4.9 locale (type3Locale)	53
140	6.4.10 notification-events (1#type2Enum)	53
141	6.4.11 notification-addresses (1#url)	53

142	6.4.12 end-user-acl (1#name)	54
143	6.4.13 maximum-printer-speed (positiveIntegerUnits)	54
144	6.4.14 fonts-substitutions (1#stringPair)	54
145	6.4.15 fonts-supported (1#stringState)	54
146	6.4.16 media-supported (1#nameState)	54
147	6.4.17 document-formats-supported (1#type2FormatState)	55
148	6.4.18 numbers-up-supported (1#type3EnumState)	55
149	6.4.19 finishings-supported (1#type2EnumState)	55
150	6.4.20 sides-supported (1#type2EnumState)	55
151	6.4.21 print-qualities-supported (1#type2EnumState)	55
152	6.4.22 printer-resolutions-supported	
153	(1#positiveIntegerCrossState)	55
154	6.4.23 code-sets-supported (1#type3EnumState)	56
155	6.4.24 off-peak-times-supported (1#type3EnumState)	56
156	6.4.25 events-supported (1#type2EnumState)	56
157	6.4.26 locales-supported (1#type3LocaleState)	56
158	6.4.27 job-sheets-supported (1#type3EnumState)	56
159	6.4.28 maximum-copies (positiveInteger)	57
160	6.4.29 maximum-job-octets (positiveInteger)	57
161	6.4.30 maximum-impressions (positiveInteger)	57
162	6.4.31 maximum-media-sheets (positiveInteger)	57
163	6.4.32 maximum-job-retention-period (deltaTime)	57
164	6.4.33 maximum-end-user-priority (type1Enum)	58
165	6.4.34 queued-job-count (cardinal)	58
166	6.4.35 scheduling-algorithm (type3Enum)	58
167	6.5 Job Templates	58
168	6.6 Conformance	58
169	7. SECURITY CONSIDERATIONS	59
170	8. REFERENCES	59
171	9. AUTHOR'S ADDRESS	60
172	10. APPENDIX A: SAMPLE IPP OPERATIONS	61
173	10.1 Querying the printer	61
174	10.2 Print Operation - with print data included	61
175	10.3 Print Operation - with no data included	62
176	10.4 Querying the state of the job	62
177	10.5 Canceling a Job	62
178	10.6 Listing jobs on a Printer	63
179		
180	1. Introduction	
181	The Internet Printing Protocol (IPP) is an application level	
182	protocol that can be used for distributed printing on the	
183	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 only with the end user role and will be version 1.0. 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 "~~P~~erprinter". 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

230 provides arguments which add information about the operation
231 requested, and receives results which describe the status and
232 outcome of the operation.

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

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

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

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

257 2.2 IPP Components

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

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

267
268 Clients use the following operations:

- 269 - Print (section 5.4.1)
- 270 - Cancel Job (section 5.4.2)
- 271 - Get Attributes (section 5.4.3)
- 272 - Get Jobs (section 5.4.4)

274 3. IPP Objects

275 This section describes the IPP objects.

276 3.1 Printer

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

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

295 Some examples of configurations containing IPP Printer object
296 include:

- 297 - An output device, with no spooling capabilities, supporting
298 IPP
- 299 - An output device, with a built-in spooler, supporting IPP
- 300 - A print server with one or more associated output devices with
301 the print server supporting IPP.
 - 302 - The associated output devices may or may not be capable of
303 spooling jobs
 - 304 - The associated output devices may or may not support IPP
305

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

Legend:

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

any indicates any network protocol or direct connect, including IPP

embedded printer:

```

                                output device
                                +-----+
0    +-----+
/|\  | client |-----IPP-----># Printer #
/\   +-----+
                                | ##### |
                                +-----+

```

hosted printer:

```

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

```

fan out:

```

fan out:

      0      +-----+
      /|\    | client |-----IPP----># Printer #
      / \    +-----+

                                #####
                                any/
                                /
                                +-----+
                                | output device |
                                +-----+

                                #####
                                any\
                                \
                                +-----+
                                | output device |
                                +-----+

```

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 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. A Job Template consists of a subset of Job object attributes defined by the Administrator using means outside the scope of IPP version 1.0. 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.

The client may use 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. 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.

ISSUE: The notion of Job Template needs more work. Suppose an Administrator wants to setup several sets of defaults for a Printer, say, for the different personalities that the actual output device contains, such as a PostScript and a PCL interpreter. Does the Printer have several URLs, where each URL implies a different Job Template? This works only if the xxx-supported attributes are the same for the different personalities. Each URL would appear in the directory, as if there were distinct Printers. However, in the case of PostScript and PCL, the fonts are different, so the fonts-supported need to be different. Also the resolutions may be different in some implementations. Therefore, the Administrator should really set up two different Printer proxy objects that don't queue and spool and that feed a single Printer object that represents the output device. Then each proxy Printer object can have different xxx-supported attributes, as well as different Job Templates. In the second approach, each proxy Printer has its own URL and appears in the directory as two different Printers. It is currently believed that when a client needs to present a Print Dialog box to an end user, it gets potential job values and default job values from a Printer. The default values are from the Job Template associated with the Printer named by the end user. If a end user sends a job to a Printer, the Printer may set unspecified attributes to the value of the associated Job Template.

415 3.4 Object Relationships

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

421 A Printer object is associated with zero or more Job Template
 422 objects. A Job Template object is associated with zero or more
 423 Printer objects, i.e., Printer objects can share the same Job
 424 Template object.

425 ISSUE - Can an Administrator define a Printer object that has no
 426 Job Template?

427 3.5 Object Identity

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

430 The following objects have the following mandatory identifier
 431 attributes:

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

432

433 4. Naming

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

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

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

441 Another example is the following URL:

442 http://1.2.3.4:nnn380/printer-2

443 In this case, the URL identifies the use of the HTTP protocol.
 444 The Printer is located at the node identified by the IP address of
 445 "1.2.3.4" using port nnn380 for the HTTP server, and "printer-2"
 446 is the name of the Printer. (The actual value of nnn is to be
 447 assigned by IANA as part of this standards project).

448 It is not necessary to expose the Job Template objects that might
 449 be associated with a given printer as separate objects. They can
 450 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~~ the ~~associated~~ Job Templates specified by a URL would contain have two a different Job Template default attribute sets. One Job Template would contain the defaults for two-sides printing and the other would contain the defaults for draft printing.

ISSUE - in this method, where is the identifier for the Printer stored? In the directory entry with the URL for the Job Template? The URL for the Printer can't be stored in the Job Template itself, otherwise the Job Template couldn't be used with more than one Printer.

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

- 3) <http://some.domain.com/hr-printer/resumes>
- 4) <http://some.domain.com/hr-printer/1040forms>

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

ISSUE: In this method, there are two directory entries that specify the same Printer. The Printer would know which Job Template to use by looking at which URL was supplied by the client in the Print operation. With this method, the Printer object could have a multi-valued attribute that contains the Job Templates supported for all of the allowed suffixes to the URLs in the directory for that Printer. Should we add such an attribute to the Printer object?

ISSUE: Will there be confusion and interoperability problems, if the URLs in directories specify Job Templates in some implementations and/or sites and specify Printers (with suffixes) in other implementations and/or sites?

ISSUE: The notion of Job Template needs more work. Suppose an Administrator wants to setup several sets of defaults for a Printer, say, for the different personalities that the actual output device contains, such as a PostScript and a PCL interpreter. Does the Printer have several URLs, where each URL implies a different Job Template? This works only if the xxx-supported attributes are the same for the different personalities. Each URL would appear in the directory, as if there were distinct Printers. However, in the case of PostScript and PCL, the fonts are different, so the fonts-supported need to be different. Also the resolutions may be different in some implementations. Therefore, the Administrator should really set up two different Printer proxy objects that don't queue and spool and that feed a single Printer

object that represents the output device. Then each proxy Printer object can have different xxx-supported attributes, as well as different Job Templates. In the second approach, each proxy Printer has its own URL and appears in the directory as two different Printers. Wouldn't it improve interoperability to always use the Printer proxy approach whenever the Administrator want to setup different sets of defaults for a Printer?

This specification establishes, through IANA, a new well known port, port ~~nnn380~~, 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 ~~IPPHTTP~~ printing would be acceptable on port ~~nnn380~~.

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.

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.

549 4.2.2 Description

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

552 4.2.3 Location

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

555 In order for filtered searches to be more effective, a given site
556 may use some regular structuring within the string values such as
557 "SITE:USA-San Jose,BUILDING:A1,FLOOR:2,ROOM:555" or "department5-
558 2ndFloor-A5-IndianHills-Chicago-IL-USA".

559 4.2.4 Maximum Print Quality

560 This directory attribute indicates a somewhat subjective
561 evaluation of the overall printing quality: ~~"high", "medium", or~~
562 ~~"low". The syntax and values shall be the same as for the print-~~
563 ~~quality job attribute.~~

564 4.2.5 Cost

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

568 4.2.6 Resolution

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

571 The syntax and semantics shall be the same as ~~forthat of~~ the
572 printer-resolution-select job attribute. ~~That syntax allows a~~
573 ~~single integer to specify the maximum resolution or a pair of~~
574 ~~integers to specify the maximum resolution when the x and y~~
575 ~~dimensions differ. When two integers are specified, the first is~~
576 ~~in the x direction, ie., the direction fo the shortest dimension~~
577 ~~of the medium, so that the value is independent of whether the~~
578 ~~Printer feeds long edge or short edge first.~~

579 4.2.7 Color Supported

580 This directory attribute specifies whether the Printer supports
581 color and, if so, what type. The values are a type2Enum (see
582 section 6). Standard values are: "none", ~~is a either a BOOLEAN~~
583 ~~for either yes, color printing is supported, or no color printing~~
584 ~~is not supported or it is an enumerated value such as "highlight",~~
585 ~~"three color (CMY)", "four color (CMYK)", "monochromatic", etc.~~

586 4.2.8 Fonts Supported

587 This directory attribute takes on a list of fonts that are
588 supported by the printer. The syntax and values shall be the same
589 as for the fonts-used job attribute.~~This is replicated from the~~
590 ~~fonts-supported attribute in the Printer object.~~

591 4.2.9 Maximum Speed

592 This directory attribute is the maximum speed of the printer ppm,
593 ipm, spm, lpm, or cps. They syntax and values shall be ~~are~~ the
594 same as for the maximum-printer-speed Printer attribute.

595 4.2.10 Device Id

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

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

603 4.2.11 Make and Model

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

608 vendor-name "/" model-name

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

611 For example: Is is recommended that the manufacturer use some
612 regular form such as "vendor-x/super-duper-printer".

