

1 | ~~Internet Draft~~

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~~ draft-isaacs on-ipp-info-00.txt
Expires May 27, 1997

20 Status of this Memo

21 This document is an Internet-Draft. Internet-Drafts are working
22 documents of the Internet Engineering Task Force (IETF), its areas,
23 and its working groups. Note that other groups may also distribute
24 working documents as Internet-Drafts.

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

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

34 Abstract

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

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

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

Table of Contents

1. Introduction	Bookmark not defined	4
2. Distributed Printing		5
2.1 Generic Print System Components		5
2.2 IPP Components		6
3. IPP Objects		6
3.1 Printer		6
3.2 Job		8
3.3 Job Template		9
3.4 Object Relationships		9
3.5 Object Identity		9
4. Naming		10
4.1 Directory Services		11
4.2 Directory Entry Schema		11
4.2.1 Name		11
4.2.2 Description		12
4.2.3 Location		12
4.2.4 Maximum Print Quality		12
4.2.5 Cost		12
4.2.6 Resolution		12
4.2.7 Color Supported		12
4.2.8 Fonts Supported		12
4.2.9 Maximum Speed		13
4.2.10 Device Id		13
4.2.11 Make and Model		13
4.2.12 Marker Type		13
4.2.13 Document Formats Supported		13
4.2.14 Sides Supported		13
4.2.15 Finishings Supported		14
4.3 Directory Entries Using LDAP		14
5. IPP Operations		15
5.1 HTTP Overview		15
5.2 IPP Operation Encoding		16
5.2.1 HTTP Request-Header Fields		17
5.2.2 HTTP Response-Header Fields		17
5.3 The Print Job		18
5.3.1 Print Job Object Header		18

98	5.3.2 Document Header	18
99	5.3.3 Document-Content Header	19
100	5.3.4 Job Attributes	19
101	5.3.5 Document Attributes	19
102	5.4 Operation Semantics	19
103	5.4.1 Print Operation	20
104	5.4.2 Cancel Job Operation	21
105	5.4.3 Get Attributes Operation	21
106	5.4.4 Get Jobs Operation	22
107	6. Object Attributes	23
108	6.1 Attribute Syntaxes	23
109	6.2 Job Attributes	26
110	6.2.1 Job Informational Attributes (Set by a Client/End User)	26
111	6.2.2 Job Informational Attributes (Set by a Printer)	27
112	6.2.3 Job Status Attributes (Set by Printer)	28
113	6.2.4 Job Sheet Attributes (Set by Client/End User)	32
114	6.2.5 Notification Attributes (Set by a Client/End User)	32
115	6.2.6 Job Scheduling Attributes (Set by Client/End User)	33
116	6.2.7 Job Production Attributes (Set by Client/End User)	35
117	6.2.8 Attributes for Conversion of Text and HTML Files (Set by	
118	Client/End User)	44
119	6.2.9 Job Resource Attributes (Set by the program that produces or	
120	senses the PDL)	46
121	6.2.10 Number of Documents (Set by Printer)	49
122	6.2.11 Document Data (Set by a Client/End User)	49
123	6.3 Operation Attributes (Set by Client)	50
124	6.3.1 operation-locale (type3Locale)	50
125	6.3.2 operation-notification-address (url)	51
126	6.3.3 operation-user-name (name)	51
127	6.3.4 operation-host-name (name)	51
128	6.4 Printer Attributes (Set by the Administrator)	51
129	6.4.1 printer-name (name)	52
130	6.4.2 printer-location (string)	52
131	6.4.3 printer-model (string)	52
132	6.4.4 printer-type (type2Enum)	52
133	6.4.5 printer-state (type1Enum)	53
134	6.4.6 printer-state-message (string)	54
135	6.4.7 message (string)	54
136	6.4.8 printer-job-templates (1#urlDefault)	54
137	6.4.9 locale (type3Locale)	55
138	6.4.10 notification-events (1#type2Enum)	55
139	6.4.11 notification-addresses (1#url)	55
140	6.4.12 end-user-acl (1#name)	55
141	6.4.13 maximum-printer-speed (positiveIntegerUnits)	55
142	6.4.14 fonts-substitutions (1#stringPair)	56
143	6.4.15 fonts-supported (1#stringValue)	56
144	6.4.16 media-supported (1#nameState)	56
145	6.4.17 document-formats-supported (1#type2FormatState)	56
146	6.4.18 numbers-up-supported (1#type3EnumState)	57
147	6.4.19 finishings-supported (1#type2EnumState)	57
148	6.4.20 sides-supported (1#type2EnumState)	57
149	6.4.21 print-qualities-supported (1#type2EnumState)	57

150	6.4.22 printer-resolutions-supported (1#positiveIntegerCrossState)	57
151		
152	6.4.23 code-sets-supported (1#type3EnumState)	57
153	6.4.24 off-peak-times-supported (1#type3EnumState)	58
154	6.4.25 events-supported (1#type2EnumState)	58
155	6.4.26 locales-supported (1#type3LocaleState)	58
156	6.4.27 job-sheets-supported (1#type3EnumState)	58
157	6.4.28 maximum-copies (positiveInteger)	59
158	6.4.29 maximum-job-octets (positiveInteger)	59
159	6.4.30 maximum-impressions (positiveInteger)	59
160	6.4.31 maximum-media-sheets (positiveInteger)	59
161	6.4.32 maximum-job-retention-period (deltaTime)	59
162	6.4.33 maximum-end-user-priority (type1Enum)	60
163	6.4.34 queued-job-count (cardinal)	60
164	6.4.35 scheduling-algorithm (type3Enum)	60
165	6.5 Job Templates	60
166	6.6 Conformance	60
167	7. Security Considerations	61
168	8. References	61
169	9. Author's Address	62
170	10. Appendix A: Sample IPP Operations	64
171	10.1 Querying the printer	64
172	10.2 Print Operation - with print data included	64
173	10.3 Print Operation - with no data included	65
174	10.4 Querying the state of the job	65
175	10.5 Canceling a Job	66
176	10.6 Listing jobs on a Printer	66
177		

178 1. Introduction

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

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

- 192 - "shall": indicates an action that the subject of the sentence
 193 must implement in order to claim conformance to this specification
- 194 - "may": indicates an action that the subject of the sentence does
 195 not have to implement in order to claim conformance to this
 196 specification, in other words that action is an implementation
 197 option

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

2. Distributed Printing

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

2.1 Generic Print System Components

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

- End Users: End Users are humans (or agents or applications who work on behalf of a human) who submit print jobs.
- Print clients: Print clients are computer network nodes with which humans interact in order to manipulate the distributed print service. A print client uses some protocol to invoke print service operations on another node. Each operation has arguments and results associated with it. The print client provides arguments which add information about the operation requested, and receives results which describe the status and outcome of the operation.
- Print servers: Print ~~ex~~ servers may be embedded in an output device or implemented in a separate system which is associated with an output device. The print server receives requests from the print client and sends back results which describe the status and outcome of the operation requested. A print server normally provides queuing, job management, and device management functions.
- Queues: Print jobs may be queued or stored on a spool prior to printing. This allows a print service provider to accept one or more print jobs while the printer (or printers) is busy processing another job. Queues, if present, may be implemented in the client, in the server, in the output device, or in some combination of the three.

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

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

2.2 IPP Components

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

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

Clients use the following operations:

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

3. IPP Objects

This section describes the IPP objects.

3.1 Printer

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

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

285 Some examples of configurations containing IPP Printer object
286 include:

- 287 - An output device, with no spooling capabilities, supporting IPP
- 288 | - An output device, with a built-in spooler, supporting IPP
- 289 | - ~~IPP~~
- 290 | -
- 291 - A print server with one or more associated output devices with
292 the print server supporting IPP.
- 293 - The associated output devices may or may not be capable of
294 spooling jobs
- 295 - The associated output devices may or may not support IPP
- 296

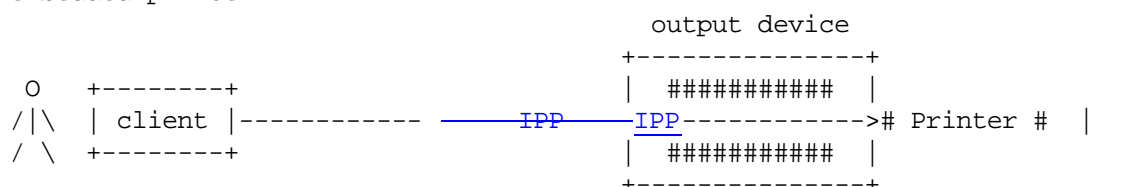
297 See the following figures for some examples on how to view IPP
298 Printer objects on top of other printing system models:

Legend:

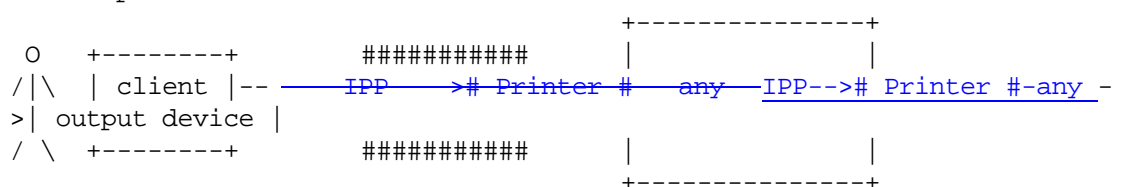
indicates an IPP ~~printer object which is either embedded~~ Printer object which is
~~in an output device or is hosted in a server. An IPP either~~
~~embedded in an output device or is~~
~~printer object hosted in a server. An IPP Printer object~~
 may or may not queue/spool.

any indicates any network protocol or direct ~~connect, including~~
connect, including IPP

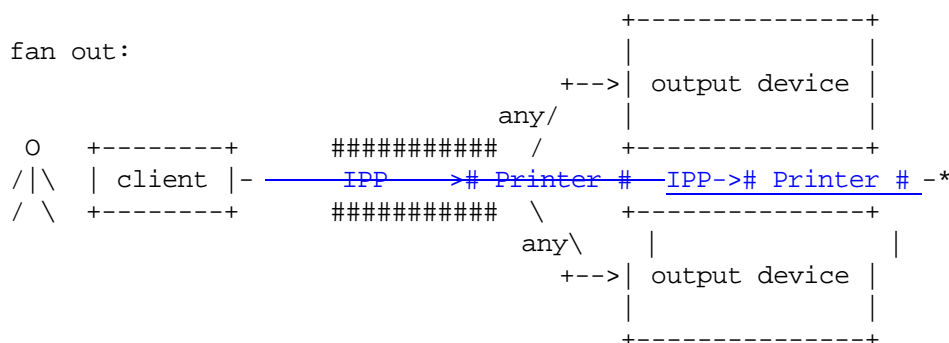
embedded printer:



hosted printer:



fan out:



346 3.2 Job

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

355 |
356 | Job attributes are broken up into the following groups:

- 357 | - Job Informational (sections 6.2.1, 6.2.2)
- 358 | - Job Status (section 6.2.3)
- 359 | - Job Sheet (section 6.2.4)
- 360 | - Notification (section 6.2.5)
- 361 | - Job Scheduling (section 6.2.6)
- 362 | - Job Production (section 6.2.7)
- 363 | - Conversion of Text Files (section 6.2.8)
- 364 | - Job Resource s (section 6.2.9)
- 365 | - Number of Documents (section 6.2.10)
- 366 | - Document Attributes (6.2.11)

368 3.3 Job Template

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

372 | ~~ISSUE~~Issue: The notion of Job Template needs more work. ~~It is~~
373 | ~~currently believed that when a client needs to present a Print Dialog~~
374 | ~~box to an end user, it gets potential job values and default job~~
375 | ~~values from a Printer. The default values are from the Job Template~~

376 | ~~associated with the Printer named by the end user. If a end user~~
377 | ~~sends a job to a Printer, the Printer may set unspecified attributes~~
378 | ~~to the value of the associated Job Template.~~

379 | 380 3.4 Object Relationships

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

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

388 3.5 Object Identity

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

391 The following objects have the following mandatory identifier
 392 attributes:

<u>Object</u>	<u>Identifier</u>	<u>Containing Object</u>
<u>Printer</u>	<u>printer-name</u>	<u>None</u>
<u>Job</u>	<u>job-identifier</u>	<u>Printer</u>
<u>Job Template</u>	<u>job-template-name</u>	<u>None</u>
<u>Identifier</u>		<u>Containing Object</u>

393
 394
 395 ~~Printer printer name None~~
 396 ~~Job job identifier Printer~~
 397 ~~Job Template job template name None~~
 398

399 4. Naming

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

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

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

407 Another example is the following URL:

408 http://1.2.3.4: ~~380~~nnn/printer-2

409 In this case, the URL identifies the use of the HTTP protocol. The
 410 Printer is located at the node identified by the IP address of
 411 "1.2.3.4" using port ~~380~~nnn for the HTTP server, and "printer-2" is
 412 the name of the Printer. (The actual value of nnn is to be assigned
 413 by IANA as part of this standards project).

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

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

- 420 1) http://some.domain.com/two-sided-printer, and
 421 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 ~~the the associated Job of the Job Templates~~ Templates would have two specified by a URL would contain a different Job Template default attribute set s-
. One Job Template would contain the defaults for two-sides printing and the other would contain the defaults for draft printing.

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

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

~~in: http://some.domain.com/hr-printer/resumes. In this case there is a "resumes" Job Template~~ In this case there are "resumes" and "1040forms" Job Templates associated with the "hr-printer" Printer.

This specification establishes, through IANA, a new well known port, port 380nnn, 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 HTTPIPP printing would be acceptable on port 380nnn.

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

465 explicit, quantified, measurable values? Some of the issues are: end
466 users don't often know what explicit objective values are or what
467 they really mean and they want to depend on an administrator to
468 define what is "high" quality printing and what is "low" quality,
469 especially since today's objective values that equate to "high" are
470 tomorrow's objective values that equate to "medium". On the other
471 hand, some end users demand the control and power explicit values can
472 give them when they do filtered searching. For example, they know
473 and appreciate the difference between 20 ppm printers and 23 ppm
474 printers.

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

479 4.2.1 Name

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

483 4.2.2 Description

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

486 4.2.3 Location

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

489 In order for filtered searches to be more effective, a given site may
490 use some regular structuring within the string values such as
491 "SITE:USA-San Jose,BUILDING:A1,FLOOR:2,ROOM:555" or "department5-
492 2ndFloor-A5-IndianHills-Chicago-IL-USA".

493 4.2.4 Maximum Print Quality

494 This directory attribute indicates a somewhat subjective evaluation
495 of the overall printing quality :- "high", "medium", or "low". The
496 syntax and values shall be the same as for the print-quality Job
497 attribute.

498 4.2.5 Cost

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

4.2.6 Resolution

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

~~The syntax shall be the same as that of the printer resolution select job attribute. That syntax allows a single integer to specify the maximum resolution or a pair of integers to specify the maximum resolution when the x and y dimensions differ. When two and semantics shall be the same as for the printer-resolution-select job attribute.~~

~~integers are specified, the first is in the x direction, ie., the direction fo the shortest dimension of the medium, so that the value is independent of whether the Prin ter feeds long edge or short edge first.~~

~~4.2.7 Color Supported~~

~~4.2.7 This is a either a BOOLEAN for either yes, color printing is supported, or no color printing is not supported or it is an enumerated value such as "highlight", "four color", "mono", etc Color Supported~~

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

4.2.7 Fonts Supported

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

4.2.8 Maximum Speed

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

4.2.9 Device Id

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

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

541 4.2.10 Make and Model

542 This directory attribute is a simple text string defined by the
543 manufacturer that contains some reference to the make and model of
544 the entity being represented to the end-user by this Printer object.
545 ~~Is is recommended that the manufacturer use some regular form such as~~
546 ~~"vendor-~~The syntax shall be:

547 ____ vendor-name "/" model-name

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

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

551 4.2.11 Marker Type

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

557 4.2.12 Document Formats Supported

558 This directory attribute is a list of all of the document formats
559 that the printer and/or its interpreter(s) support. The syntax and
560 values ~~are the same as those for the document formats supported~~
561 ~~Printer attributes shall be the same as for the document-format Job~~
562 attribute.

563 4.2.13 Sides Supported

564 This directory attribute specifies the capabilities of the Printer
565 for marking on sides of the medium. The syntax and values shall be
566 the same as the sides ~~supported Printer attribute. Standard values~~
567 ~~are: 1 sided (simplex), 2 sided long edge (duplex), and 2 sided~~ Job
568 attribute.

569 ~~short edge (tumble).~~

570 4.2.14 Finishings Supported

571 This directory attribute identifies the finishing operations
572 supported by the Printer. The s ~~tandard finishing objects are defined~~
573 ~~in the section on~~ yntax and values shall be the same as the finishing
574 job attribute.

575 4.3 Directory Entries Using LDAP

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

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

584 To create a Printer object directory entry using LDAP:

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

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

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

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

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

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

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

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

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

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

608 Example: ldap_unbind (ld)

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

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

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

616 ~~ldap:///dir.host.name???(sides-supported=2-sided-long-binding-edge)~~
 617 ldap:///dir.host.name???(objectClass=printer
 618 (sides-supported=2-sided-long-edge))

619
 620 Issue: Is it allowed to filter the search based on the object class
 621 itself, in this case the object class of Printer? We need to define
 622 this new object class. How do we do this? One proposal is to
 623 subclass the device class defined in X.500:

624 printer OBJECT-CLASS ::= {
 625 SUBCLASS OF {device}
 626 MUST CONTAIN {<list of mandatory attributes>}
 627 MAY CONTAIN {<list of optional attributes>}
 628
 629

630 5. IPP Operations

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

634 635 5.1 HTTP Overview

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

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

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

652 containing server information, entity meta-information, and possibly
 653 some content.

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

659 5.2 IPP Operation Encoding

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

662 ~~HTTP~~IPP MESSAGE = Request | Response

663
 664 Requests and responses use the generic message format of RFC 822 for
 665 transferring entities. Both messages may include optional header
 666 fields and an entity body. The entity body is separated from the
 667 headers by a null line (a line with nothing preceding the CRLF).

668
 669 ~~Request = Request-line~~
 670 ~~* (General-Header~~
 671 ~~| Request-Header~~
 672 ~~| Entity-Header)~~
 673 ~~CRLF~~
 674 ~~[Entity-Body]~~

675
 676 ~~Response = Status-line~~
 677 ~~* (General-Header~~
 678 ~~| Request-Header~~
 679 ~~| Entity-Header)~~
 680

681 Request = Request-line
 682 * (General-Header
 683 | Request-Header
 684 | Entity-Header)
 685 CRLF
 686 [Entity-Body]

687
 688 Response = Status-line
 689 * (General-Header
 690 | Request-Header
 691 | Entity-Header)
 692 CRLF
 693 [Entity-Body]

694
 695 All IPP headers conform to the syntax

696 ~~IPP-Header = field~~ IPP-Header = field- name ":" [field-value]
 697 CRLF.

698
 699 IPP/1.0 defines the octet sequence CRLF as the end-of-line marker for
 700 all protocol elements except the entity-body.

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

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

710 5.2.1 HTTP Request-Header Fields

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

717
 718 HTTP-~~Entity-Header~~ = Content-Encoding
 719 | Content-Length
 720 | Content-Type
 721 | extension-header
 722

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

728
 729 Content-Type will always be "Application/IPP".

730 5.2.1.1 IPP Request-Line

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

734
 735 IPP-Request-Line = Operation-token IPP/1.0 CRLF
 736 ~~IPP Request Line = Operation token IPP/1.0 CRLF~~
 737
 738 ~~Operation token = Print | CancelJob | GetAttributes |~~
 739 ~~Get~~

```

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

```

5.2.2 HTTP Response-Header Fields

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

5.2.2.1 IPP Status-Line

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

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

5.3 The Print Job

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

```

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

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

```

5.3.1 Print Job Object Header

```

Print-Job-Object Header = Content-Encoding
                          | Content-Length
                          | Content-Type
                          | extension-header

```

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

785 5.3.2 Document Header

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

```
790 Document-Header = Content-Encoding  
791 | Content-Length  
792 | Content-Type  
793 | extension-header  
794
```

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

797 5.3.3 Document-Content Header

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

```
803  
804 Document-Content-Header = Content-Encoding  
805 | Content-Length  
806 | Content-Type  
807 | extension-header  
808
```

809 Content-Type is defined as :

810 Content-Type = Data ~~Stream~~ Stream Format "/" Version

811
812 Thus, for example, if the document to be printed was a Postscript
813 Level 2 document, the Content-Type would be specified as:

814 Content-Type: Postscript/2.0

815
816 Other header fields are as defined by HTTP 1.0.

817 5.3.4 Job Attributes

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

```
820  
821 Job Attribute = attribute name ":" Attribute value CRLF  
822  
823 Attribute value = Value | *(Value "," Value)
```

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

Attr-value = 1#Value

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

5.3.5 Document Attributes

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

~~Document Attribute = attribute name ":" Attribute value CRLF~~

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

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

~~Attr-Value = 1#Value~~

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

5.4 Operation Semantics

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

5.4.1 Print Operation

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

Note that the Printer name is not needed since it is the target of the entire operation. A Print Job contains the information needed by the Print er object to print a document or set of documents. When the print operation is invoked, the Entity-Body in the HTTP request includes an IPP Print Job. The concrete syntax of the Print Job is defined in section 6.2.5.3.

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

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

865
866 5.4.1.1 Print Request

867 The following abstract data types are part of the Print Request 7:

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 not shall not be included when a URL is provided <u>in the document- to point URL attribute which points</u> to the content.

868
869
870 5.4.1.2 Print Response

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

872

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

873
874
875 5.4.2 Cancel Job Operation

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

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

884 5.4.2.1 Cancel-Job Request

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

887

Message	Optional message to the operator.
---------	-----------------------------------

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

888
 889 5.4.2.2 Cancel-Job Response

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

892

Job Status	Optional Job status information
Errors	Optional Error Information

893

894 5.4.3 Get Attributes Operation

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

905 5.4.3.1 Get-Attributes Request

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

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

~~Requested~~ ~~A set of attributes in which the requester is~~
~~Attributes~~ ~~interested~~

908

909 5.4.3.2 Get-Attributes Response

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

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

~~Result~~ ~~The requested attributes of the object~~
~~Attributes~~

Errors Optional error information

912

913 5.4.4 Get Jobs Operation

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

920 5.4.4.1 Get~~Jobs~~ Jobs Request

921 The following abstract data types are part of the Get Jobs Request:
922

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

all jobs (including completed jobs), "pending" -
all jobs which are pending and
processing, or just " myprocessing

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

~~selector~~ ~~all jobs (including completed jobs), all jobs~~
 ~~which are pending and processing, or just "my~~
 ~~jobs" that are pending or processing.~~

Requested A set of attributes without values in whose

Attributes values the requester is interested.