613 4.2.12 Marker Type

614 This directory attribute is the printing mechanism of the print
615 device: electrophotographic-laser, ink-jet-aqueous, thermal-
616 transfer, etc. The syntax and values shall be ~~are~~ the same as for
617 the printer-types Printer attribute, except the ~~-~~value of the
618 Marker Type directory attribute shall be single-valued

619 4.2.13 Document Formats Supported

620 This directory attribute is a list of all of the document formats
621 that the printer and/or its interpreter(s) support. The syntax
622 and values shall be ~~are~~ the same as ~~those~~ for the document-formats-
623 supported JobPrinter attributes.

624 4.2.14 Sides Supported

625 This directory attribute specifies the capabilities of the Printer
626 for marking on sides of the medium. The syntax and values shall
627 be the same as the sides ~~supported JobPrinter~~ attribute. ~~Standard~~
628 ~~values are: 1 sided (simplex), 2 sided long edge (duplex), and 2~~
629 ~~sided short edge (tumble).~~

630 4.2.15 Finishings Supported

631 This directory attribute identifies the finishing operations
632 supported by the Printer. The syntax and values shall be the same
633 ~~as standard finishing objects are defined in the section on the~~
634 finishing job attribute.

635 4.3 Directory Entries Using LDAP

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

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

645 To create a Printer object directory entry using LDAP:

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

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

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

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

656 where ld is the connection handle for subsequent LDAP APIs.

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

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

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

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

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

669 Example: ldap_unbind (ld)

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

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

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

677 ldap:///dir.host.name??(sides-supported=2-sided-long-binding-
678 edge) |

679 Issue: Is it allowed to filter the search based on the object
680 class itself, in this case the object class of Printer? We need
681 to define this new object class. How do we do this?

682 5. IPP Operations

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

686 5.1 HTTP Overview 687

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

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

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

707 Current practice requires that the connection be established by
708 the client prior to each request and closed by the server after
709 sending the response. Both clients and servers ~~shall~~^{must} be |
710 capable of handling cases where either party closes the connection
711 prematurely, due to user action, automated time out, or program
712 failure.

713 5.2 IPP Operation Encoding

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

716 | _____ HTTP MESSAGE = Request | Response

717
 718 Requests and responses use the generic message format of RFC 822
 719 for transferring entities. Both messages may include optional
 720 header fields and an entity body. The entity body is separated
 721 from the headers by a null line (a line with nothing preceding the
 722 CRLF).

723
 724 Request = Request-line
 725 * (General-Header
 726 | Request-Header
 727 | Entity-Header)
 728 CRLF
 729 [Entity-Body]

730
 731 Response = Status-line
 732 * (General-Header
 733 | Request-Header
 734 | Entity-Header)
 735 CRLF
 736 [Entity-Body]

737
 738 All IPP headers conform to the syntax

739 | _____ IPP Header = field name ":" [field-value] CRLF.

740
 741 IPP/1.0 defines the octet sequence CRLF as the end-of-line marker
 742 for all protocol elements except the entity-body.

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

748 5.2.1 HTTP Request-Header Fields

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

755
 756 HTTP Entity-Header = Content-Encoding
 757 | Content-Length
 758 | Content-Type
 759 | extension-header

760
 761 The Content-Length field must always be a valid length, This means
 762 that for any Print Operations based on HTTP 1.0, the entire
 763 content must be generated before this header can be built. HTTP

764 1.1 provides the notion of "chunks" which will allow the content
765 to be generated dynamically as the data is sent.

766
767 Content-Type will always be "Application/IPP".

768 5.2.1.1 IPP Request-Line

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

772
773 IPP Request-Line = Operation token IPP/1.0 CRLF
774
775 Operation token = Print | CancelJob | GetAttributes |
776 GetJobs
777

778 5.2.2 HTTP Response-Header Fields

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

783 5.2.2.1 IPP Status-Line

784 The first line of the entity body in an IPP response is the IPP
785 Status-Line. The status-line consists of a protocol version
786 followed by a numeric status-code and an associated text message.
787

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

789 5.3 The Print Job

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

796
797 Print Job = Print-Job-Object-Header ;section (5.3.1)
798 [Job Attributes] ;section (5.3.4)
799 *(Documents)
800
801
802 Document = Document-Header ;section (5.3.2)
803 [Document attributes] ;section (5.3.5)
804 [Content-Header ;section (5.3.3)
805 content]
806

807 5.3.1 Print Job Object Header

808 Print-Job-Object Header = Content-Encoding
809 | Content-Length
810 | Content-Type

811 | extension-header

812

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

815 5.3.2 Document Header

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

821 Document Header = Content-Encoding

822 | Content-Length

823 | Content-Type

824 | extension-header

825

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

828 5.3.3 Document-Content Header

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

834

835 Document-Content-Header = Content-Encoding

836 | Content-Length

837 | Content-Type

838 | extension-header

839

840 Content-Type is defined as :

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

842

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

845 Content-Type: Postscript/2.0

846

847 Other header fields are as defined by HTTP 1.0.

848 5.3.4 Job Attributes

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

851

852 Job-Attribute = attribute name ":" Attribute value CRLF

853

854 Attribute value = Value | *(Value "," Value)

855

856

857 5.3.5 Document Attributes

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

860 Document-Attribute = attribute name ":" Attribute value CRLF
861

862 Attribute value = Value | *(Value "," Value)
863

864 865 5.4 Operation Semantics

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

868 5.4.1 Print Operation

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

871 Note that the Printer name is not needed since it is the target of
872 the entire operation. A Print Job contains the information needed
873 by the Printer object to print a document or set of documents.
874 When the print operation is invoked, the Entity-Body in the HTTP
875 request includes an IPP Print Job. The concrete syntax of the
876 Print Job is defined in section 6.2.

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

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

886 887 5.4.1.1 Print Request

888 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
-----------------------------	---

<u>Requested Attributes</u>	<u>A set of attributes without values in whose values the requester is interested.</u>
Document Contents	Document content is optional and <u>shall not be</u> included when a URL is provided to point to the content.

889 890 891 5.4.1.2 Print Response

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

893 Job-Identifier A URL Used for all other operations on this Job.
 Job Status Current-job-state
 Printer State Printer-state
 Result The requested attributes with their currrent
 Attributes values, if the requester supplied any Requested
 Attributes
 Message Optional message
 Errors Optional Error Information

894

895

896 5.4.2 Cancel Job Operation

897 This operation allows a user to cancel one specific Print Job any
 898 time after the print job has been established on the Printer
 899 Object. Some pages may be printed before a job is terminated if
 900 printing has already started when the Cancel Job operation is
 901 received.

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

904 5.4.2.1 CancelJob Request

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

907

 Message Optional message to the operator.

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

908

909 5.4.2.2 CancelJob Response

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

912

 Job Status Optional Job status information

 Errors Optional Error Information

913

914 5.4.3 Get Attributes Operation

915 This operation allows an end-user to obtain information from the
 916 Print object concerning jobs, printers, and print queues, based on
 917 ISO 10175. The entity-body of the Get Attributes operation
 918 contains the set of attributes that the requester is interested
 919 in. The requester should not supply values in the Requested

920 Attributes input parameter; the Printer shall ignore the values of
 921 any supplied by the requester. However, the attribute values may be
 922 null and are ignored by the server. The attribute list ~~is~~
 923 returned in the response with the appropriate attribute values
 924 filled in. If no attribute list is supplied, then all attributes
 925 defined for that object are returned.

926 5.4.3.1 GetAttributes Request

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

Selector	Job-Identifier (URL) or Printer URL <u>or</u> <u>Job Template URL</u>
Requested Attributes	A set of attributes <u>without values in whose</u> <u>values in which</u> the requestor is interested

929 5.4.3.2 GetAttributes Response

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

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

933

934 5.4.4 Get Jobs Operation

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

941 5.4.4.1 GetJobs

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

selector	<u>Indicates which jobs the requester seeks. The</u> <u>values are type2Enum (see section 6). Standard</u> <u>values are: "</u> <u>all--jobs" - (including completed jobs),</u> <u>"pending" - all jobs which are pending and</u> <u>processing, or just</u> <u>"my--jobs" - my jobs that are pending or</u> <u>processing.</u>
<u>Requested</u> <u>Attributes</u>	<u>A set of attributes without values in whose</u> <u>values the requester is interested.</u>

945

946

947 5.4.4.2 Get Jobs Response

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

950

Jobs	A list of Job URLs is returned. The list is in "scheduled" order. <u>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.</u>
<u>Result Attributes</u>	<u>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</u> For each Job URL the following attributes are returned: <u>job-total-octets and number-of-intervening-jobposition-in-list.</u> 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 <u>or cannot be because of site security policy restrictions.</u>
Errors	Optional Error Information

951
 952
 953 6. Object Attributes

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

959 6.1 Attribute Syntaxes

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

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

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

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

974 An attribute value with a **":on-order"** State means that the
 975 provider has placed an order for the indicated value and that the

976 operator must wait until the resource is delivered before the job
 977 can be printed. However, an end-user may submit a job that
 978 requires such a resource and the Printer shall accept such a job.

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

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

986 **media-supported=na-letter-white, na-letter-transparent, b:not-**
 987 **ready**

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

991 The sections below reference the following syntax items:

string	arbitrary ASCII strings, no control characters, except <SPACE>. TBD
stringPair	string ":" string
stringState	string S state
name	arbitrary ASCII strings, no control characters, and no <SPACE> characters. TBD
urlURL	Universal Resource Locator TBD
dateTime	date and time in RFC 822 format TBD
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. TBD
type2Enum	standard names, but an implementor can add new TBD by proposing them to the PWG for registration (or an IANA-appointed registry advisor after the PWG is no longer certified) anytime. IANA keeps the registry. Implementors can supportadd private (un-registered) with a suitable distinguishing prefix, such as -xxx- where xxx is the company name registered with IANA <u>for use in domain names</u> .
type3Enum	standard names, but an implementor can add new names by submitting a registration request directly to IANA, no PWG or IANA-appointed registry advisor review is required. Implementors can supportadd private (un-registered) names with a suitable distinguishing prefix, such as -xxx- where xxx is the company name

	registered with IANA <u>for use in domain names.</u> TBD
type2EnumState	type2Enum <u>S</u> state
type3EnumState	type3Enum <u>S</u> state
state	TBD
<u>b</u> Boolean	tokens: yes, y, true, or t and no, n, false, or f. TBD
positiveInteger	1 .. n represented as ASCII digits TBD
positiveIntegerCross	positiveInteger ["x" positiveInteger]
positiveIntegerCrossState	positiveIntegerCross <u>S</u> state
positiveIntegerRange	positiveInteger ":" positiveInteger
positiveIntegerUnits	positiveInteger units
positiveIntegerState	positiveInteger <u>S</u> state
units	"ppm" "ipm" "spm" "cps" "lpm"
type3Locale	type3Country ":" type3Language ":" type3CodeSet
type3Country	type3Enum - <u>Standard values are the two-character country codes from ISO 639.</u>
type3Language	type3Enum - <u>Standard values are the two-character language codes from ISO 3166.</u>
type3CodeSet	type3Enum - <u>Standard values are from the IANA Code Set registry.</u>
type2Format	name ["/" version]
version	name
type3LocaleState	type3Locale <u>S</u> state

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

"#" in front of a data syntax	means zero or more
<u>"1#" in front of a data syntax</u>	<u>means one or more values separated by ",."</u>

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.