5.4.4.2 Get--Jobs Response

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

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

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

Jobs A list of Job URLs is r are returned:eturned . The list is in "scheduled" order. For each J position-in-list. ob URL the following attributes are returned: job-total-octets and andposition-in-list. This last attribute is necessary since and end user may request just their own jobs and they need some relative position indicator if there are other jobs inter esponse.spersed in the waiting list which are not returned in the response.

Errors Optional Error Information

6. Object Attributes

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

936 6.1 Attribute Syntaxes

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

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

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

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

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

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

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

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

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

968 The sections below reference the following syntax items:

string	arbitrary ASCII strings, no control characters, except <SPACE>.
<u>s</u> StringPair	string
	<u>":":</u> string
stringState	string
	<u>s</u> State
name	arbitrary ASCII strings, no control characters, and no <SPACE> characters.

U	Universal Resource Locator
R rl	
dateTime	date and time in RFC 822 format
deltaTime	[hours
	<u>":"</u>] 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.
<u>type2Enum</u>	<u>standard names, but an implementor can, at any time, add new</u>
	<u>TBD by values by proposing them to</u>
	<u>the PWG for registration (or an</u>
	<u>IANA-appointed registry advisor</u>
	<u>after the PWG is no longer</u>
	<u>certified) where they are</u>
	<u>anytime reviewed for approval..</u>
	<u>IANA keeps the registry.</u>
	<u>Implementors can</u>
	<u>add support private (un-registered)</u>
	<u>with a suitable distinguishing</u>
	<u>prefix, such as - xxx- where xxx is</u>
	<u>the company name registered with</u>
	<u>IANA. IANA for use in domain names.</u>
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 add
	private (un registered) with a suitable
	distinguishing prefix, such as xxx
	where xxx is the company name
	registered with IANA.
<u>Type3Enum</u>	<u>standard names, but an implementor</u>
	<u>can add new</u>
	<u>names by submitting a</u>
	<u>registration request directly to</u>
	<u>IANA, no PWG or IANA-appointed</u>
	<u>registry advisor review is</u>
	<u>required. Implementors can</u>
	<u>add private (un-support private</u>
	<u>(un-registered) names with a</u>
	<u>suitable distinguishing prefix,</u>
	<u>such as -xxx- where xxx is the</u>
	<u>company name registered with IANA</u>
	<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 add private (un

	registered) names with a suitable distinguishing prefix, such as xxx where xxx is the company name registered with IANA.
<u>type2EnumState</u>	<u>type2Enum State</u>
<u>type3EnumState</u>	<u>type3Enum State</u>
<u>boolean</u>	<u>tokens: yes, y, true, or t and no, n, false, or f.</u>
type2EnumState	type2Enum state
type3EnumState	type3Enum state
state	type1Enum
Boolean	tokens: yes, y, true, or t and no, n, false, or f.
<u>positiveInteger</u>	<u>1 .. n represented as ASCII digits</u>
	TBD
<u>positiveIntegerCross</u>	<u>positiveInteger [</u>
	<u>"#""x" positiveInteger]</u>
<u>positiveIntegerCrossState</u>	<u>positiveIntegerCross State</u>
positiveIntegerCross	positiveIntegerCross state
State	
<u>positiveIntegerRange</u>	<u>positiveInteger</u>
	<u>"#"":" positiveInteger</u>
<u>positiveIntegerUnits</u>	<u>positiveInteger units</u>
<u>positiveIntegerState</u>	<u>positiveInteger State</u>
<u>units</u>	<u>"ppm" "ipm" "spm" "cps" </u>
	<u>"lpm"</u>
<u>type3Locale</u>	<u>type3Country ":" type3Language ":"</u>
	<u>type3CodeSet</u>
positiveIntegerState	positiveInteger state
units	"ppm" "ipm" "spm" "cps" "lpm"
type3Locale	type3Country ":" type3Language ":"
	type3CodeSet
<u>type3Country</u>	<u>type3Enum - Standard values are the</u>
	<u>two-character country codes from</u>
	<u>ISO 639.</u>
<u>type3Language</u>	<u>type3Enum - Standard values are the</u>
	<u>two-character language codes from</u>
	<u>ISO 3166.</u>
<u>type3CodeSet</u>	<u>type3Enum - Standard values are</u>
	<u>from the IANA Code Set registry.</u>
type3Country	type3Enum
type3Language	type3Enum
type3CodeSet	type3Enum
<u>type2Format</u>	<u>name [</u>
	<u>"#""/" version]</u>
<u>version</u>	<u>name</u>
<u>type3LocaleState</u>	<u>type3Locale</u>
	<u>State</u>

969

970

Also, the following conventions (from RFC 822) are used:"1#" in front of a data means one or more values separated

syntax by ",",.

"#" in front of a data syntax means zero or more

"1#" in front of a data syntax means one or more

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

6.2 Job Attributes

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

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

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

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

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

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

6.2.1.1 job-name (string)

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

1010 This attribute is intended ~~for~~ to be printed on a start sheet,
1011 returned in a Get-Jobs result, or used in notification messages.

1012 If the client does not specify this attribute, a Printer shall set it
1013 to the ~~nam~~value of the ~~fil~~document-name attribute of the first
1014 document in the job.

1015 6.2.2 Job Informational Attributes (Set by a Printer)

1016 The Print shall add all of these attributes to a job to provide
1017 information to identify a print-job.

1018 The client may specify these attributes in the operations: Get-
1019 Attributes and Get-Jobs, but not in Print.

1020 6.2.2.1 job-identifier (url)

1021 This attribute provides the job-identifier for this job on the
1022 Printer. The Printer shall generate a job-identifier value as a
1023 URL.~~7~~

1024 The value of the job-identifier attribute shall be returned by the
1025 Printer as part of the PrintResult in the Print operation.

1026 6.2.2.2 job-originator (name)

1027 This attribute specifies the name of the person submitting the print
1028 job. The Printer shall set this attribute to the most authentic name
1029 that it can obtain from the client. The operation-user-name attribute
1030 is intended to be a source of the most authentic name.

1031 6.2.2.3 job-originating-host (name)

1032 This attribute identifies the originating host of the job. The
1033 Printer shall set this attribute to the value of the operation-host-
1034 name which is intended to be the most authentic host name of the
1035 client.

1036

1037 ~~6.2.2.4 notification address (name)~~

1038 ~~This address specifies the email address of the client. The client~~
1039 ~~specifies this attribute in the operation notification address~~
1040 ~~attribute which the Printer in turn uses to set this attribute.~~

1041 ~~The Printer shall use this attribute as the address for sending~~
1042 ~~messages to a job submitter when an event occurs that the end user has~~
1043 ~~registered an interest in or when certain other events occur, such as~~
1044 ~~Cancel Job.~~

1045 ~~Note: The only type of notification is email.~~

1046 ~~Issue: Can the email address be inferred with job originator and the~~
 1047 ~~originating host?~~

1048 6.2.2.4 job-locale (type3Locale)

1049 This attribute identifies the locale of the job , i.e., the country,
 1050 language, and code d character set. The Printer sets this attribute
 1051 from the value of the operation-locale.

1052 The Printer shall use this attribute to determine the locale for
 1053 notification messages that it sends.

1054 ~~The type3EnumTrip consists of 3 colon separated type 3 enums. The~~
 1055 ~~first shall be the two character country code from ISO 639. The~~
 1056 ~~second shall be the two character language code from ISO 3166. The~~
 1057 ~~third is the code set from the IANA Code Set Registry.~~

1058 Issue: Is there a more standard syntax for locale?

1059 6.2.3 Job Status Attributes (Set by Printer)

1060 The Printer shall add these attributes to a job when a client submits
 1061 a job, and the Printer shall assign appropriate values to each such
 1062 job-status attribute.

1063 The Printer uses these attributes to specify the job status before,
 1064 during and after the processing of the print-job by the Printer.

1065 The client may specify job-status attributes in: Get-Attributes and
 1066 Get-Jobs, but not Print.

1067 6.2.3.1 current-job-state (type1Enum)

1068 This attribute identifies the current state of the job ~~with the~~
 1069 ~~following values: unknown, pre processing, pending, processing,~~
 1070 ~~printing, held, terminating, retained, completed.~~ Standard values
 1071 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	<p>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.</p>
<u>processing</u>	<p><u>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.</u></p> <p><u>Or</u></p> <p><u>The server has completed processing the job and the output device is currently printing the job on at least one printer. That is, a print engine in the job. That is, an output device is either printing pages of the job, or failing in its attempt to print pages of the job because of some wait state, such as, start-wait, end-end-wait, needs-attention, etc. The complete job state includes the detailed status represented in the printer s' 's printer-state attribute.</u></p>
processing	<p>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.</p> <p>Or</p> <p>The server has completed processing the job and the output device is currently printing the job on at least one printer. That is, a print engine 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' printer state attribute(s)</p>
<u>paused</u> paused	<p><u>The job has been paused</u></p> <p>The job has been paused as a result of a PauseJob operation.</p>

<u>Interrupted</u>	<u>The job has been interrupted by some intervening job, and shall resume processing automatically once the intervening job has completed.</u>
interrupted	The job was interrupted by the InterruptJob request for an intervening job, and shall resume processing automatically once the intervening job has completed.
<u>Terminating</u>	<u>The job has been canceled by a Cancel-Job request or aborted by the server and is in the process of terminating. The job's job-state-reasons attribute contains the reasons that the job is being terminated.</u>
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.
<u>rRetained</u>	<u>The job is being retained at the server as a result of the job's job-retention-period being non-zero. The job has (1) completed successfully or with warnings or errors, (2) been aborted while printing by the server, or (3) been canceled by the Canceled by the Cancel-Job request before or during processing. The job's job-state-reasons attribute contains the reasons that the job has been retained. While in the retained state, all of the job's document data (and resources, if any) shall be retained by the server; thus a job in the retained state could be reprinted, using some means outside the scope of IPP V1.0.</u>
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 ISO/IEC 10175 Part 1.

Completed

The job has:

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

AND the job's:

- (1) job-retention-period was zero or has
expired, or
- (2) job-discard-time has arrived.

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

completed The
~~The job has:~~
~~(1) completed successfully or with~~
~~warnings or errors,~~
~~(2) been aborted by the server while~~
~~printing, or~~
~~cancelled by the CancelJob~~ (3) been cancelled
~~by the CancelJob~~
~~request, 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 s tate: job~~
~~identifier, job owner, job name, current job~~
~~state, printers assigned, and job state~~
~~reasons.~~

1072

1073 The IPP protocol supports all values for job states, but Printers ~~are~~
 1074 need only support those states which are appropriate for the
 1075 particular implementation.

1076 6.2.3.2 output-device-assigned (name)

1077 This attribute identifies the Output Device to which the Printer has
 1078 assigned this job.

1079 If an Output Device implements a Printer, the Printer does need not
 1080 set this attribute.

1081 If a Print Server implements a Printer, the value shall be empty
 1082 until the Printer assigns an Output Device to the job. -

1083 The value of the job's printer-output-device-assigned attribute shall
 1084 remain after the job has completed, so that end users can determine
 1085 the Output Device on which the job was printed.