1011 A client may use a job template associated with the selected
1012 printer in order to initialize the job. To do so, the client uses
1013 the Get-Attributes operation to get the URLs of the Printer's Job
1014 Templates. Then the client may get the default attributes from
1015 the Printer's default Job Template in order to initialize a
1016 display to the end-user with the Printer's defaults. See the
1017 printer-job-templates Printer attribute. However, a client need
1018 not access the Job Template in order to issue a Print operation;
1019 the client can depend on the Printer to supply the default job
1020 object attribute values as part of the Print operation.

1021 Each section heading below contains the name of an attribute and
1022 its syntax in parentheses using the rules of RFC 822.

1023 6.2.1 Job Informational Attributes (Set by a Client/End User)

1024 The client may specify these attributes in the Print operation to
1025 provide information to identify a print-job.

1026 The client may also specify these attributes in the operations:
1027 Get-Attributes, and Get-Jobs.

1028 6.2.1.1 job-name (string)

1029 This attribute supplies a human readable string for naming the
1030 print-job.

1031 This attribute is intended ~~for~~ to be printed on a start sheet,
1032 returned in a Get-Jobs result, or used in notification messages.

1033 If the client does not specify this attribute, a Printer shall set
1034 it to the value of the document-name attribute ~~name of the file~~ of
1035 the first document in the job.

1036 6.2.2 Job Informational Attributes (Set by a Printer)

1037 The Print shall add all of these attributes to a job to provide
1038 information to identify a print-job.

1039 The client may specify these attributes in the operations: Get-
1040 Attributes and Get-Jobs, but not in Print.

1041 6.2.2.1 job-identifier (url)

1042 This attribute provides the job-identifier for this job on the
1043 Printer. The Printer ~~shall~~ generate a job-identifier value as a
1044 URL.~~␣~~

1045 The value of the job-identifier attribute shall be returned by the
1046 Printer ~~as~~ part of the PrintResult in the Print operation.

1047 6.2.2.2 job-originator (name)

1048 This attribute specifies the name of the person submitting the
1049 print job. The Printer shall set this attribute to the most
1050 authentic name that it can obtain from the client. The operation-
1051 user-name attribute is intended to be a source of the most
1052 authentic name.

1053 6.2.2.3 job-originating-host (name)

1054 This attribute identifies the originating host of the job. The
1055 Printer shall set this attribute to the value of the operation-
1056 host-name which is intended to be the most authentic host name of
1057 the client.

1058 ~~6.2.2.4 notification-address (name)~~

1059 ~~This address specifies the email address of the client. The client~~
1060 ~~specifies this attribute in the operation notification address~~
1061 ~~attribute which the Printer in turn uses to set this attribute.~~

1062 ~~The Printer shall use this attribute as the address for sending~~
1063 ~~messages to a job submitter when an event occurs that the end user~~
1064 ~~has registered an interest in or when certain other events occur,~~
1065 ~~such as Cancel Job.~~

1066 ~~Note: The only type of notification is email.~~

1067 ~~Issue: Can the email address be inferred with job-originator and~~
1068 ~~the originating host?~~

1069 6.2.2.4 job-locale (type3Locale)

1070 This attribute identifies the locale of the job, i.e., the country,
1071 language, and coded character set. The Printer sets this attribute
1072 from the value of the operation-locale.

1073 The Printer shall use this attribute to determine the locale for
1074 notification messages that it sends.

1075 ~~The type3EnumTrip consists of 3 colon separated type 3 enums. The~~
1076 ~~first shall be the two character country code from ISO 639. The~~
1077 ~~second shall be the two character language code from ISO 3166. The~~
1078 ~~third is the code set from the IANA Code Set Registry.~~

1079 Issue: Is there a more standard syntax for locale?

1080 6.2.3 Job Status Attributes (Set by Printer)

1081 The Printer shall add these attributes to a job when a client
1082 submits a job, and the Printer shall assign appropriate values to
1083 each such job-status attribute.

1084 The Printer uses these attributes to specify the job status
1085 before, during and after the processing of the print-job by the
1086 Printer.

1087 The client may specify job-status attributes in: Get-Attributes
1088 and Get-Jobs, but not Print.

1089 6.2.3.1 current-job-state (type1Enum)

1090 This attribute identifies the current state of the job ~~with the~~
1091 ~~following values: unknown, pre-processing, pending, processing,~~
1092 ~~printing, held, terminating, retained, completed.~~ Standard values
1093 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's job-submission-complete attribute is TRUE since the server has received a print request with the job-submission-complete parameter TRUE and 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 on at least one printer . That is, an output device <u>print engine</u> 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 printers' <u>s</u> printer-state attribute(s).
paused	The job has been paused as a result of a PauseJob operation.
interrupted	The job was interrupted by the InterruptJob request for an intervening job, and shall resume processing automatically once the intervening job has completed.
terminating	The job has been cancelled by a CancelJob 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 cancelled by the CancelJob 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 <u>IPP V1.0 ISO/IEC 10175-Part 1</u> .

completed The job has:

- (1) completed successfully or with warnings or errors,
- (2) been aborted by the server while printing, or
- (3) been cancelled by the CancelJob 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-~~originatorowner~~, job-name, current-job-state, ~~output-deviceprinters-~~assigned, and job-state-reasons.

1094

1095 The IPP protocol supports all values for job states, but Printers
 1096 ~~are-need-~~ only support those states which are appropriate for the
 1097 particular implementation.

1098 6.2.3.2 output-device-assigned (name)

1099 This attribute identifies the Output Device to which the Printer
 1100 has assigned this job.

1101 If an Output Device implements a Printer, the Printer needs~~does~~ not
 1102 set this attribute.

1103 If a Print Server implements a Printer, the value shall be empty
 1104 until the Printer assigns an Output Device to the job.~~-~~

1105 The value of the job's ~~output-deviceprinter-~~assigned attribute
 1106 shall remain after the job has completed, so that end users can
 1107 determine the Output Device on which the job was printed.

1108 6.2.3.3 submission-time (dateTime)

1109 This attribute indicates the time at which this job was accepted
 1110 by the Printer. If the Printer does not support the notion of
 1111 time, the attribute needs~~is~~ not be stored as part of the job
 1112 object.

1113 6.2.3.4 number-of-intervening-jobs (cardinalpositiveInteger)

1114 This attribute indicates the number of jobs that are "ahead" of
 1115 this job in the current scheduled order. For efficiency, it is

1116 only necessary to calculate this value when an operation is
 1117 performed that requests this attribute.

1118 NOTE - This attribute is necessary since an end user may request
 1119 just their own jobs and they need some relative position indicator
 1120 if there are other jobs interspersed in the waiting list which are
 1121 not returned in the response or cannot be because of site security
 1122 policy restrictions.

1123 6.2.3.5 job-message-from-administrator (string)

1124 This attribute provides a message from an operator, system
 1125 administrator or "intelligent" process to indicate to the end user
 1126 the reasons for modification or other management action taken on a
 1127 job.

1128 6.2.3.6 completion-time (dateTime)

1129 This attribute indicates the time at which this job completed.
 1130 This time is useful for jobs which are retained after printing.
 1131 If the Printer does not support the notion of time, the attribute
 1132 is not stored as part of the Job object.

1133 6.2.3.7 job-state-reasons (1#type2Enum)

1134 This attribute identifies the reason or reasons that the job is in
 1135 the state that it is in (e.g., held, terminating, retained,
 1136 completed, etc.). The printer shall indicate the particular
 1137 reason(s) by setting the value of the job-state-reasons attribute.
 1138 ~~It is valid for the printer to set the value of the job state~~
 1139 ~~reasons attribute to the empty set.~~

1140 The following standard values are defined:

<u>none</u>	<u>There are not reasons associated with the job's current state.</u>
documents-needed	The complete job has been accepted by the server (the value of the job-submission-complete element was TRUE in the last print request for the job) , 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 <u>or print-off-peak</u> attributes <u>has</u> 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.

1141

1142

1143 6.2.3.8 impressions-completed (cardinal)

1144 This attribute contains the number of impressions that the Printer

1145 has completed printing. If the Printer cannot report this number,

1146 the Printer leaves this attribute unspecified.

1147 6.2.3.9 media-sheets-completed (cardinal)

1148 This attribute contains the number of media-sheets that the

1149 Printer has completed printing. If the Printer cannot report this

1150 number, the Printer leaves this attribute unspecified.

1151 6.2.4 Job Sheet Attributes (Set by Client/End User)

1152 The client shall specify these attributes to control the printing

1153 of job sheets.

1154 The client may also specify job sheet attributes in: Get-

1155 Attributes and Get-Jobs.

1156 6.2.4.1 job-sheets (type3Enum)

1157 This attribute determines what type of job-sheets the Printer

1158 shall print with the job.

1159 The standard -values are: none, and default-sheet.

1160 The value "none" means that the Printer shall print no job sheets.

1161 The value "default-sheet" means that the Printer shall print the

1162 job sheets defined by an administrator. If the administrator's

1163 policy is not to support none, the Printer shall use the default-

1164 sheet value if the client supplies the "none" value.

1165 NOTE - The effect of this attribute on jobs and documents is

1166 controlled by the files-are-one-document and files-are-interleaved

1167 job attributes.

1168 6.2.5 Notification -Attributes (Set by a Client/End User)

1169 The client shall specify these attributes to indicate events that

1170 the client is interested in, along with the notification address

1171 and method for performing the notification.

1172 The client may also specify notification attributes in: Get-

1173 Attributes and Get-Jobs.

1174 6.2.5.1 notification-events (1#type21Enum)

1175 This attribute specifies the ~~-events~~ about which the end user want
1176 to be notified.

1177 ~~Standard values are~~This attribute will support four events
1178 ~~classes:~~ none, job-completion, job-problems and printer-problems.

1179 If this attribute contains the event none, the Printer shall not
1180 notify. This value is useful if an administrator has set up a
1181 notification Printer default but the end user does not ~~want~~which
1182 notification. If the none value and other values are supplied,
1183 the Printer shall ignore the none value.

1184 ~~This attribute will support only one delivery method, namely~~
1185 ~~email. The attribute notification-address specifies the email~~
1186 ~~address.~~

1187 If this attribute contains the value: event job-completion, the
1188 Printer shall notify the client when the job containing this
1189 attribute completes with or without errors or is cancelled by the
1190 end-user or the operator.

1191 If this attribute contains the value: event job-problems, the
1192 Printer shall notify the client when this~~the~~ job ~~containing this~~
1193 ~~attribute~~ has a problem while this~~the~~ job is printing. Problems
1194 include: paper jam and out-of-paper.

1195 If this attribute contains the value: event printer-problems, the
1196 Printer shall notify the client when any~~the~~ job, including this
1197 job, containing this attribute has a problem while this~~the~~ job is
1198 ~~printing or~~ waiting to print or printing. Problems include: paper
1199 jam and out-of-paper.

1200 6.2.5.2 notification-address (url~~name~~)

1201 This address specifies both the address and mechanism for delivery
1202 of notification events to~~the email address of~~ the client. The
1203 client specifies this attribute in the operation-notification-
1204 address attribute which the Printer in turn uses to set this
1205 attribute.

1206 The Printer shall use this attribute as the address for sending
1207 messages to a job submitter when an event occurs that the end user
1208 has registered an interest in or when certain other events occur,
1209 such as Cancel-Job.

1210 If the URL has a "mailto:" scheme, then email is used and the rest
1211 of the URL is used as the email address. If the URL has a "http:"
1212 scheme, then an HTTP APPEND method is used to add HTML formatted
1213 events to the end of the specified HTML file.

1214 ~~Note: The only type of notification is email.~~

1215 ~~Issue: Can the email address be inferred with job-originator and~~
1216 ~~the originating host?~~

1217 6.2.6 Job Scheduling Attributes (Set by Client/End User)

1218 The client shall specify these attributes to provide the Printer
1219 with information ~~for~~ the scheduling a print-job.

1220 The client may also specify these ~~attributes~~ in: Get-Attributes
1221 and Get-Jobs.

1222 6.2.6.1 job-priority (type1Enum)

1223 This attribute specifies a priority for scheduling the print-job.
1224 Printers ~~that~~ employ a priority-based scheduling algorithm use
1225 this attribute.

1226 There are three standard values: high, default, and low. Among
1227 those jobs that are ready to print, a Printer shall print all such
1228 jobs with a high priority before printing those with a default or
1229 low priority, and a Printer shall print all such jobs with a
1230 default priority before printing those with a low priority.

1231 If the client does not specify this attribute, the Printer assumes
1232 that the end user places no constraints concerning priority on the
1233 scheduling of the print-job, and it has a priority value of
1234 default.

1235 An operator can modify a job to have any priority. An end-user is
1236 restricted by the value of the maximum-end-user-priority Printer
1237 attribute.

1238 6.2.6.2 job-print-after (dateTime)

1239 This attribute specifies the calendar date and time of day after
1240 which the print-job shall become a candidate for printing.

1241 If the value of this attribute is in the future, the Printer shall
1242 set the value of the job's current-job-state to held and add the
1243 job-print-after-specified value to the job's job-state-reasons
1244 attribute and shall not schedule the print-job for printing until
1245 the specified date and time has passed. When the specified date
1246 and time arrives, the Printer shall remove the job-print-after-
1247 specified value from the job's job-state-reason attribute and, if
1248 no other reasons remain, shall change the job's current-job-state
1249 to pending so that the job becomes a candidate for being scheduled
1250 to print.

1251 If this attribute is unspecified or the value is in the past, the
1252 job shall be a candidate for scheduling immediately.

1253 6.2.6.3 job-print-off-peak (type3Enum)

1254 This attribute specifies the off-peak period during which the
1255 print-job shall become a candidate for printing.

1256 Standard values are: "evening", "night", "weekend", "second-
1257 shift", "third-shift".

1258 If this attribute is specified, it contains a value with which an
1259 administrator has associated allowable print times. An

1260 administrator is encouraged to pick names that suggest the type of
1261 off-peak period.

1262 If the value of this attribute is in the future, the Printer shall
1263 set the value of the job's current-job-state to held and add the
1264 job-print-after-specified value to the job's job-state-reasons
1265 attribute and shall not schedule the print-job for printing until
1266 the specified date and time has passed. When the specified date
1267 and time arrives, the Printer shall remove the job-print-after-
1268 specified value from the job's job-state-reason attribute and, if
1269 no other reasons remain, shall change the job's current-job-state
1270 to pending so that the job becomes a candidate for being scheduled
1271 to print.

1272 If this attribute is unspecified, the job shall be a candidate for
1273 scheduling immediately.

1274 6.2.6.4 job-retention-period (deltaTime)

1275 The retention time is expressed in hours and minutes, e.g. 6:00 (6
1276 hours), or 20 (20 minutes).

1277 This attribute specifies the minimum period of time following the
1278 completion of job processing and printing that the server shall
1279 keep job attributes and document data. The Printer may keep these
1280 attributes and data longer than the value of the job-retention-
1281 period attribute.

1282 NOTE - the requester may change this job attribute using the input
1283 parameter to the CancelJob operation.

1284 ~~Issue: There is some discussion about whether or not this should~~
1285 ~~be removed from the spec?~~

1286 6.2.7 Job Production Attributes (Set by Client/End User)

1287 The client shall specify these attributes to affect the rendering,
1288 production and finishing of the documents in the job. Similar
1289 types of instructions ~~may~~ also be contained in the document to be
1290 printed.

1291 If there is a conflict between the value of one of these
1292 attributes, and a corresponding instruction in the document
1293 (either implicit or explicit), the value of the attribute shall
1294 take precedence over the document instruction.

1295 Job Production and Resource Attributes each address a similar set
1296 of features but they have different uses.

1297 A job production attribute provides a client with a way to request
1298 some feature at print time that may ~~is~~ not have been embedded
1299 within the document data when the document was created. A job
1300 production attribute also provides a client with a way to override
1301 a feature at print time that was embedded within the document data
1302 when the document was created.~~After some program has merged the~~
1303 ~~production attributes into the document data After the information~~
1304 ~~from these attributes has been folded into the document data~~
1305 ~~(possibly during a translation process of the document data),~~

1306 ~~these attributes are no longer relevant and shall can be~~
 1307 ~~discarded from a job. Instead, the resource attributes specify~~
 1308 ~~the resources needed to print the job as modified by the job~~
 1309 ~~production attributes.~~

1310 Note: until companies that supply interpreters for PDL's, such as
 1311 PostScript and PCL allow a way to specify overrides for internal
 1312 job production instructions, a Printer may not be able to
 1313 implement these attributes for some PDL's.

1314 A job resource attribute tells a Printer what features the job
 1315 needs. A program that translates document data to a Printer's PDL,
 1316 and/or merges production attributes into the document data should
 1317 add job resource attributes to a job.

1318 For example, a job production attribute medium-select with the
 1319 value of "letter" requests that a job be printed on letter paper,
 1320 but gives no information about what resources the job needs. ~~For~~
 1321 ~~example, Aa job~~ resource ~~production~~-attribute media-used with the
 1322 values of "letter" and "ledger" tell a Printer that the job needs
 1323 letter and ledger paper, but gives no information about which
 1324 pages use each medium.

1325
 1326 The client may also specify ~~jobdocument~~ production-instruction
 1327 attributes in: Get-Attributes and GetJobs.

1328 6.2.7.1 medium-select (type2Enum)

1329 This attribute identifies the medium that the Printer shall use
 1330 for all ~~-~~pages of the document regardless of what media are
 1331 specified within the document.

1332 The values for medium include medium-names, medium-sizes, input-
 1333 trays and electronic forms so that one attribute specifies the
 1334 media.

1335 Standard values are ~~defined~~ (taken from ISO DPA and the Printer
 1336 MIB):

1337

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

jis-b4-coloured	Specifies the JIS B4 coloured medium
jis-b5-white	Specifies the JIS B5 white medium
jis-b5-coloured	Specifies the JIS B5 coloured medium

1338

```
1339     The following standard values are defined for North American
1340     media:
```

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

1341

```
1342      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

1343

```

1344     The following standard values are defined for the less commonly
1345     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

1346

1347 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

1348

1349 The following standard values are defined for input-trays (from
 1350 ISO DPA and the Printer MIB):

<u>top</u>	<u>The top input tray in the printer.</u>
<u>middle</u>	<u>The middle input tray in the printer.</u>
<u>bottom</u>	<u>The bottom input tray in the printer.</u>
<u>envelope</u>	<u>The envelope input tray in the printer.</u>
<u>manual</u>	<u>The manual feed input tray in the printer.</u>
<u>large-capacity</u>	<u>The large capacity input tray in the printer.</u>
<u>main</u>	<u>The main input tray</u>
<u>side</u>	<u>The side input tray</u>

1351

1352 The following standard values are defined for media sizes (from
 1353 ISO dPA):

	<u>iso-a0</u>	<u>Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216</u>
	<u>iso-a1</u>	<u>Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216</u>
	<u>iso-a2</u>	<u>Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216</u>
	<u>iso-a3</u>	<u>Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216</u>
	<u>iso-a4</u>	<u>Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216</u>
	<u>iso-a5</u>	<u>Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216</u>
	<u>iso-a6</u>	<u>Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216</u>
	<u>iso-a7</u>	<u>Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216</u>
	<u>iso-a8</u>	<u>Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216</u>
	<u>iso-a9</u>	<u>Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216</u>
	<u>iso-a10</u>	<u>Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216</u>
1354	<u>iso-b0</u>	<u>Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216</u>
	<u>iso-b1</u>	<u>Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216</u>
	<u>iso-b2</u>	<u>Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216</u>
	<u>iso-b3</u>	<u>Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216</u>
	<u>iso-b4</u>	<u>Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216</u>
	<u>iso-b5</u>	<u>Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216</u>
	<u>iso-b6</u>	<u>Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216</u>
	<u>iso-b7</u>	<u>Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216</u>
	<u>iso-b8</u>	<u>Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216</u>
	<u>iso-b9</u>	<u>Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216</u>
	<u>iso-b10</u>	<u>Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216</u>
1355	<u>na-letter</u>	<u>Specifies the North American letter size: 8.5 inches by 11 inches</u>
	<u>na-legal</u>	<u>Specifies the North American legal size: 8.5 inches by 14 inches</u>
	<u>executive</u>	<u>Specifies the executive size (7.25 X 10.5 in)</u>
	<u>folio</u>	<u>Specifies the folio size (8.5 X 13 in)</u>
	<u>invoice</u>	<u>Specifies the invoice size (5.5 X 8.5 in)</u>
	<u>ledger</u>	<u>Specifies the ledger size (11 X 17 in)</u>
	<u>quarto</u>	<u>Specifies the quarto size (8.5 X 10.83 in)</u>
1356	<u>iso-c3</u>	<u>Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269</u>