1086 6.2.3.3 submission-time (dateTime)

1087 This attribute indicates the time at which this job was accepted by
1088 the Printer. If the Printer does not support the notion of time, the
1089 attribute ~~is not~~need not be stored as part of the job object.

1090 6.2.3.4 number-of-intervening-jobs (cardinal)

1091 This attribute indicates the number of jobs that are "ahead" of this
1092 job in the current scheduled order. For efficiency, it is only
1093 necessary to calculate this value when an operation is performed that
1094 requests this attribute.

1095 NOTE - This attribute is necessary since an end user may request just
1096 their own jobs and they need some relative position indicator if
1097 there are other jobs interspersed in the waiting list which are not
1098 returned in the response or cannot be because of site security policy
1099 restrictions.

1100 6.2.3.5 job-message-from- ~~administ~~opeerator (string)

1101 This attribute provides a message from an operator, system
1102 administrator or "intelligent" process to indicate to the end user
1103 the reasons for modification or other management action taken on a
1104 job.

1105 6.2.3.6 completion-time (dateTime)

1106 This attribute indicates the time at which this job completed. This
1107 time is useful for jobs which are retained after printing. If the
1108 Printer does not support the notion of time, the attribute is not
1109 stored as part of the Job object.

1110 6.2.3.7 job-state-reasons (1#type2Enum)

1111 This attribute identifies the reason or reasons that the job is in
1112 the state that it is in (e.g., held, terminating, retained,
1113 completed, etc.). The printer shall indicate the particular
1114 reason(s) by setting the value of the job-state-reasons attribute.
1115 ~~It is~~

1116 ~~valid for the printer to set the value of the job state reasons~~
1117 ~~attribute to the empty set.~~

1118 The following standard values are defined:

none

There are not reasons associated with
the job's current state.

documents-needed	The complete job has been accepted by the server (the value of the job submission complete element was TRUE in the last print request for the job) server, but the server is waiting for its files to be transferred before the job can be scheduled to be printed.
job-hold-set	The value of the job's job-hold attribute is TRUE.
job-print-after-specified	The value of the job's job-print-after or print-off-peak attributes have attribute has 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.

1119

1120

1121 6.2.3.8 impressions-completed (cardinal)

1122 This attribute contains the number of impressions that the Printer
 1123 has completed printing. If the Printer cannot report this number,
 1124 the Printer leaves this attribute unspecified.

1125 6.2.3.9 media-sheets-completed (cardinal)

1126 This attribute contains the number of media-sheets that the Printer
 1127 has completed printing. If the Printer cannot report this number,
 1128 the Printer leaves this attribute unspecified.

1129 6.2.4 Job Sheet Attributes (Set by Client/End User)

1130 The client shall specify these attributes to control the printing of
 1131 job sheets.

1132 The client may also specify job sheet attributes in: Get-Attributes
1133 and Get-Jobs.

1134 6.2.4.1 job-sheets (type3Enum)

1135 This attribute determines what type of job-sheets the Printer shall
1136 print with the job.

1137 The standard values are: none, and default-sheet.

1138 The value "none" means that the Printer shall print no job sheets.
1139 The value "default-sheet" means that the Printer shall print the job
1140 sheets defined by an administrator. If the administrator's policy is
1141 not to support none, the Printer shall use the default-sheet value if
1142 the client supplies the "none" value.

1143 NOTE - The effect of this attribute on jobs and documents is
1144 controlled by the files-are-one-document and files-are-interleaved
1145 job attributes.

1146 6.2.5 Notification Attributes (Set by a Client/End User)

1147 The client shall specify these attributes to indicate events that the
1148 client is interested in , along with the notification address and
1149 method for performing the notification .

1150 The client may also specify notification attributes in: Get-
1151 Attributes and Get-Jobs.

1152 6.2.5.1 notification-events (~~#type1~~#type2Enum)

1153 This attribute specifies the events about which the end user want to
1154 be notified.

1155 ~~This attribute will support four events classes~~ Standard values are :
1156 none, job-completion, job-problems and printer-problems.

1157 If this attribute contains the event none, the Printer shall not
1158 notify. This value is useful if an administrator has set up a
1159 notification Printer default but the end user does not w hichant
1160 notification. If the none value and other values are supplied, the
1161 Printer shall ignore the none value.

1162 ~~This attribute will support only one delivery method, namely email.~~
1163 ~~The attribute notification address specifies the email address.~~

1164 If this attribute contains the event value: job-completion, the
1165 Printer shall notify the client when the job containing this
1166 attribute completes with or without errors or is cancelled by the
1167 end-user or the operator .

1168 If this attribute contains the ~~event-job-problem-value: job-problems~~,
1169 the Printer shall notify the client when ~~the job containing~~ this
1170 ~~attribute-job~~ has a problem while th ~~eis~~ job is printing. Problems
1171 include: paper jam and out-of-paper.

1172 If this attribute contains the ~~event-printer-problem-value: printer-~~
1173 ~~problems~~, the Printer shall notify the client when ~~the job containing~~
1174 ~~this attribute has a problem while the job is printing or waiting to~~
1175 ~~print~~any job, including this job, has a problem while this job is
1176 ~~waiting to print or printing~~. Problems include: paper jam and out-of-
1177 paper.

1178 6.2.5.2 notification-address (url)

1179 This address specifies both the address and mechanism for delivery of
1180 notification events to the client. The client specifies this
1181 attribute in the operation-notification-address attribute which the
1182 Printer in turn uses to set this attribute.

1183 The Printer shall use this attribute as the address for sending
1184 messages to a job submitter when an event occurs that the end user
1185 has registered an interest in or when certain other events occur,
1186 such as Cancel-Job.

1187 If the URL has a "mailto:" scheme, then email is used and the rest of
1188 the URL is used as the email address. If the URL has a "http:"
1189 scheme, then an HTTP method is used to add HTML formatted events to
1190 the end of the specified HTML file.

1191 6.2.6 Job Scheduling Attributes (Set by Client/End User)

1192 The client shall specify these attributes to provide the Printer with
1193 information for the scheduling a print-job.

1194 The client may also specify these attributes in: Get-Attributes and
1195 Get-Jobs.

1196 6.2.6.1 job-priority (typeName)

1197 This attribute specifies a priority for scheduling the print-job.
1198 Printers that employ a priority-based scheduling algorithm use this
1199 attribute.

1200 There are three standard values: high, default, and low. Among those
1201 jobs that are ready to print, a Printer shall print all such jobs
1202 with a high priority before printing those with a default or low
1203 priority, and a Printer shall print all such jobs with a default
1204 priority before printing those with a low priority.

1205 If the client does not specify this attribute, the Printer assumes
1206 that the end user places no constraints concerning priority on the
1207 scheduling of the print-job, and it has a priority value of default .

1208 An operator can modify a job to have any priority. An end-user is
1209 restricted by the value of the maxi mum-end-user-priority Printer
1210 attribute.

1211 6.2.6.2 job-print-after (dateTime)

1212 This attribute specifies the calendar date and time of day after
1213 which the print-job shall become a candidate for printing.

1214 If the value of this attribute is in the future, the Printer shall
1215 set the value of the job's current-job-state to held and add the job-
1216 print-after-specified value to the job's job-state-reasons attribute
1217 and shall not schedule the print-job for printing until the specified
1218 date and time has passed. When the specified date and time arrives,
1219 the Printer shall remove the job-print-after-specified value from the
1220 job's job-state-reason attribute and, if no other reasons remain,
1221 shall change the job's current-job-state to pending so that the job
1222 becomes a candidate for being scheduled to print.

1223 If this attribute is unspecified or the value is in the past, the job
1224 shall be a candidate for scheduling immediately.

1225 6.2.6.3 job-print-off-peak (type3Enum)

1226 This attribute specifies the off-peak period during which the print-
1227 job shall become a candidate for printing.

1228 Standard values are: "evening", "night", "weekend", "second-shift",
1229 "third-shift".

1230 If this attribute is specified, it contains a value with which an
1231 administrator has associated allowable print times. An administrator
1232 is encouraged to pick names that suggest the type of off-peak period.

1233 If the value of this attribute is in the future, the Printer shall
1234 set the value of the job's current-job-state to held and add the job-
1235 print-after-specified value to the job's job-state-reasons attribute
1236 and shall not schedule the print-job for printing until the specified
1237 date and time has passed. When the specified date and time arrives,
1238 the Printer shall remove the job-print-after-specified value from the
1239 job's job-state-reason attribute and, if no other reasons remain,
1240 shall change the job's current-job-state to pending so that the job
1241 becomes a candidate for being scheduled to print.

1242 If this attribute is unspecified, the job shall be a candidate for
1243 scheduling immediately.

1244 6.2.6.4 job-retention-period (deltaTime)

1245 The retention time is expressed in hours and minutes, e.g. 6:00 (6
1246 hours), or 20 (20 minutes).

1247 This attribute specifies the minimum period of time following the
1248 completion of job processing and printing that the server shall keep
1249 job attributes and document data. The Printer may keep these
1250 attributes and data longer than the value of the job-retention-period
1251 attribute.

1252 ~~Issue: There is some discussion about whether or not this should be~~
1253 ~~removed from the spec? NOTE - the requester may change this job~~
1254 ~~attribute using the input parameter to the Cancel-Job operation.~~

1255 6.2.7 Job Production Attributes (Set by Client/End User)

1256 The client shall specify these attributes to affect the rendering,
1257 production and finishing of the documents in the job. Similar types
1258 of instructions may also be contained in the document to be printed.

1259 If there is a conflict between the value of one of these attributes,
1260 and a corresponding instruction in the document (either implicit or
1261 explicit), the value of the attribute shall take precedence over the
1262 document instruction.

1263 Job Production and Resource Attributes each address a similar set of
1264 features but they have different uses.

1265 A job production attribute provides a client with a way to request
1266 some feature ~~that is not embedded within the document data. After~~
1267 ~~some program has merged the production attributes into the document~~
1268 ~~data After the information from these attributes has been at print~~
1269 ~~time that may not have been embedded within the document data when~~
1270 ~~the document was created. A job production attribute also provides a~~
1271 ~~client with a way to override a folded into the document data~~
1272 ~~(possibly during a translation process of the document data), these~~
1273 ~~attributes are no longer relevant and sh all can be discarded from a~~
1274 ~~job. Instead, the resource attributes specify the resources needed~~
1275 ~~to print the job as modified by the job production attributes eature~~
1276 ~~at print time that was embedded within the document data when the~~
1277 ~~document was created.~~

1278 Note: until companies that supply interpreters for PDL's, such as
1279 PostScript and PCL allow a way to specify overrides for internal job
1280 production instructions, a Printer may not be able to implement these
1281 attributes for some PDL's.

1282 A job resource attribute tells a Printer what features the job needs.
1283 A program that translates document data to a Printer's PDL, and/or
1284 merges production attributes into the document data should add job
1285 resource attributes to a job.

1286 For example, a job production attribute medium-select with the value
1287 of "letter" requests that a job be printed on letter paper, but gives
1288 no information about what resources the job needs. ~~A resource~~
1289 ~~production~~ For example, a job resource attribute media-used with the

1290 values of "letter" and "ledger" tell a Printer that the job needs
 1291 letter and ledger paper, but gives no information about which pages
 1292 use each medium.