	<u>iso-c4</u>	<u>Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269</u>
	<u>iso-c5</u>	<u>Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269</u>
	<u>iso-c6</u>	<u>Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269</u>
	<u>iso-designated-long</u>	<u>Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO 269</u>
1357	<u>na-10x13-envelope</u>	<u>Specifies the North American 10x13 size: 10 inches by 13 inches</u>
	<u>na-9x12-envelope</u>	<u>Specifies the North American 9x12 size: 9 inches by 12 inches</u>
	<u>na-number-10-envelope</u>	<u>Specifies the North American number 10 business envelope size: 4.125 inches by 9.5 inches</u>
	<u>na-7x9-envelope</u>	<u>Specifies the North American 7x9 inch envelope size</u>
	<u>na-9x11-envelope</u>	<u>Specifies the North American 9x11 inch envelope size</u>
	<u>na-10x14-envelope</u>	<u>Specifies the North American 10x14 inch envelope size</u>
	<u>na-number-9-envelope</u>	<u>Specifies the North American number 9 business envelope size</u>
	<u>na-6x9-envelope</u>	<u>Specifies the North American 6x9 envelope size</u>
	<u>na-10x15-envelope</u>	<u>Specifies the North American 10x15 envelope size</u>
	<u>monarch-envelope</u>	<u>Specifies the Monarch envelope size (3.87 x 7.5 in)</u>
1358	<u>a</u>	<u>Specifies the engineering A size: 8.5 inches by 11 inches</u>
	<u>b</u>	<u>Specifies the engineering B size: 11 inches by 17 inches</u>
	<u>c</u>	<u>Specifies the engineering C size: 17 inches by 22 inches</u>
	<u>d</u>	<u>Specifies the engineering D size: 22 inches by 34 inches</u>
	<u>e</u>	<u>Specifies the engineering E size: 34 inches by 44 inches</u>
1359	<u>jis-b0</u>	<u>Specifies the JIS B0 size: 1030mm x 1456mm</u>
	<u>jis-b1</u>	<u>Specifies the JIS B1 size: 728mm x 1030mm</u>
	<u>jis-b2</u>	<u>Specifies the JIS B2 size: 515mm x 728mm</u>
	<u>jis-b3</u>	<u>Specifies the JIS B3 size: 364mm x 515mm</u>
	<u>jis-b4</u>	<u>Specifies the JIS B4 size: 257mm x 364mm</u>
	<u>jis-b5</u>	<u>Specifies the JIS B5 size: 182mm x 257mm</u>
	<u>jis-b6</u>	<u>Specifies the JIS B6 size: 128mm x 182mm</u>
	<u>jis-b7</u>	<u>Specifies the JIS B7 size: 91mm x 128mm</u>
	<u>jis-b8</u>	<u>Specifies the JIS B8 size: 64mm x 91mm</u>
	<u>jis-b9</u>	<u>Specifies the JIS B9 size: 45mm x 64mm</u>
	<u>jis-b10</u>	<u>Specifies the JIS B10 size: 32mm x 45mm</u>
1360		

1361 6.2.7.2 finishing (type2Enum)

1362 This attribute identifies the finishing operation that the Printer
 1363 should apply to each copy of the printed document.

1364 ~~NOTE - The effect of this attribute on jobs and documents is~~
 1365 ~~controlled by the files-are-one-document and files-are-interleaved~~
 1366 ~~job attributes. Examples include stapling, saddle stitching, hole-~~
 1367 ~~drilling, binding with tape, etc.~~

1368 Standard values for this attribute are:

<u>none</u>	<u>Perform no finishing.</u>
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.

1369

1370 6.2.7.3 number-up (~~type3Enum~~positiveInteger)

1371 This attribute specifies the number of source page-images to
 1372 impose upon a single side of an instance of a selected medium.

1373 In general, only certain numeric values are valid for this
 1374 attribute and the value "none", depending upon the Printer
 1375 implementation to which the print-request is directed. Standard
 1376 values are: "none", "1", "2", "4". Typical supported values are 2
 1377 and 4. If this attribute is unspecified or has a value of 1, then
 1378 the Printer does not apply any number up transformation to the
 1379 pages.

1380 This attribute primarily controls the translation, scaling and
 1381 rotation of page images, but a site may choose to add
 1382 embellishments, such as borders to each logical page. The value
 1383 "none" shall not include any embellishments and shall place one
 1384 logical page on a single side of an instance of the selected
 1385 medium without any translation, scaling, or rotation. If
 1386 embellishments are added, especially for the number up = 1 case,
 1387 these are controlled through some other mechanism or attribute.
 1388 The user expects that if number up is absent or equal to 1 then no
 1389 other imposition embellishments are added via this attribute.

1390 6.2.7.4 sides (type2Enum)

1391 This attribute specifies how source page-images are to be imposed
 1392 upon the sides of an instance of a selected medium. whether the
 1393 document should be printed in one of three ways: 1-sided
 1394 (simplex), 2-sided-long-binding edge (duplex), 2-sided-short-
 1395 binding edge (tumble).

1396 The standard values are: 1-sided, 2-sided-long-edge, 2-sided-
 1397 short-edge.

1398 1-sided imposes each consecutive source page-image upon the same
 1399 side of consecutive media sheets.

1400 2-sided-long-edge imposes each consecutive pair of source page-
 1401 image upon front and back sides of consecutive media sheets, such
 1402 that the orientation of each pair of source-pages on the medium
 1403 would be correct for the reader as if for binding on the long
 1404 edge. This imposition is sometimes called "duplex".

1405 2-sided-short-edge imposes each consecutive pair of source page-
 1406 image upon front and back sides of consecutive media sheets, such
 1407 that the orientation of each pair of source-pages on the medium
 1408 would be correct for the reader as if for binding on the short
 1409 edge. This imposition is sometimes called "tumble" or "head-to-
 1410 toe".

1411 ISSUE - How does sides interact with portrait vs. landscape and
 1412 reverse-landscape documents?

1413 6.2.7.5 copies (positiveInteger)

1414 This attribute specifies the number of copies of the job to be
 1415 printed. If this attribute is unspecified by both the client and
 1416 the Printer's Job Template, its default value shall be ~~is~~ 1 ~~copy~~.

1417 NOTE - The effect of this attribute on jobs and documents is
 1418 controlled by the files-are-one-document and files-are-interleaved
 1419 job attributes.

1420 6.2.7.6 printer-resolution-select (positiveIntegerCross)

1421 This attribute specifies the resolution that the Printer should
 1422 use.

1423 The syntax allows a single integer to specify the resolution or a
 1424 pair of integers to specify the resolution when the x and y
 1425 dimensions differ. When two integers are specified, the first is
 1426 in the x direction, ie., in the direction ~~off~~ the shortest
 1427 dimension of the medium, so that the value is independent of
 1428 whether the printer feeds long edge or short edge first.

1429 6.2.7.7 print-quality (type2Enum)

1430 This attribute specifies the print quality that the Printer should
 1431 use.

1432 The standard values are:

1433	draft	Lowest quality available on the printer
1434	normal	Normal or intermediate quality on the printer
1435	high	Highest quality available on the printer

1436

1437

1438 6.2.7.8 page-select (positiveIntegerRange)

1439 This attribute specifies the pages in the document that the
 1440 Printer shall ~~use~~. This attribute is unlikely to be useful for
 1441 jobs with more than one document or in Job Templates. If this
 1442 attribute is unspecified, then the Printer shall print ~~s~~ all pages
 1443 in a document.

1444 6.2.7.9 files-are-one-document (~~b~~Boolean)

1445 This attribute is relevant only if a job consists of two or more
 1446 documents. It controls finishing operations, job-sheet placement,
 1447 and the order of documents when the copies attribute exceeds 1.

1448 If the files for the job are a and b and this attribute is true,
 1449 then files a and b are treated as a single document for finishing
 1450 operations. Also, there will be no slip sheets between files a and
 1451 b. If more than one copy is made, the ordering must be a, b, a,
 1452 b, The attribute files-are-interleaved is ignored.

1453 If the files for the job are a and b ~~and this attribute is false~~
 1454 or unspecified by both the client and the Printer's Job Template,
 1455 then each file is treated as a single document for finishing
 1456 operations. Also, a client may specify that a slip sheet be

1457 between files a and b. If more than one copy is made, and the
1458 attribute files-are-interleaved false or unspecified, the ordering
1459 is a, a, b, b, If more than one copy is made, and the
1460 attribute files-are-interleaved true, the ordering is a, b, a, b,
1461

1462 6.2.7.10 files-are-interleaved (~~b~~Boolean)

1463 This attribute is used in conjunction with files-are-one-document
1464 (q.v.).

1465

1466 6.2.8 Attributes for Conversion of Text and HTML Files (Set by
1467 Client/End User)

1468 The client shall specify these attributes to control -formatting
1469 for text documents or HTML documents. ~~If the client does not~~
1470 ~~specify any of these attributes, a Printer shall uses its own~~
1471 ~~defaults.~~

1472 A client need not specify these attributes for other types of
1473 documents, such as PostScript or PCL.

1474 6.2.8.1 width (cardinalUnits)

1475 This attribute specifies the media width for the document in
1476 characters.

1477 6.2.8.2 length (cardinalUnits)

1478 This attribute specifies the media length for the document in
1479 characters.

1480 6.2.8.3 left-margin (cardinalUnits)

1481 This attribute specifies the left-margin for the document in
1482 characters.

1483 6.2.8.4 right-margin (cardinalUnits)

1484 This attribute specifies the right-margin for the document in
1485 characters.

1486 6.2.8.5 top-margin (cardinalUnits)

1487 This attribute specifies the top-margin for the document in lines.

1488 6.2.8.6 bottom-margin (cardinalUnits)

1489 This attribute specifies the bottom-margin for the document in
1490 lines.

1491 6.2.8.7 repeated-tab-stops (cardinalUnits)

1492 This attribute specifies the tab stops for the document in
1493 characters.

1494 6.2.8.8 header-text (string)

1495 This attribute specifies the header text for the document.

1496 6.2.8.9 footer-text (string)

1497 This attribute specifies the footer text for the document.

1498 6.2.8.10 number-pages (~~b~~Boolean)1499 This attribute specifies that the pages should be numbered in the
1500 document.

1501 6.2.8.11 default-font (string)

1502 This attribute specifies the font to use for all text in the
1503 document.

1504 6.2.8.12 font-size (cardinalUnits)

1505 This attribute specifies the font-size in points for text in the
1506 document. The value of this attribute affects the size of the
1507 other text attributes.1508 If this attribute is omitted and the Printer's default Job
1509 Template does not contain this attribute, the Printer shall assume
1510 a value of 10. A value of 10 with a fixed pitch font, shall
1511 produce 12 characters per inch in the horizontal direction and
1512 with 6 lines per inch in the vertical direction.

1513 6.2.8.13 default-code-set (type3Enum)

1514 This attribute specifies the code-set in which the document is
1515 encoded.

1516 6.2.8.14 content-orientation (type2Enum)

1517 This attribute specifies the orientation of the document.

1518 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
reverse- portrait	NOTE - The +90 direction was chosen because simple finishing on the long edge is the same edge whether portrait or landscape 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.