1293

1294

1295 The client may also specify document-job production-instruction
 1296 attributes in: Get-Attributes and GetJobs.

1297 6.2.7.1 medium-select (type2Enum)

1298 This attribute identifies the medium that the Printer shall use for
 1299 all pages of the document regardless of what media are specified
 1300 within the document.

1301 The values for medium include medium-names, medium-sizes, input-trays
 1302 and electronic forms so that one attribute specifies the media.

1303 Standard values are ~~defined~~ (taken from ISO DPA and the Printer MIB)

1304 :

1305

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

1306

1307 The following standard values are defined for North American media:

na-letter	Specifies the North American letter
--white	white medium
na-letter	Specifies the North American letter
na-letter <u>na-letter</u> colored	coloured medium
na-letter	Specifies the North American letter
--transparent	transparent medium
na-legal	Specifies the North American legal
--white	white medium
na-legal	Specifies the North American legal
na-legal <u>na-legal</u> colored	coloured medium

1308

1309 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

1310

1311 The following standard values are defined for the less commonly used
1312 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	Specifies 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

1313

1314 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

1315

1316 ~~6.2.7.2 number up (positiveInteger)~~

1317 ~~This attribute specifies the number of source page images to impose~~
1318 ~~upon a single side of an instance of a selected medium.~~

1319 ~~In general, only certain numeric values are valid for this attribute,~~
1320 ~~depending upon the Printer implementation to which the print request~~
1321 ~~is directed. Typical supported values are 2 and 4. If this attribute~~

1322 ~~is unspecified or has a value of 1, then the Printer does not apply~~
 1323 ~~any number up transformation to the pages.~~

1324 ~~This attribute primarily controls the translation, scaling and~~
 1325 ~~rotation of page images, but a site may choose to add embellishments,~~
 1326 ~~such as borders to each logical page. If embellishments are added,~~
 1327 ~~especially for the number up = 1 case, these are controlled through~~
 1328 ~~some other mechanism or attribute. The user expects that if number up~~
 1329 ~~is absent or equal to 1 then no other imposition embellishments are~~
 1330 ~~added via this attribute.~~

1331

1332 finishing (type2Enum) The following standard values are defined for
 1333 input-trays (from 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>

1334

1335 The following standard values are defined for media sizes (from ISO
 1336 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>

1337

<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>

1338

<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>

1339

<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>

1340

<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>

1341

<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>

1342

<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>

1343

1344 6.2.7.2 finishing (type2Enum)

1345 This attribute identifies the finishing operation that the Printer
 1346 should apply to each copy of the printed document. ~~Examples include~~
 1347 ~~stapling, saddle stitching, hole drilling, binding with tape, etc.~~

1348 NOTE - The effect of this attribute on jobs and documents is
 1349 controlled by the files-are-one-document and files-are-interleaved
 1350 job attributes.

1351 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.

1352

1353 6.2.7.3 number-up (type3Enum)

1354 This attribute specifies the number of source page-images to impose
1355 upon a single side of an instance of a selected medium.

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

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

none Perform no finishing. See 9.1.2

6.2.7.4 sides (type2Enum)

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

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

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

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

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

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

6.2.7.5 copies (positiveInteger)

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

1393 NOTE - The effect of this attribute on jobs and documents is
 1394 controlled by the files-are-one-document and files-
 1395 are-interleaved job attributes.

1396 6.2.7.6 printer-resolution-select (positiveIntegerCross)

1397 This attribute specifies the resolution that the Printer should use.

1398 The syntax allows a single integer to specify the resolution or a
 1399 pair of integers to specify the resolution when the x and y
 1400 dimensions differ. When two integers are specified, the first is in
 1401 the x direction, ie., in the direction ~~feof~~ the shortest dimension of
 1402 the medium, so that the value is independent of whether the printer
 1403 feeds long edge or short edge first. -

1404 6.2.7.7 print-quality (type2Enum)

1405 This attribute specifies the print quality that the Printer should
 1406 use.

1407 The standard values are:

1408	draft	Lowest quality available on the printer
1409	normal	Normal or intermediate quality on the printer
1410	high	Highest quality available on the printer

1411

1412

1413 6.2.7.8 page-select (positiveIntegerRange)

1414 This attribute specifies the pages in the document that the Printer
 1415 shall use. This attribute is unlikely to be useful for jobs with more
 1416 than one document or in Job Templates. If this attribute is
 1417 unspecified, then the Printer ~~prints~~shall print all pages in a
 1418 document.

1419 6.2.7.9 files-are-one-document (~~B~~boolean)

1420 This attribute is relevant only if a job consists of two or more
 1421 documents. It controls finishing operations, job-sheet placement, and
 1422 the order of documents when the copies attribute exceeds 1.

1423 If the files for the job are a and b and this attribute is true, then
 1424 files a and b are treated as a single document for finishing
 1425 operations. Also, there will be no slip sheets between files a and b.
 1426 If more than one copy is made, the ordering must be a, b, a, b,
 1427 The attribute files-are-interleaved is ignored.

1428 If the files for the job are a and b and this attribute is false or
 1429 unspecified by both the client and the Printer's Job Template, then
 1430 each file is treated as a single document for finishing operations.
 1431 Also, a client may specify that a slip sheet be between files a and
 1432 b. If more than one copy is made, and the attribute files-are-

1433 interleaved false or unspecified, the ordering is a, a, b, b, If
1434 more than one copy is made, and the attribute files-are-interleaved
1435 true, the ordering is a, b, a, b,

1436 | 6.2.7.10 files-are-interleaved (Bboolean)

1437 | This attribute is used in conjunction with files-are-one-document
1438 | (q.v.).

1439

1440 | 6.2.8 Attributes for Conversion of Text and HTML Files (Set by
1441 | Client/End User)

1442 | The client shall specify these attributes to control formatting for
1443 | text documents or HTML documents. ~~If the client does not specify any~~
1444 | ~~of these attributes, a Printer shall uses its own defaults.~~

1445 | A client need not specify these attributes for other types of
1446 | documents, such as PostScript or PCL.

1447 6.2.8.1 width (cardinalUnits)

1448 | This attribute specifies the media width for the document in
1449 | characters.

1450 6.2.8.2 length (cardinalUnits)

1451 | This attribute specifies the media length for the document in
1452 | characters.

1453 6.2.8.3 left-margin (cardinalUnits)

1454 | This attribute specifies the left-margin for the document in
1455 | characters.

1456 6.2.8.4 right-margin (cardinalUnits)

1457 | This attribute specifies the right-margin for the document in
1458 | characters.

1459 6.2.8.5 top-margin (cardinalUnits)

1460 | This attribute specifies the top-margin for the document in lines.

1461 | 6.2.8.6 bottom-margin (cardinalUnits)

1462 | This attribute specifies the bottom-margin for the document in lines.

1463 6.2.8.7 repeated-tab-stops (cardinalUnits)

1464 This attribute specifies the tab stops for the document in
1465 characters.

1466 6.2.8.8 header-text (string)

1467 This attribute specifies the header text for the document.

1468 6.2.8.9 footer-text (string)

1469 This attribute specifies the footer text for the document.

1470 6.2.8.10 number-pages (boolean)

1471 This attribute specifies that the pages should be numbered in the
1472 document.

1473 6.2.8.11 default-font (string)

1474 This attribute specifies the font to use for all text in the
1475 document.

1476 6.2.8.12 font-size (cardinalUnits)

1477 This attribute specifies the font-size in points for text in the
1478 document. The value of this attribute affects the size of the other
1479 text attributes.

1480 If this attribute is omitted and the Printer's default Job Template
1481 does not contain this attribute, the Printer shall assume a value of
1482 10. A value of 10 with a fixed pitch font, shall produce 12
1483 characters per inch in the horizontal direction and with 6 lines per
1484 inch in the vertical direction.

1485 ~~6.2.8.13 number-pages (Boolean)~~

1486 ~~This attribute specifies that the pages should be numbered in the~~
1487 ~~document.~~

1488 ~~default font (string) This attribute specifies the font to use for~~
1489 ~~all text in the document.~~

1490 6.2.8.13 default-code-set (type3Enum)

1491 This attribute specifies the code-set in which the document is
1492 encoded.

1493 6.2.8.14 content-orientation (type2Enum)

1494 This attribute specifies the orientation of the document.

1495 | The standard values are :

portrait	The page orientation such that the sides are longer than the top when the page is held in the intended human reading orientation
landscape	The page orientation such that the sides are shorter than the top when the page is held in the intended human readable orientation. Landscape is defined to be a rotation of the page by +90 degrees with respect to the medium (i.e. anti-clockwise) from the portrait orientation NOTE - The +90 direction was chosen because simple finishing on the long edge is the same edge whether portrait or landscape
reverse- portrait	The page orientation defined to be a rotation of 180 degrees with respect to portrait
reverse- landscape	The page orientation defined to be a rotation of 180 degrees with respect to landscape. Landscape is defined to be a rotation of the page by -90 degrees with respect to the medium (i.e. clockwise) from the portrait orientation NOTE - Reverse-landscape was added because some applications rotate landscape -90 degrees from portrait, rather than +90 degrees.

1496

1497 6.2.9 Job Resource Attributes (Set by the program that produces or
1498 senses the PDL)

1499 | A program ~~described below~~ (described below) shall add these
1500 | attributes, which describe the resources needed to print the job.

1501 | A Printer may use these attributes to validate and schedule the
1502 | print-job without interpreting the contents of the document. This
1503 | provides the opportunity for a Printer to support a broad set of
1504 | document formats yet still support fast efficient scheduling and
1505 | validation of each job.

1506 |
1507 | The client/end user shall not specify these attributes. Instead, it
1508 | is the duty of the program that translates the document to the
1509 | printer's PDL (or analyzes it) to add these attributes ~~s~~ and their
1510 | values to the job. Such a program may execute at a number of
1511 | different points in time:

1512 1. The program produces a final form document and stores it+these
1513 resource attributes in a file before the end-user submits the
1514 print job.

1515 2. The program produces a final form document data stream when the
1516 end-user specifies "Print" to the application program (e.g.,
1517 Windows GDI driver).

1518 3. The program running in the context of the Printer or server
1519 translates a revisable or final form document into a PDL that the
1520 printer output device understands.

1521 If any of these attributes is unspecified, the Printer shall assume
1522 that the all resources required by the document of the type specified
1523 by the missing attributes are ready, ie., are available to the
1524 Printer and/or output device without human intervention.

1525 These attributes may be unspecified if the translation program fails
1526 to provides such values, or if no translation occurs (e.g. the
1527 document is a PostScript document).

1528 Note: The Printer does not use these attributes during the actual
1529 printing of a document.

1530 Note: these attributes allow more than one value wherever it is
1531 possible for a job to specify more than one value of the
1532 corresponding job attribute, possibly by embedded instructions.

1533 The client may specify these attributes in: Get-Attributes and Get-
1534 Jobs.

1535 See the section on job production attributes for an explanation of
1536 how the job resource attributes differ from the job production
1537 attributes.

1538 6.2.9.1 document-format s-used (1#type2Format)

1539 This attribute identifies the document formats needed to print the
1540 document(s) in this job.

1541 A format consists of two elements, a name and a version. The latter
1542 element is optional.

1543 The syntax is for type2Format:

1544 name ["/" version]

1545 Examples include: PostScript, PostScript/2.0 and PCL/5e

1546 Note: The version component is optional.

1547 | The names shall be registered with IANA as "printer languages"
1548 | following the procedures established by the Printer MIB (currently
1549 | proposed as an IETF standard by RFC 1759).