1519

1520 6.2.9 Job Resource Attributes (Set by the program that produces or
 1521 senses the PDL)

1522 A program described below shall add these attributes, which
 1523 describe the resources needed to print the job.

1524 A Printer may use these attributes to validate and schedule the
 1525 print-job without interpreting the contents of the document. This
 1526 provides the opportunity for a Printer to support a broad set of
 1527 document formats yet still support fast efficient scheduling and
 1528 validation of each job.

1529
 1530 The client/end user shall not specify these attributes. Instead,
 1531 it is the duty of the program that translates the document to the
 1532 printer's PDL (or analyzes it) to add these attributes and their
 1533 values to the job. Such a program may execute at a number of
 1534 different points in time:

1535 1. The program produces a final form document and stores these
 1536 resource attributes in a file before the end-user submits the
 1537 print job.

1538 2. The program produces a final form document data stream when
 1539 the end-user specifies "Print" to the application program
 1540 (e.g., Windows GDI driver).

1541 3. The program running in the context of the Printer or server
 1542 translates a revisable or final form document into a PDL that
 1543 the output device~~printer~~ understands.

1544 If any of these attributes is unspecified, the Printer shall
 1545 assume that the all resources required by the document of the type
 1546 specified by the missing attributes are ready, ie., are available
 1547 to the Printer and/or output device without human intervention.

1548 ISSUE - what does the above mean?

1549 These attributes may be unspecified if the translation program
 1550 fails to provides such values, or if no translation occurs (e.g.
 1551 the document is a PostScript document.

1552 Note: The Printer does not use these attributes during the actual
 1553 printing of a document.

1554 Note: these attributes allow more than one value wherever it is
1555 possible for a job to specify more than one value of the
1556 corresponding job attribute, possibly by embedded instructions.

1557 The client may specify these attributes in: Get-Attributes and
1558 Get-Jobs.

1559 See the section on job production attributes for an explanation of
1560 how the job resource attributes differ from the job production
1561 attributes.

1562 6.2.9.1 document-format~~s~~-used (1#type2Format)

1563 This attribute identifies the document formats needed to print the
1564 document(s) in this job.

1565 A format ~~consists~~ of two elements, a name and a version. The
1566 latter element is optional.

1567 The syntax is for type2Format:

1568 name ["/" version]

1569 Examples include: PostScript, PostScript/2.0 and PCL/5e

1570 Note: The version component is optional.

1571 The names shall be registered with IANA as "printer languages"
1572 following the procedures established by the Printer MIB (currently
1573 proposed as an IETF standard by RFC 1759).

1574 6.2.9.2 fonts-used (1#string)

1575 This attribute identifies the font resources used in the
1576 document(s) in the job.

1577 6.2.9.3 code-sets-used (1#type3Enum)

1578 This attribute identifies the code-sets used in the document(s) in
1579 the Job. This attribute is relevant only for files that are not in
1580 ASCII, such as text files and possibly PCL files. PostScript files
1581 are always ASCII. Normally there is at most 1 code-set.

1582 Standard values are defined in the section specifying the default-
1583 code-set attribute.

1584 6.2.9.4 media-used (1#type2Enum)

1585 This attribute identifies the media, media-sizes, input-trays or
1586 electronic forms needed to print the document(s) in the job.

1587 Standard values for this attribute are defined in the section
1588 specifying the medium-select attribute.

1589

1590 6.2.9.5 sides-used (type2Enum)

1591 This attribute specifies whether a job needs 1-sided, 2-sided-
1592 long-~~binding~~ edge, or 2-sided-short-~~binding~~ edge printing.

1593 Standard values for this attribute are defined in the section
1594 specifying ~~the~~ sides Job attribute.

1595 6.2.9.6 print-quality-used (type2Enum)

1596 This attribute specifies what print quality the job needs.

1597 Standard values for this attribute are defined in the section
1598 specifying the print-quality attribute.

1599 6.2.9.7 finishing-used (type2Enum)

1600 This attribute specifies what finishing the job needs.

1601 Standard values for this attribute are defined in the section
1602 specifying the finishing attribute.

1603 6.2.9.8 printer-resolution-used (positiveIntegerCrossState)

1604 This attribute specifies what resolution the job needs.

1605 The interpretation of the values for this attribute are defined in
1606 the section on printer-resolution-select Job attribute.

1607 6.2.9.9 total-job-octets (positiveInteger)

1608 This attribute specifies the total size of the job in octets. This
1609 attribute is the first of three that a translation program can use
1610 to specify the size of a job.

1611 6.2.9.10 job-impression-count (positiveInteger)

1612 This attribute specifies the total size of the job in impressions.

1613 6.2.9.11 job-media-sheet-count (positiveInteger)

1614 This attribute specifies the total size of the job in media-
1615 sheets.

1616 ~~6.2.9.12 job-intervening-jobs (positiveInteger)~~

1617 ~~This attribute indicates the number of jobs that are "ahead" of~~
1618 ~~this job in the current scheduled order. For efficiency, it is~~
1619 ~~only necessary to calculate this value when an operation is~~
1620 ~~performed that requests this attribute.~~

1621 6.2.10 Number of Documents (Set by PrinterClient)

1622 This group contains a single attribute which specifies the number
1623 of documents in the job.

1624 The Printer sets the value of this attribute depending on the
1625 number of documents that the client supplies in the ~~shall specify~~
1626 ~~this attribute in~~ Print operation. The client shall not specify
1627 this attribute (directly) in Print, but ~~and~~ may specify this
1628 attribute in: Get-Attributes and Get-Jobs.

1629 6.2.10.1 number-of-documents (positiveInteger)

1630 This attribute specifies the number of documents in the job. Each
1631 document shall contain its own set of document content attributes
1632 described below.

1633 6.2.11 Document Data (Set by a Client/End User)

1634 This group of attributes describes the document data for the job.
1635 These attributes also include the document data or reference it.

1636 All job attributes in other sections of this document occur only
1637 once per job and apply to all documents in a job.

1638 The client may specify document-data attributes in Print. The
1639 client must specify either the document-URL or document-content in
1640 Print.

1641 Except for document-content, the client may specify document-data
1642 attributes in: Get-Attributes, and Get-Jobs.

1643 6.2.11.1 document-format (type2Format)

1644 This attribute identifies the document format of this document.

1645 If the client does not specify this attribute, then the Printer
1646 shall attempt to determine the format in order to decide if the
1647 document data needs to be translated. The version component is
1648 optional.

1649 6.2.11.2 document-name (~~string~~name)

1650 This attribute contains the name of the document used by the
1651 client to initially identify the document.

1652 6.2.11.3 document-URL (~~url~~name)

1653 This attribute contains the URL of the document if the client
1654 specified the document with a URL.

1655 If this attribute is specified, then document-content shall be
1656 unspecified.

1657 6.2.11.4 document-content (octetString)

1658 This attribute contains the actual contents of the document.

1659 If this attribute is specified, then document-URL shall be
1660 unspecified.

1661 This attribute shall be used during the transmission of the Print
1662 operation over a network. A Printer shall save the document data
1663 to a file and reference it with the document-URL ~~or document path~~
1664 ~~attribute~~. A Get-Attribute or Get-Jobs operation shall always find
1665 that this attribute is unspecified.

1666 6.3 Operation Attributes (Set by Client)

1667 NOTE: These attributes have just been introduced and they are not
1668 as stable as the attributes in the other sections. Some work is
1669 still needed to show the relationship between these attributes,
1670 job attributes, printer attributes, and authentication and
1671 authorization.

1672 The client shall set these attributes and associate them with an
1673 operation rather than an object.

1674 It is intended that a client program rather than an end-user has
1675 control over the setting of these values so that they cannot be
1676 easily forged.

1677 6.3.1 operation-locale (type3Locale)

1678 This attribute identifies the locale of the client. The Printer
1679 uses this attribute to determine the locale of (1) messages in the
1680 result of the operation, (2) ~~or~~ in errors returned by the operation
1681 or (3) notification events sent to the submitter.

1682 The standard values are defined in the section on the job-locale
1683 attribute.

1684 If an operation does not specify this attribute, the Printer shall
1685 assume that the operation has the same locale as the Printer.

1686 6.3.2 operation-notification-address (url)

1687 This attribute ~~specifies~~identifies ~~the~~ both the address and
1688 mechanism for delivery of events. If the URL has a "mailto:"
1689 scheme, then email is used and the rest of the URL is used as the
1690 email address. If the URL has a "http:" scheme, then an HTTP
1691 APPEND method is used to add HTML formatted events to the end of
1692 ~~the~~a specified HTML file.

1693 6.3.3 operation-user-name (name)

1694 This attribute identifies the most authenticated end user name
1695 that the client can supply. This name identifies the end user
1696 performing the operation.

1697 This value shall be set by the system rather than the end-user in
1698 order to minimize the chance of forgery.

1699 6.3.4 operation-host-name (name)

1700 This attribute identifies the most authenticated host name that
1701 the client can supply. This name identifies the host from which
1702 the operation comes.

1703 This value shall be set by the system rather than the end-user in
1704 order to minimize the chance of forgery.

1705 6.4 Printer Attributes (Set by the Administrator)

1706 A printer object may be realized in either a Print Server or
1707 Output Device. Note: How these attribute are set by an
1708 Administrator is outside the scope of this specification.

1709 A Printer Object in an Output Device ~~contains~~ a set of printer
1710 object attributes that represent an Output Device capable of
1711 rendering a document in visible form. Examples include electronic
1712 and electro-mechanical printers such as laser printers, ink-jet
1713 printers, and various kinds of impact printers, but may include
1714 other types of output devices such as microfiche imagers and
1715 plotters as well.

1716 A Printer Object in a Print Server ~~maythat~~ supply~~ies~~ queuing,
1717 spooling, and scheduling for an Output device that does not queue
1718 or spool.

1719 A Print Server, in the most common case, controls exactly one
1720 downstream Output Device. The Print Server's Printer object has
1721 attributes whose values are the same as those of the Printer
1722 object in the downstream Output Device.

1723 A Printer Object in a Print Server ~~may~~ contains a set of printer
1724 object attributes that are the union of the Printer objects in the
1725 downstream Output Devices. This object extends the capabilities
1726 of an Output Device. For example, an administrator might define a
1727 single Print Server to represent all of the Output Devices of the
1728 same type and capability in a single location, associated with a
1729 particular server. A end user would normally send a print-job to
1730 a Print Server~~,~~ and allow the Print Server to assign the job to a
1731 particular Output Device based on the relative load and
1732 availability of the printers under its control, thus providing a
1733 load balancing service. However, nothing precludes an
1734 administrator from configuring a print system so that ~~a~~ end user
1735 can send a print-job directly to an Output Device .