1550 6.2.9.2 fonts-used (1#string)

1551 This attribute identifies the font resources used in the document(s)
1552 in the job.

1553 6.2.9.3 code-sets-used (1#type3Enum)

1554 This attribute identifies the code-sets used in the document(s) in
1555 the Job. This attribute is relevant only for files that are not in
1556 ASCII, such as text files and possibly PCL files. PostScript files
1557 are always ASCII. Normally there is at most 1 code-set.

1558 Standard values are defined in the section specifying the default-
1559 code-set attribute.

1560 6.2.9.4 media-used (1#type2Enum)

1561 This attribute identifies the media, media-sizes, input-trays or
1562 electronic forms needed to print the document(s) in the job.

1563 Standard values for this attribute are defined in the section
1564 specifying the medium-select attribute.

1565 |
1566 6.2.9.5 sides-used (type2Enum)

1567 This attribute specifies whether a job needs 1-sided, 2-sided-long-
1568 | binding edge, or 2-sided short binding edge, or 2-sided-short -edge
1569 printing.

1570 Standard values for this attribute are defined in the section
1571 specifying the sides Job attribute.

1572 6.2.9.6 print-quality-used (type2Enum)

1573 This attribute specifies what print quality the job needs.

1574 Standard values for this attribute are defined in the section
1575 specifying the print-quality attribute.

1576 6.2.9.7 finishing-used (type2Enum)

1577 This attribute specifies what finishing the job needs.

1578 Standard values for this attribute are defined in the section
1579 specifying the finishing attribute.

1580 6.2.9.8 printer-resolution-used (positiveIntegerCrossState)

1581 This attribute specifies what resolution the job needs.

1582 The interpretation of the values for this attribute are defined in
1583 the section on printer-resolution-select Job attribute.

1584 6.2.9.9 total-job-octets (positiveInteger)

1585 This attribute specifies the total size of the job in octets. This
1586 attribute is the first of three that a translation program can use to
1587 specify the size of a job.

1588 6.2.9.10 job-impression-count (positiveInteger)

1589 This attribute specifies the total size of the job in impressions.

1590 6.2.9.11 job-media-sheet-count (positiveInteger)

1591 This attribute specifies the total size of the job in media-sheets.

1592 ~~6.2.9.12 job-intervening-jobs (positiveInteger)~~

1593 ~~This attribute indicates the number of jobs that are "ahead" of this~~
1594 ~~job in the current scheduled order. For efficiency, it is only~~
1595 ~~necessary to calculate this value when an operation is performed that~~
1596 ~~requests this attribute.~~

1597 6.2.10 Number of Documents (Set by Client Printer)

1598 This group contains a single attribute which specifies the number of
1599 documents in the job.

1600 The Printer sets the value of this attribute depending on the number
1601 of documents that the client supplies in the Print operation. The
1602 client shall not specify this attribute ~~in Print and (directly) in~~
1603 Print, but may specify this attribute in: Get-Attributes and Get-
1604 Jobs.

1605 6.2.10.1 number-of-documents (positiveInteger)

1606 This attribute specifies the number of documents in the job. Each
1607 document shall contain its own set of document content attributes
1608 described below.

1609 6.2.11 Document Data (Set by a Client/End User)

1610 This group of attributes describes the document data for the job.
1611 These attributes also include the document data or reference it.

1612 All job attributes in other sections of this document occur only once
1613 per job and apply to all documents in a job.

1614 The client may specify document-data attributes in Print. The client
1615 must specify either the document-URL or document-content in Print.

1616 Except for document-content, the client may specify document-data
1617 attributes in: Get-Attributes, and Get-Jobs.

1618 6.2.11.1 document-format (type2Format)

1619 This attribute identifies the document format of this document.

1620 If the client does not specify this attribute, then the Printer shall
1621 attempt to determine the format in order to decide if the document
1622 data needs to be translated. The version component is optional.

1623 | 6.2.11.2 document-name (namestring)

1624 This attribute contains the name of the document used by the client
1625 to initially identify the document.

1626 | 6.2.11.3 document-URL (nameurl)

1627 This attribute contains the URL of the document if the client
1628 specified the document with a URL.

1629 If this attribute is specified, then document-content shall be
1630 unspecified.

1631 6.2.11.4 document-content (octetString)

1632 This attribute contains the actual contents of the document.

1633 If this attribute is specified, then document-URL shall be
1634 unspecified.

1635 This attribute shall be used during the transmission of the Print
1636 operation over a network. A Printer shall save the document data to a
1637 file and reference it with the document-URL ~~or document path~~
1638 attribute. A Get-Attribute or Get-Jobs operation shall always find
1639 that this attribute is unspecified.

1640 6.3 Operation Attributes (Set by Client)

1641 NOTE: These attributes have just been introduced and they are not as
1642 stable as the attributes in the other sections. Some work is still
1643 needed to show the relationship between these attributes, job
1644 attributes, printer attributes, and authentication and authorization.

1645 The client shall set these attributes and associate them with an
1646 operation rather than an object.

1647 It is intended that a client program rather than an end-user has
1648 control over the setting of these values so that they cannot be
1649 easily forged.

1650 6.3.1 operation-locale (type3Locale)

1651 This attribute identifies the locale of the client. The Printer uses
1652 this attribute to determine the locale of (1) messages in the result
1653 of the operation ~~or~~, (2) in errors returned by the operation or (3)
1654 notification events sent to the submitter.

1655 The standard values are defined in the section on the job-locale
1656 attribute.

1657 If an operation does not specify this attribute, the Printer shall
1658 assume that the operation has the same locale as the Printer.

1659 6.3.2 operation-notification-address (url)

1660 This attribute ~~identifies the specifies~~ both the address and mechanism
1661 for delivery of events. If the URL has a "mailto:" scheme, then
1662 email is used and the rest of the URL is used as the email address.
1663 If the URL has a "http:" scheme, then an HTTP APPEND method is used
1664 to add HTML formatted events to the end of ~~athe~~ specified HTML file.

1665 6.3.3 operation-user-name (name)

1666 This attribute identifies the most authenticated end ~~--~~user name that
1667 the client can supply. This name identifies the end ~~--~~user performing
1668 the operation.

1669 This value shall be set by the system rather than the end-user in
1670 order to minimize the chance of forgery.

1671 6.3.4 operation-host-name (name)

1672 This attribute identifies the most authenticated host name that the
1673 client can supply. This name identifies the host from which the
1674 operation comes.

1675 This value shall be set by the system rather than the end-user in
1676 order to minimize the chance of forgery.

1677 6.4 Printer Attributes (Set by the Administrator)

1678 A printer object may be realized in either a Print Server or Output
1679 Device. Note: How these attribute are set by an Administrator is
1680 outside the scope of this specification.

1681 A Printer Object in an Output Device contains a set of printer object
1682 attributes that represent an Output Device capable of rendering a
1683 document in visible form. Examples include electronic and electro-
1684 mechanical printers such as laser printers, ink-jet printers, and
1685 various kinds of impact printers, but may include other types of
1686 output devices such as microfiche imagers and plotters as well.

1687 | A Printer Object in a Print Server ~~that supplies~~ may supply queuing,
1688 spooling, and scheduling for an Output device that does not queue or
1689 spool.

1690 | A Print Server, in the most common case, controls exactly one
1691 downstream Output Device. The Print Server's Printer object has
1692 attributes whose values are the same as those of the Printer object
1693 in the downstream Output Device.

1694 | A Printer Object in a Print Server ~~contains~~ may contain a set of
1695 printer object attributes that are the union of the Printer objects
1696 in the downstream Output Devices. This object extends the
1697 capabilities of an Output Device. For example, an administrator
1698 might define a single Print Server to represent all of the Output
1699 Devices of the same type and capability in a single location,
1700 associated with a particular server. A end user would normally send
1701 | a print-job to a Print Server -, and allow the Print Server to assign
1702 the job to a particular Output Device based on the relative load and
1703 availability of the printers under its control, thus providing a load
1704 balancing service. However, nothing precludes an administrator from
1705 configuring a print system so that a n end user can send a print-job
1706 directly to an Output ~~Device~~.

1707 | ~~A Print Server, in the most common case, controls exactly one~~
1708 ~~downstream Output Device. The Print Server's Printer object has~~
1709 ~~attributes whose values are the same as those of the Printer object~~
1710 ~~in the downstream Output~~ Device.

1711 The attributes defined in this section provide information about a
1712 particular Printer.

1713 6.4.1 printer-name (name)

1714 This attribute uniquely identifies the printer on its host.

1715 6.4.2 printer-location (string)

1716 This attribute identifies the location of this printer.

1717 6.4.3 printer-model (string)

1718 This attribute identifies the make and model of the printer.

1719 6.4.4 printer-type [s](#) (type2Enum)

1720 This attribute identifies the marking technology of the printer.

1721 The standard value [s](#) for this attribute are the descriptive names
1722 specified by ISO DPA which have corresponding enum symbolic and
1723 numeric values assigned by the Printer MIB (RFC 1759).. These
1724 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

1725

1726 6.4.5 printer-state (type1Enum)

1727 This attribute identifies the current state of the printer [and shall](#)
1728 [be set by the Printer](#). The protocol support all values for printer
1729 states, however a Printer shall only generate the printer states
1730 which are appropriate for the particular implementation.

1731 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 this IPP part of ISO/IEC 10175-V1.0.
shutdown	The printer has been taken out of service, (for a long time), whether for repairs or others reasons. The printer's message generic attribute may be used to record a reason and estimated time for return to service
job-start-wait	The currently processing job was started with the job-start-wait attribute set, and is awaiting operator intervention or time-out.
job-end-wait	The currently processing job was started with the job-end-wait attribute set, and is awaiting operator intervention or time-out.
job-password-wait	The currently processing job was started with the job-password attribute set, and is awaiting the operator or user to enter the password supplied by the job-password attribute.
needs-key-operator	The printer needs the attention of a key operator. Key operator functions are printer-specific, but typically include adding toner or developer, or attending to a hardware fault.
connecting-to-printer	The server has scheduled a job on the printer and is in the process of connecting to a shared network printer (and may not be able to actually start printing the job for an arbitrarily long time depending on the usage of the printer by other servers).
timed-out	The server was able to connect to the printer (or is always connected), but was unable to get a response from the printer in the time specified by the printer's printer-timeout-period attribute.

1732

1733 6.4.6 printer-state-message (string)

1734 This attribute specifies a message that gives further information
1735 about the current printer state ~~---~~ and shall be set by the Printer.

1736 6.4.7 message (string)

1737 This attribute provides a message from an operator, system
1738 administrator or "intelligent" process to indicate to the end user
1739 information or status of the printer, such as why it is unavailable
1740 or when it is expected to be available. ~~---~~

1741 6.4.8 printer-job-templates (1#urlDefault)

1742 This attribute identifies the URL of each of the Job Templates that
1743 this Printer is associated with and the one Job Template this Printer
1744 uses as its default for supply job attributes that the client omits.
1745 There shall be only one value with the default qualifier. Other
1746 Printers can be associated with the same Job Templates.