1736 ~~A Print Server, in the most common case, controls exactly one~~
1737 ~~downstream Output Device. The Print Server's Printer object has~~
1738 ~~attributes whose values are the same as those of the Printer~~
1739 ~~object in the downstream Output Device.~~

1740 The attributes defined in this section ~~provide~~ information about
1741 a particular Printer.

1742 6.4.1 printer-name (name)

1743 This attribute uniquely identifies the printer on its host.

1744 6.4.2 printer-location (string)

1745 This attribute identifies the location of this printer.

1746 6.4.3 printer-model (string)

1747 This attribute identifies the make and model of the printer.

1748 6.4.4 printer-types (type2Enum)

1749 This attribute identifies the marking technology of the printer.

1750 The standard values for this attribute are the descriptive names
 1751 specified by ISO DPA which have corresponding enum symbolic and
 1752 numeric values assigned by the Printer MIB (RFC 1759).. These
 1753 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

1754

1755 6.4.5 printer-state (type1Enum)

1756 This attribute identifies the current state of the printer and
 1757 shall be set by the Printer. The protocol support all values for
 1758 printer states, however a Printer shall only generate the printer
 1759 states which are appropriate for the particular implementation.

1760 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 <u>IPP V1.0</u> this part of ISO/IEC 10175 .
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.

1761

1762 6.4.6 printer-state-message (string)

1763 This attributes specifies a message that gives further information
 1764 about the current printer state and shall be set by the Printer.—

1765 6.4.7 message (string)

1766 This attribute provides a message from an operator, system
 1767 administrator or "intelligent" process to indicate to the end user

1768 information or status of the printer, such as why it is
1769 unavailable or when it is expected to be available. .

1770 6.4.8 printer-job-templates (1#urlDefault)

1771 This attribute identifies the URL of each of the Job Templates
1772 that this Printer is associated with and the one Job Template this
1773 Printer uses as its default for supply job attributes that the
1774 client omits. There shall be only one value with the default
1775 qualifier. Other Printers can be associated with the same Job
1776 Templates.

1777 The syntax is:

1778 url [":" default]

1779 6.4.9 locale (type3Locale)

1780 This attribute specifies the locale that the Printer operates in.

1781 The standard values are defined in the section on the job-locale
1782 attribute.

1783 6.4.10 notification-events (1#type2Enum)

1784 This attribute specifies the events on whose occurrence the
1785 Printer should notify those addresses specified by the
1786 notification-addresses attribute.

1787 If the attribute is unspecified ~~or empty~~, the Printer does not
1788 perform notification, though the Printer still checks the job's
1789 notification-events attribute.

1790 In this attribute, job-problem and printer-problem have the same
1791 meaning.

1792 The standard values are defined in the section on the job's
1793 notification-events attribute.

1794 NOTE - This attribute is intended to notify operators, not end-
1795 users.

1796 6.4.11 notification-addresses (1#urlname)

1797 This attribute specifies the ~~method and email~~ addresses to which
1798 the Printer should send messages when events specified by the
1799 notification-~~events~~ attribute occur.

1800 If the attribute is unspecified ~~or empty~~, the Printer does not
1801 perform notification, though the Printer still checks the job's
1802 notification-events attribute.

1803 NOTE - This attribute is intended to notify operators, not end-
1804 users.

1805 6.4.12 end-user-acl (1#name)

1806 This attribute specifies the end users who are allowed to print on
1807 the Printer.

1808 If the attribute is unspecified ~~or empty~~, the Printer allows
1809 anyone to print.

1810 6.4.13 maximum-printer-speed (positiveIntegerUnits)

1811 This attribute indicates the maximum printer speed of the Printer
1812 in units of pages per minute, impressions per minute, lines per
1813 minute, and characters per minute. A job cannot control a
1814 Printer's speed, but a Printer Browser can use printer speed as a
1815 criteria.

1816 The standard units are a type2Enum and are: ppm, ipm, spm, lpm,
1817 cps.

1818 6.4.14 fonts-substitutions (1#stringPair)

1819 This attribute specifies an appropriate substitute for a font that
1820 is advertised as supported in the fonts-supported attribute, even
1821 though the Printer doesn't actually have the font available.

1822 This attribute consists of a set of font pairs: a font name and
1823 the font to use instead.

1824 If this attribute is unspecified, the Printer does not perform any
1825 font substitutions.

1826 6.4.15 fonts-supported (1#stringState)

1827 This attribute identifies the font resources supported by this
1828 printer and indicates the state of readiness for each font.

1829 The standard names are defined in the section on default-font.

1830 Each item in the list contains the pair consisting of a font name
1831 and a state indicating the font's readiness state.

1832 6.4.16 media-supported (1#nameState)

1833 This attribute identifies the media, media-sizes, input trays, and
1834 electronic forms supported by this printer, and indicates the
1835 state of readiness for each medium resource.

1836 The standard names are defined in the section on the section on
1837 the medium-select.

1838 Standard states are: not-ready, on-order, and special-order. The
1839 omission of a state shall indicate that the medium is ready, i.e.,
1840 can be used without human intervention.~~There may be just two~~
1841 ~~states: ready and needs installing, or there may be a third state:~~
1842 ~~needs purchasing.~~

1843 6.4.17 document-formats-supported (1#type2FormatState)

1844 This attribute identifies the document-formats, including the
1845 document-format-versions, supported by the Printer. This set
1846 includes both the formats that are native to the Printer and
1847 those formats that the Printer can translate to one that is
1848 native to the Printer. From the client's point of view, this set
1849 contains all formats in which documents can be submitted to this
1850 Printer.

1851 Proprietary document format identifiers, and versions are assigned
1852 by the owners of those formats.

1853 The state of readiness for each format is also included, though
1854 all formats should normally always be ready.

1855 6.4.18 numbers-up-supported (1#~~type3EnumState~~positiveIntegerState)

1856 This attribute identifies the number-up values supported by this
1857 printer..

1858 The state of readiness for each number-up value is also included,
1859 though all number-up conversions should always be ready.

1860 6.4.19 finishings-supported (1#type2EnumState)

1861 This attribute identifies the finishing operations ~~supported~~ by
1862 this Printer and states of readiness for each finishing.

1863 The standard finishing objects are defined in the section on the
1864 finishing Job attribute.

1865 6.4.20 sides-supported (1#type2EnumState)

1866 This attribute indicates the values of the sides attribute
1867 supported by this printer and the states of readiness of each
1868 value.

1869 The standard values are defined in the section on the sides
1870 attribute.

1871 6.4.21 print-qualities-supported (1#type2EnumState)

1872 This attribute indicates the values of the printer-quality
1873 attribute supported by this printer and the states of readiness
1874 for each print-quality value.

1875 The standard values are defined in the printer-quality attribute.

1876 6.4.22 printer-resolutions-supported (1#positiveIntegerCrossState)

1877 This attribute indicates the values of the printer-resolution-
1878 select attribute supported by this printer and their states of
1879 readiness.

1880 The state of readiness for each printer resolution is also
1881 included, though normally all printer-resolutions should always be
1882 ready.

1883 The syntax is discussed in the section on the printer-resolution-
1884 select attribute.

1885 6.4.23 code-sets-supported (1#type3EnumState)

1886 This attribute indicates the values of the default-code-set
1887 attribute supported by this printer and the states of readiness
1888 for each code-set.

1889 The standard values are defined in the default-code-set attribute.

1890 6.4.24 off-peak-times-supported (1#type3EnumState)

1891 This attribute indicates the values of the job-print-off-peak
1892 attribute supported by this printer and the states of readiness
1893 for each value.

1894 If this attribute is unspecified, then the Printer has no off-peak
1895 periods.

1896 The standard values are defined in the section on the job-print-
1897 off-peak Job attribute.

1898 Note: this document does not define how an administrator
1899 associates the off-peak names with actual time periods.

1900 6.4.25 events-supported (1#type2EnumState)

1901 This attribute indicates the values of the job and printer
1902 notification-events attribute supported by this Printer and the
1903 states of readiness for each value.

1904 If this attribute is unspecified, then the Printer does not
1905 support notification.

1906 The standard values are defined in the section on the
1907 notification-events attribute.

1908 6.4.26 locales-supported (1#type3LocaleState)

1909 This attribute indicates the values of the job-locale attribute
1910 supported by this Printer and the states of readiness for each
1911 value.

1912 The standard values are defined in the section on the job-locale
1913 attribute.

1914 6.4.27 job-sheets-supported (1#type3EnumState)

1915 This attribute identifies the job-sheet ~~values~~ supported by this
1916 printer, and the state of readiness for each job-sheet.

1917 To allow no job sheets, the system administrator shall include the
1918 value "none" as a value for this attribute. The client specifies
1919 that there are no job sheets by using the value "none" as the
1920 value of the job-sheets attribute.

1921 If the job-sheets attribute is not specified or contains a value
1922 which the Printer does not support, then the server shall select
1923 from among the values of this attribute. The server shall not
1924 | select the value "none" unless it is the only value specified for
1925 the job-sheets-supported attribute.

1926 | NOTE - When the client supplies a value other than "none", it is
1927 preferable for the server to produce some job jobsheet, even if
1928 not the desired one, rather than produce none at all or reject the
1929 job.

1930
1931 6.4.28 maximum-copies (positiveInteger)

1932 This attribute indicates the maximum number of copies of a
1933 document that can be rendered by this printer in a single print-
1934 job.

1935 | If the attribute is unspecified ~~or has a value of 0~~, there is no
1936 limit on the maximum number of copies for this Printer.

1937 6.4.29 maximum-job-octets (positiveInteger)

1938 This attribute indicates that the Printer shall accept a job only
1939 if its size in octets is less than the value specified by this
1940 attribute.

1941 | If the attribute is unspecified ~~or has a value of 0~~, there is no
1942 limit on the size of a job in octets.

1943 6.4.30 maximum-impressions (positiveInteger)

1944 This attribute indicates that the Printer shall accept a job only
1945 if its size in impression is less than the value specified by this
1946 attribute.

1947 | If the attribute is unspecified ~~or has a value of 0~~, there is no
1948 limit on the size of a job in impressions.

1949 6.4.31 maximum-media-sheets (positiveInteger)

1950 This attribute indicates that the Printer shall accept a job only
1951 if its size in media-sheets is less than the value specified by
1952 this attribute.

1953 | If the attribute is unspecified ~~or has a value of 0~~, there is no
1954 limit on the size of a job in media-sheets.

1955 6.4.32 maximum-job-retention-period (deltaTime)

1956 This attribute indicates that when the Printer accepts a job, the
1957 retention period must not exceed the value of this attribute.
1958 Otherwise, the Printer sets the job's retention-period to the
1959 value of this attribute.

1960 If this attribute is unspecified, then the Printer places no limit
1961 on the retention time.

1962 6.4.33 maximum-end-user-priority (type1Enum)

1963 This attribute indicates that when the Printer accepts a job, the
 1964 job-priority must not exceed the value of this attribute.
 1965 Otherwise, the Printer sets the job's job-priority to the value of
 1966 this attribute.