1747 The syntax is:

1748 url [":" default]

1749 6.4.9 locale (type3Locale)

1750 This attribute specifies the locale that the Printer operates in.

1751 The standard values are defined in the section on the job-locale
1752 attribute.

1753 6.4.10 notification-events (1#type2Enum)

1754 This attribute specifies the events on whose occurrence the Printer
1755 should notify those addresses specified by the notification-addresses
1756 attribute.

1757 If the attribute is unspecified ~~or empty~~, the Printer does not
1758 perform notification, though the Printer still checks the job's
1759 notification-events attribute.

1760 In this attribute, job-problem and printer-problem have the same
1761 meaning.

1762 The standard values are defined in the section on the job's
1763 notification-events attribute.

1764 NOTE - This attribute is intended to notify operators, not end-users.

1765 6.4.11 notification-addresses (~~#name~~1#url)

1766 This attribute specifies the email method and addresses to which the
1767 Printer should send messages when events specified by the
1768 notification-~~events~~ attribute occur.

1769 If the attribute is unspecified ~~or empty~~, the Printer does not
1770 perform notification, though the Printer still checks the job's
1771 notification-events attribute.

1772 NOTE - This attribute is intended to notify operators, not end-users.

1773 6.4.12 end-user-acl (1#name)

1774 This attribute specifies the end users who are allowed to print on
1775 the Printer.

1776 If the attribute is unspecified ~~or empty~~, the Printer allows anyone
1777 to print.

1778 6.4.13 maximum-printer-speed (positiveIntegerUnits)

1779 This attribute indicates the maximum printer speed of the Printer in
1780 units of pages per minute, impressions per minute, lines per minute,
1781 and characters per minute. A job cannot control a Printer's speed,
1782 but a Printer Browser can use printer speed as a criteria.

1783 The standard units are a type2Enum and are: ppm, ipm, spm, lpm, cps.

1784 6.4.14 fonts-substitutions (1#stringPair)

1785 This attribute specifies an appropriate substitute for a font that is
1786 advertised as supported in the fonts-supported attribute, even though
1787 the Printer doesn't actually have the font available.

1788 This attribute consists of a set of font pairs: a font name and the
1789 font to use instead.

1790 If this attribute is unspecified, the Printer does not perform any
1791 font substitutions.

1792 6.4.15 fonts-supported (1#stringState)

1793 This attribute identifies the font resources supported by this
1794 printer and indicates the state of readiness for each font.

1795 The standard names are defined in the section on default-font.

1796 Each item in the list contains the pair consisting of a font name and
1797 a state indicating the font's readiness state.

1798 6.4.16 media-supported (1#nameState)

1799 This attribute identifies the media, media-sizes, input trays, and
1800 electronic forms supported by this printer, and indicates the state
1801 of readiness for each medium resource.

1802

1803 ~~There may be just two states: ready and needs installing, or there~~
1804 ~~may be a third state: needs purchasing.~~

1805 The standard names are defined in the section on the section on the
1806 medium-select.

1807 Standard states are: not-ready, on-order, and special-order. The
1808 omission of a state shall indicate t hat the medium is ready, i.e.,
1809 can be used without human intervention..

1810 6.4.17 document-formats-supported (1#type2FormatState)

1811 This attribute identifies the document-formats, including the
1812 document-format-versions, supported by the Printer. This set includes
1813 both the formats that are native to the Printer and those formats
1814 that the Printer can translate to one that is native to the Printer.
1815 From the client's point of view, this set contains all formats in
1816 which documents can be submitted to this Printer.

1817 Proprietary document format identifiers, and versions are assigned by
1818 the owners of those formats.

1819 The state of readiness for each format is also included, though all
1820 formats should normally always be ready.

1821 6.4.18 numbers-up-supported (1# ~~positiveInteger~~ type3Enum State)

1822 This attribute identifies the number-up values supported by this
1823 printer..

1824 The state of readiness for each number-up value is also included,
1825 though all number-up conversions should always be ready.

1826 6.4.19 finishings-supported (1#type2EnumState)

1827 This attribute identifies the finishing operations supported by this
1828 Printer and states of readiness for each finishing.

1829 The standard finishing objects are defined in the section on the
1830 finishing Job attribute.

1831 6.4.20 sides-supported (1#type2EnumState)

1832 This attribute indicates the values of the sides attribute supported
1833 by this printer and the states of readiness of each value.

1834 The standard values are defined in the section on the sides
1835 attribute.

1836 6.4.21 print-qualities-supported (1#type2EnumState)

1837 This attribute indicates the values of the printer-quality attribute
1838 supported by this printer and the states of readiness for each print-
1839 quality value.

1840 The standard values are defined in the printer-quality attribute.

1841 6.4.22 printer-resolutions-supported (1#positiveIntegerCrossState)

1842 This attribute indicates the values of the printer-resolution-select
1843 attribute supported by this printer and their states of readiness.

1844 The state of readiness for each printer resolution is also included,
1845 though normally all printer-resolutions should always be ready.

1846 The syntax is discussed in the section on the printer-resolution-
1847 select attribute.

1848 6.4.23 code-sets-supported (1#type3EnumState)

1849 This attribute indicates the values of the default-code-set attribute
1850 supported by this printer and the states of readiness for each code-
1851 set.

1852 The standard values are defined in the default-code-set attribute.

1853 | 6.4.24 off-peak-times-supported (1#type3EnumState)

1854 This attribute indicates the values of the job-print-off-peak
1855 attribute supported by this printer and the states of readiness for
1856 each value.

1857 If this attribute is unspecified, then the Printer has no off-peak
1858 periods.

1859 The standard values are defined in the section on the job-print-off-
1860 peak Job attribute.

1861 Note: this document does not define how an administrator associates
1862 the off-peak names with actual time periods.

1863 6.4.25 events-supported (1#type2EnumState)

1864 This attribute indicates the values of the job and printer
1865 notification-events attribute supported by this Printer and the
1866 states of readiness for each value.

1867 If this attribute is unspecified, then the Printer does not support
1868 notification.

1869 The standard values are defined in the section on the notification-
1870 events attribute.

1871 6.4.26 locales-supported (1#type3LocaleState)

1872 This attribute indicates the values of the job-locale attribute
1873 supported by this Printer and the states of readiness for each value.

1874 The standard values are defined in the section on the job-locale
1875 attribute.

1876 6.4.27 job-sheets-supported (1#type3EnumState)

1877 This attribute identifies the job-sheet values supported by this
1878 printer, and the state of readiness for each job-sheet.

1879 To allow no job sheets, the system administrator shall include the
1880 value ~~none~~ "none" as a value for this attribute. The client specifies
1881 that there are no job sheets by using the value ~~none~~ "none" as the
1882 value of the job-sheets attribute.

1883 If the job-sheets attribute is not specified or contains a value
1884 which the Printer does not support, then the server shall select from
1885 among the values of this attribute. The server shall not select the
1886 value ~~none~~ "none" unless it is the only value specified for the job-
1887 sheets-supported attribute.

1888 NOTE - ~~When the client supplies a value other than "none", i~~ t is
1889 preferable for the server to produce some job jobsheet, even if not
1890 the desired one, rather than produce none at all or reject the job.

1891
1892 6.4.28 maximum-copies (positiveInteger)

1893 This attribute indicates the maximum number of copies of a document
1894 that can be rendered by this printer in a single print-job.

1895 If the attribute is unspecified ~~or has a value of 0~~, there is no
1896 limit on the maximum number of copies for this Printer.

1897 6.4.29 maximum-job-octets (positiveInteger)

1898 This attribute indicates that the Printer shall accept a job only if
1899 its size in octets is less than the value specified by this
1900 attribute.

1901 | If the attribute is unspecified ~~or has a value of 0~~, there is no
1902 limit on the size of a job in octets.

1903 6.4.30 maximum-impressions (positiveInteger)

1904 This attribute indicates that the Printer shall accept a job only if
1905 its size in impression is less than the value specified by this
1906 attribute.

1907 | If the attribute is unspecified ~~or has a value of 0~~, there is no
1908 limit on the size of a job in impressions.

1909 6.4.31 maximum-media-sheets (positiveInteger)

1910 This attribute indicates that the Printer shall accept a job only if
1911 its size in media-sheets is less than the value specified by this
1912 attribute.

1913 | If the attribute is unspecified ~~or has a value of 0~~, there is no
1914 limit on the size of a job in media-sheets.

1915 6.4.32 maximum-job-retention-period (deltaTime)

1916 This attribute indicates that when the Printer accepts a job, the
1917 retention period must not exceed the value of this attribute.
1918 Otherwise, the Printer sets the job's retention-period to the value
1919 of this attribute.

1920 If this attribute is unspecified, then the Printer places no limit on
1921 the retention time.

1922 6.4.33 maximum-end-user-priority (type1Enum)

1923 This attribute indicates that when the Printer accepts a job, the
1924 job-priority must not exceed the value of this attribute. Otherwise,
1925 the Printer sets the job's job-priority to the value of this
1926 attribute.

1927 | If this attribute is unspecified, then the Printer places no limit on
1928 the job-priority ~~time.~~.

1929 The standard values are defined in the section on the job-priority
1930 attribute.

1931 6.4.34 queued-job-count (positiveInteger-cardinal)

1932 This attribute contains a count of the number of jobs that are either
1933 pending and/or processing and shall be set by the Printer .

1934 6.4.35 scheduling-algorithm (type3Enum)

1935 This attribute indicates the current scheduling algorithm for this
1936 Printer: ~~"none", "shortest job first", "time received", etc~~ .
1937 Standard values are: "none", "smallest-job-first", "time -received" .

1938 6.5 Job Templates

1939 The attributes for a Job Template can be any of the Job object
1940 attributes defined in the sections:

1941 Job Sheet Attributes
1942 Notification Attributes
1943 Job Scheduling Attributes
1944 (except job-print-after)
1945 Job Production Attributes
1946 (except page-select)
1947 Attributes for Conversion of Text and HTML Files
1948

1949 6.6 Conformance

1950 A conforming implementation shall implement all operations, objects
1951 and attributes defined in this document. ~~IPP is explicitly designed to~~
1952 ~~be extensible. This means that in addition to the attributes defined~~
1953 ~~in this specification, specific implementation~~

1954 ~~instances may support not only the basic protocol as defined in this~~
1955 ~~specification, but might add vendor specific extensions.~~

1956 Also, for the core set of attributes listed in this specification, it
1957 is not required that a conforming server support all (standard)
1958 values of all supported attributes. For example, it is not required
1959 that a printer implement all finishing methods indicated by the
1960 standard values.

1961 The explicit requirement of the term "supported", with respect to one
1962 of the attributes that deal with printer functions or resources, is
1963 that the server shall recognize the attribute and those values that
1964 are supported, and shall be able to respond to a query about which
1965 values that printer does, in fact, support.

1966 IPP is explicitly designed to be extensible. Additional attributes
1967 can be proposed to be registered by going through the type 2 enum
1968 process which will register their specification after approval with
1969 IANA. In addition specific implementation instances may support not
1970 only the basic protocol as defined in this specification, but may add
1971 vendor-specific private extensions by prefixing attribute-names with

1972 | their company name registered with IANA for use in domains. See
1973 | attribute syntax section. However, such private extensions shall not
1974 | duplicate attribute semantics already in this specification.

1975 | 7. Security Considerations

1976 | This protocol does not identify any new authentication mechanisms.
1977 | The authentication mechanisms built into HTTP (such as SSL and SHTTP)
1978 | are recommended.

1979 | This protocol does define a simple authorization mechanism by
1980 | introducing the "end-user- acl" attribute as part of the Printer
1981 | object. This ACL attribute is a multi-valued list of all of the
1982 | authenticated names of end-users. This protocol does not specify
1983 | what the domain is for names in this ACL attribute.

1984 | Issue: Will it always be possible for a Printer to obtain a
1985 | meaningful authenticated name that the Printer can match against the
1986 | end-user-acl, or will some other mechanism be necessary, such as a
1987 | password?

1988 | 8. References

- 1989 | [1] Smith, R., Wright, F., Hastings, T., Zilles, S., and
1990 | Gyllenskog, J., "Printer MIB", RFC 1759, March 1995.
1991 |
1992 | [2] Berners-Lee, T, Fielding, R., and Nielsen, H., "Hypertext
1993 | Transfer Protocol - HTTP/1.0", RFC 1945, August 1995.
1994 |
1995 | [3] Crocker, D., "Standard for the Format of ARPA Internet Text
1996 | Messages", RFC 822, August 1982.
1997 |
1998 | [4] Postel, J., "Instructions to RFC Authors", RFC 1543, October
1999 | 1993.
2000 |
2001 | [5] ISO/IEC 10175 Document Printing Application (DPA), Final, June
2002 | 1996.
2003 |
2004 | [6] Herriot, R. (editor), X/Open A Printing System Interoperability
2005 | Specification (PSIS), August 1995.
2006 |
2007 | [7] Kirk, M. (editor), POSIX System Administration - Part 4:
2008 | Printing Interfaces, POSIX 1387.4 D8, 1994.
2009 |
2010 | [8] Borenstein, N., and Freed, N., "MIME (Multi-purpose Internet
2011 | Mail Extensions) Part One: Mechanism for Specifying and
2012 | Describing the Format of Internet Message Bodies", RFC 1521,
2013 | September, 1993.
2014 |
2015 | [9] Braden, S., "Requirements for Internet Hosts - Application and
2016 | Support", ~~FR~~RFC 1123, October, 1989,
2017 |

2018 [10] McLaughlin, L. III, (editor), "Line Printer Daemon Protocol"
2019 RFC 1179, August 1990.

2020
2021 [11] Berners-Lee, T., Masinter, L., McCahill, M., "Uniform Resource
2022 Locators (URL)", RFC 1738, December, 1994.

2023
2024 9. Author's Address

2025 Scott A. Isaacson
2026 Novell, Inc.
2027 122 E 1700 S
2028 Provo, UT 84606
2029
2030 Phone: 801-861-7366
2031 Fax: 801-861-4025
2032 EMail: scott_isaacson@novell.com
2033

2034 Tom Hastings
2035 Xerox Corporation
2036 701 S. Aviation Blvd.
2037 El Segundo, CA 90245
2038
2039 Phone: 310-333-6413
2040 Fax: 310-333-5514
2041 EMail: hastings@cp10.es.xerox.com
2042

2043 Robert Herriot
2044 Sun Microsystems Inc.
2045 2550 Garcia Ave., MPK-17
2046 Mountain View, CA 94043
2047
2048 Phone: 415-786-8995
2049 Fax: 415-786-7077
2050 Email: robert.herriot@eng.sun.com
2051

2052 Roger deBry
2053 HUC/003G
2054 IBM Corporation
2055 P.O. Box 1900
2056 Boulder, CO 80301-9191
2057
2058 Phone: (303) 924-4080
2059 Fax: (303) 924-9889
2060 Email: debry@vnet.ibm.com
2061

2062 Other ~~Contributors~~ Participants:

2063 Devon Taylor, Novell, Inc.
2064 Mike MacKay, Novell, Inc.
2065 Peter Zehler, Xerox, Corp.
2066 Keith Carter, IBM Corporation
2067 Carl-Uno Manros, Xerox, Corp.
2068 ~~<add the list of IPP attendees and participants>~~ Don Wright - Lexmark

2069 | Steve Gebert - IBM
2070 | Ray Lutz - Cognisys
2071 | Mabry Dozier - QMS
2072 | Lee Ferrel - Canon
2073 | Hiro Sato - Canon
2074 | Pat Nogay - IBM
2075 | Jim Walker - Dazel
2076 | Jay Martin - Underscore
2077 | Bill Wagner - DPI
2078 | Stan McConnell - Xerox
2079 | Bob Setterbo - Adobe
2080 | Randy Turner - Sharp
2081 | Rob Whittle - Novell
2082 | Ron Bergman - Data Products
2083 | Lloyd Young - Lexmark
2084 | Andy Davidson - Tektronix
2085 | Rick Landau - Digital
2086 | David Kellerman - Northlake Software
2087 | David Roach - Unisy s
2088 | Mike Timperman - Lexmark
2089 | Chris Wellens - Interworking Labs
2090 | Pete Loya - HP
2091 | Bob Pentecost - HP
2092 | Harry Lewis - IBM
2093 | William Wagner - Digital Products
2094 | Atsushi Yuki - Kyocera
2095 | Rob Rhoads - Intel
2096 | Jeff Barnett - IBM
2097 | Jim Walker - Dazel
2098 | Jeff Copeland - QMS
2099 | Chuck Adams - Tektronix
2100 |
2101 |

2102 |

2103 10. Appendix A: Sample IPP Operations

2104 The following examples illustrate typical flows using the IPP
 2105 protocol. In these examples, the IPP Printer object named "printer-1"
 2106 is located at the node identified by the DNS name " some.domain.com".
 2107 A ~~Job~~ Job Template has been defined for printer-1 which establishes the
 2108 print defaults.

2109 For brevity in the following flows, none of the HTTP headers are
 2110 shown. CRLF sequences are not shown.

2111 10.1 Querying the printer

2112 ~~Client~~ some.domain.com

2113 Client some.domain.com

2114

2115 ----->

2116 Post http://some.domain.com/printer-1 http/1.0

2117 ~~Get~~ ~~Attributes IPP/1.0~~

2118 ~~Printer state :-~~

2119 ~~Sides supported :-~~

2120 ~~Media supported :-~~

2121 ~~Document formats supported :-~~

2122

2123 <-----

2124 ~~-Attributes IPP/1.0~~

2125 ~~printer-state :~~

2126 ~~sides-supported :~~

2127 ~~media-supported :~~

2128 ~~document-formats-supported :~~

2129

2130 <-----

2131 http/1.0 201 "Created" (a response)

2132 IPP/1.0 xxx "attribute list returned"

2133 ~~Printer state : running~~

2134 ~~Sides supported : 1 sided~~

2135 ~~Media supported : iso a4 white, iso b4 white~~

2136 ~~Document formats supported : Postscript/2.0~~ printer-state : idle

2137 sides-supported : 1-sided

2138 media-supported : iso-a4-white, iso-b4-white

2139 document-formats-supported : Postscript/2.0

2140

2141

2142

2143

2144

2145

2146

2147

2148

2149

2150
2151
2152
2153
2154
2155
2156
2157 10.2 Print Operation - with print data included

2158 Client some.domain.com
2159 Client some.domain.com
2160

2161 ----->
2162 Post http://some.domain.com/printer-1 http/1.0
2163 Print IPP/1.0
2164 Print-Job-Object Header
2165 Job name : My Job
2166 Medium : iso-a4 white
2167 Njob-name : My Job
2168 medium : iso-a4-white
2169 notification-events : Job-completion
2170 Notification-address : joe@pc.domain.com
2171 Document Header
2172 Document-name : Letter to Mom
2173 Document-Content Header (content type = Postscript/2.0)
2174 Document in Postscript level 2 format
2175

2176
2177 <Document in Postscript level 2 format>
2178
2179

2180 <-----
2181 http/1.0 200 "accepted"
2182 IPP/1.0 xxx "print job accepted and queued"
2183 Job-identifier : some.domain.com/printer-1/0037
2184 Current-job-state : pending
2185 Printer-state : running
2186
2187 ~~10.3 Print Operation with no data included~~

2188 Client some.domain.com
2189
2190 current-job-state : pending
2191 printer-state : needs- attention
2192
2193

2194 10.3 Print Operation - with no data included

2195 Client some.domain.com

```

2196
2197 ----->
2198 Post http://some.domain.com/printer-1 http/1.0
2199   Print IPP/1.0
2200   Print-Job-Object Header
2201     Job name : My Job
2202     Medium : iso-a4-white
2203     Njob-name : My Job
2204     medium : iso-a4-white
2205     notification-events : Job-completion
2206     Nnotification-address : joe@some.domain.com
2207   Document Header
2208     Ddocument-name : Letter to Mom
2209     Ddocument-URL : joe@pc.domain.com/ Docs/To-mom.ps
2210
2211 <-----
2212 http/1.0 200 "accepted"
2213   IPP/1.0 xxx "print job accepted and queued"
2214     Job-identifier : some.domain.com/printer-1/0037
2215     Current-job-state : pending
2216     Printer-state : running
2217     current-job-state : pending
2218     printer-state : processing
2219 10.4 Querying the state of the job
2220
2221 In this example, no attributes are specified, so all job attributes
2222 are returned.
2223
2224 Client some.domain.com
2225
2226 ----->
2227 Post http://some.domain.com/printer-1/0037 http/1.0
2228   Get Attributes IPP/1.0
2229
2230 <-----
2231 http/1.0 201 "Created" (a response)
2232   IPP/1.0 xxx "attribute list returned"
2233     Job Name : My Job
2234     Job Originator : Joe@some.domain.com
2235     Job-originating host : pc.domain.com
2236     Notification address : joe@pc.domain.com
2237     Job locale : xx:xx:xx
2238     Current job status : printing
2239     Printer assigned : printer 1
2240     Submission time : 1214
2241     Media sheets completed : 2
2242

```

2243 ~~10.5 Canceling a Job~~

2244 Client some.domain.com

2245 job-Name : My Job
2246 job-Originator : Joe@some.domain.com
2247 job-originating-host : pc.domain.com
2248 notification-address : joe@pc.domain.com
2249 job-locale : xx:xx:xx
2250 current-job-status : printing
2251 submission-time : 1996 Nov 22 1214
2252 media-sheets-completed : 2

2255 10.5 Canceling a Job

2256 Client some.domain.com

2257 ----->
2258 Post: http://some.domain.com/printer-1/0037
2259 Cancel ~~Job~~ IPP/1.0

2261 <-----
2262 -Job IPP/1.0

2266 <-----
2267 http/1.0 200 "okay"
2268 Current-job-state : terminating

2279 10.6 Listing jobs on a Printer

2280 List jobs on printer-1, only return job sizes. Jobs are returned in
2281 the order they are scheduled for printing. A Job-identifier attribute
2282 precedes the attributes returned for each job to delimit job
2283 boundaries.

2284
2285 Client some.domain.com

2286 ----->

```
2287 Post http/1.0 some.domain.com/printer-1
2288   Get Jobs IPP/1.0
2289   total-job-octets :
2290
2291   ←-----
2292   -Jobs IPP/1.0
2293   total-job-octets :
2294
2295   <-----
2296 http/1.0 201 "Created" (a response)
2297   IPP/1.0 xxx "created an attribute list"
2298   ⚡job-identifier : 0033
2299   total-job-octets : 4567
2300   ⚡job-identifier : 0034
2301   total-job-octets : 12345
2302   ⚡job-identifier : 0035
2303   total-job-octets : 12356
```

2304