1967 If this attribute is unspecified, then the Printer places no limit
 1968 on the job-priority~~time~~.

1969 The standard values are defined in the section on the job-priority
 1970 attribute.

1971 6.4.34 queued-job-count (~~cardinalpositiveInteger~~)

1972 This attribute contains a count of the number of jobs that are
 1973 either pending and/or processing and shall be set by the Printer.

1974 6.4.35 scheduling-algorithm (type3Enum)

1975 This attribute indicates the current scheduling algorithm for this
 1976 Printer. Standard values are: "none", "~~smallestshortest~~-job-
 1977 first", "time-received", ~~etc~~.

1978 6.5 Job Templates

1979 The attributes for a Job Template can be any of the Job object
 1980 attributes defined in the sections:

1981 Job Sheet Attributes
 1982 Notification Attributes
 1983 Job Scheduling Attributes
 1984 (except job-print-after)
 1985 Job Production Attributes
 1986 (except page-select)
 1987 Attributes for Conversion of Text and HTML Files
 1988

1989 6.6 Conformance

1990 A conforming implementation shall implement all operations,
 1991 objects and attributes defined in this document. ~~IPP is explicitly
 1992 designed to be extensible. This means that in addition to the
 1993 attributes defined in this specification, specific implementation
 1994 instances may support not only the basic protocol as defined in
 1995 this specification, but might add vendor specific extensions.~~

1996 Also, for the core set of attributes listed in this specification,
 1997 it is not required that a conforming server support all (standard)
 1998 values of all supported attributes. For example, it is not
 1999 required that a printer implement all finishing methods indicated
 2000 by the standard values.

2001 The explicit requirement of the term "supported", with respect to
 2002 one of the attributes that deal with printer functions or
 2003 resources, is that the server shall recognize the attribute and
 2004 those values that are supported, and shall be able to respond to a
 2005 query about which values that printer does, in fact, support.

2006 | IPP is explicitly designed to be extensible. Additional
2007 | attributes can be proposed to be registered by going through the
2008 | type 2 enum process which will register their specification after
2009 | approval with IANA. In addition specific implementation instances
2010 | may support not only the basic protocol as defined in this
2011 | specification, but may add vendor-specific private extensions by
2012 | prefixing attribute-names with their company name registered with
2013 | IANA for use in domains. See attribute syntax section. However,
2014 | such private extensions shall not duplicate attribute semantics
2015 | already in this specification.

2016 7. Security Considerations

2017 This protocol does not identify any new authentication mechanisms.
2018 The authentication mechanisms built into HTTP (such as SSL and
2019 SHTTP) are recommended.

2020 This protocol does define a simple authorization mechanism by
2021 introducing the "end-user-acl" attribute as part of the Printer
2022 object. This ACL attribute is a multi-valued list of all of the
2023 authenticated names of end-users. This protocol does not specify
2024 what the domain is for names in this ACL attribute.

2025 Issue: Will it always be possible for a Printer to obtain a
2026 meaningful authenticated name that the Printer can match against
2027 the end-user-acl, or will some other mechanism be necessary, such
2028 as a password?

2029 8. References

- 2030 [1] Smith, R., Wright, F., Hastings, T., Zilles, S., and
2031 Gyllenskog, J., "Printer MIB", RFC 1759, March 1995.
2032
2033 [2] Berners-Lee, T, Fielding, R., and Nielsen, H., "Hypertext
2034 Transfer Protocol - HTTP/1.0", RFC 1945, August 1995.
2035
2036 [3] Crocker, D., "Standard for the Format of ARPA Internet Text
2037 Messages", RFC 822, August 1982.
2038
2039 [4] Postel, J., "Instructions to RFC Authors", RFC 1543, October
2040 1993.
2041
2042 [5] ISO/IEC 10175 Document Printing Application (DPA), Final,
2043 June 1996.
2044
2045 [6] Herriot, R. (editor), X/Open A Printing System
2046 Interoperability Specification (PSIS), August 1995.
2047
2048 [7] Kirk, M. (editor), POSIX System Administration - Part 4:
2049 Printing Interfaces, POSIX 1387.4 D8, 1994.
2050
2051 [8] Borenstein, N., and Freed, N., "MIME (Multi-purpose Internet
2052 Mail Extensions) Part One: Mechanism for Specifying and
2053 Describing the Format of Internet Message Bodies", RFC 1521,
2054 September, 1993.
2055
2056 [9] Braden, S., "Requirements for Internet Hosts - Application
2057 and Support", RFC 1123, October, 1989,
2058

2059 [10] McLaughlin, L. III, (editor), "Line Printer Daemon Protocol"
2060 RFC 1179, August 1990.

2061

2062

2063 9. Author's Address

2064 Scott A. Isaacson

2065 Novell, Inc.

2066 122 E 1700 S

2067 Provo, UT 84606

2068

2069 Phone: 801-861-7366

2070 Fax: 801-861-4025

2071 EMail: scott_isaacson@novell.com

2072

2073 Tom Hastings

2074 Xerox Corporation

2075 701 S. Aviation Blvd.

2076 El Segundo, CA 90245

2077

2078 Phone: 310-333-6413

2079 Fax: 310-333-5514

2080 EMail: hastings@cpl0.es.xerox.com

2081

2082 Robert Herriot

2083 Sun Microsystems Inc.

2084 2550 Garcia Ave., MPK-17

2085 Mountain View, CA 94043

2086

2087 Phone: 415-786-8995

2088 Fax: 415-786-7077

2089 Email: robert.herriot@eng.sun.com

2090

2091 Roger deBry

2092 HUC/003G

2093 IBM Corporation

2094 P.O. Box 1900

2095 Boulder, CO 80301-9191

2096

2097 Phone: (303) 924-4080

2098 Fax: (303) 924-9889

2099 Email: debry@vnet.ibm.com

2100

2101 Other Contributors

2102 Devon Taylor, Novell, Inc.

2103 Mike MacKay, Novell, Inc.

2104 Peter Zehler, Xerox, Corp.

2105 Keith Carter, IBM Corporation

2106 Carl-Uno Manros, Xerox, Corp.

2107 <add the list of IPP attendees and participants>

2108

2109

2110 10. Appendix A: Sample IPP Operations

2111 The following examples illustrate typical flows using the IPP
2112 protocol. In these examples, the IPP Printer object named
2113 "printer-1" is located at the node identified by the DNS name
2114 "some.domain.com". AJjob Template has been defined for printer-1
2115 which establishes the print defaults.

2116 For brevity in the following flows, none of the HTTP headers are
2117 shown. CRLF sequences are not shown.

2118 10.1 Querying the printer

2119 Client some.domain.com

2120 ----->

2121 Post http://some.domain.com/printer-1 http/1.0

2122 GetAttributes IPP/1.0

2123 Printer-state :

2124 Sides-supported :

2125 Media-supported :

2126 Document-formats-supported :

2127

2128 <-----

2129

2130 http/1.0 201 "Created" (a response)

2131 IPP/1.0 xxx "attribute list returned"

2132 Printer-state : idlerunning

2133 Sides-supported : 1-sided

2134 Media-supported : iso-a4-white, iso-b4-white

2135 Document-formats-supported : Postscript/2.0

2136

2137

2138 10.2 Print Operation - with print data included

2139 Client some.domain.com

2140 ----->

2141 Post http://some.domain.com/printer-1 http/1.0

2142 Print IPP/1.0

2143 Print-Job-Object Header

2144 Job-name : My Job

2145 Medium : iso-a4-white

2146 Notification-events : Job-completion

2147 Notification-address : joe@pc.domain.com

2148 Document Header

2149 Document-name : Letter to Mom

2150 Document-Content Header (content type = Postscript/2.0)

2151 Document in Postscript level 2 format

2152

2153

2154 <-----

2155 http/1.0 200 "accepted"

```
2156      IPP/1.0 xxx "print job accepted and queued"
2157      Job-Identifier : some.domain.com/printer-1/0037
2158      Current-job-state : pending
2159      Printer-state : needs-attentionrunning
2160
2161 10.3 Print Operation - with no data included

2162      Client                                some.domain.com
2163
2164      ----->
2165
2166      Post http://some.domain.com/printer-1 http/1.0
2167      Print IPP/1.0
2168      Print-Job-Object Header
2169      Job-name : My Job
2170      Medium : iso-a4-white
2171      Notification-events : Job-completion
2172      Notification-address : joe@some.domain.com
2173      Document Header
2174      Document-name : Letter to Mom
2175      Document-URL : joe@pc.domain.com/Docs/To-mom.ps
2176
2177      <-----

2178      http/1.0 200 "accepted"
2179      IPP/1.0 xxx "print job accepted and queued"
2180      Job-Identifier : some.domain.com/printer-1/0037
2181      Current-job-state : pending
2182      Printer-state : processingrunning
2183
2184 10.4 Querying the state of the job

2185      In this example, no attributes are specified, so all job
2186      attributes are returned.

2187      Client                                some.domain.com
2188
2189      ----->
2190
2191      Post http://some.domain.com/printer-1/0037 http/1.0
2192      GetAttributes IPP/1.0
2193
2194      <-----

2195      http/1.0 201 "Created" (a response)
2196      IPP/1.0 xxx "atribute list returned"
2197      Job-Name : My Job
2198      Job-Originator : Joe@some.domain.com
2199      Job-originating-host : pc.domain.com
2200      Notification-address : joe@pc.domain.com
2201      Job-locale : xx:xx:xx
2202      Current-job-status : printing
2203      Printer-assigned : printer-1
2204      Submission-time : 1996 Nov 22 1214
2205      Media-sheets-completed : 2
2206
2207 10.5 Canceling a Job

2208      Client                                some.domain.com
```

```
2207 ----->
2208 Post: http://some.domain.com/printer-1/0037
2209     CancelJob IPP/1.0
2210
2211 <-----
2212
2213 http/1.0 200 "okay"
2214     Current-job-state : terminating
2215
2216 10.6 Listing jobs on a Printer
2217
2218 List jobs on printer-1, only return job sizes. Jobs are returned
2219 in the order they are scheduled for printing. A Job-identifier
2220 attribute preceeds the attributes returned for each job to delimit
2221 job boundaries.
2222
2223 Client some.domain.com
2224
2225 ----->
2226 Post http/1.0 some.domain.com/printer-1
2227     GetJobs IPP/1.0
2228     total-job-octets :
2229
2230 <-----
2231 http/1.0 201 "Created" (a response)
2232     IPP/1.0 xxx "created an attribute list"
2233     Job-identifier : 0033
2234     total-job-octets : 4567
2235     Job-identifier : 0034
2236     total-job-octets : 12345
2237     Job-identifier : 0035
2238     total-job-octets : 12356
2239
```