

Internet-Draft

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

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

Status of this Memo

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

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

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

Abstract

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

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

The IPP protocol covers only end user operations on basic print service objects. Authentication is realized by mechanisms outside the scope of the protocol, but the protocol does introduce some access control functionality so that only authorized end users are allowed to submit print jobs to printers whose implementation and site policy support access control. Also, the Cancel Job

operation requires some authentication so that jobs can only be canceled by the end user who submitted the job. Extended monitoring and management is possible through other protocols such as the SNMP Printer MIB. In the areas where there are no existing standards, some proposed and emerging standards are being worked (management, security, etc.). As these services become more stable, this document (and hence the protocol) can be updated to reflect the integration and relationships with these other standards.

## Table of Contents

1. Introduction .....	5
2. Distributed Printing .....	5
2.1 Generic Print System Components .....	5
2.2 IPP Components .....	6
3. IPP Objects .....	6
3.1 Printer .....	7
3.2 Job .....	8
3.3 Job Template .....	9
3.4 Object Relationships .....	9
3.5 Object Identity .....	9
4. Naming .....	9
4.1 Directory Services .....	10
4.2 Directory Entry Schema .....	11
4.2.1 Name .....	11
4.2.2 Description .....	11
4.2.3 Location .....	11
4.2.4 Maximum Print Quality .....	11
4.2.5 Cost .....	11
4.2.6 Resolution .....	11
4.2.7 Color Supported .....	12
4.2.8 Fonts Supported .....	12
4.2.9 Maximum Speed .....	12
4.2.10 Device Id .....	12
4.2.11 Make and Model .....	12
4.2.12 Marker Type .....	12
4.2.13 Document Formats Supported .....	12
4.2.14 Sides Supported .....	13
4.2.15 Finishings Supported .....	13
4.3 Directory Entries Using LDAP .....	13
5. IPP Operations .....	14
5.1 HTTP Overview .....	14
5.2 IPP Operation Encoding .....	15
5.2.1 HTTP Request-Header Fields .....	15
5.2.1.1 IPP Request-Line .....	16
5.2.2 HTTP Response-Header Fields .....	16
5.2.2.1 IPP Status-Line .....	16
5.3 The Print Job .....	16
5.3.1 Print Job Object Header .....	17
5.3.2 Document Header .....	17
5.3.3 Document-Content Header .....	17
5.3.4 Job Attributes .....	17
5.3.5 Document Attributes .....	18
5.4 Operation Semantics .....	18
5.4.1 Print Operation .....	18
5.4.1.1 Print Request .....	18
5.4.1.2 Print Response .....	18
5.4.2 Cancel Job Operation .....	19

110	5.4.2.1 CancelJob Request .....	19
111	5.4.2.2 CancelJob Response .....	19
112	5.4.3 Get Attributes Operation .....	19
113	5.4.3.1 GetAttributes Request .....	19
114	5.4.3.2 GetAttributes Response .....	20
115	5.4.4 Get Jobs Operation .....	20
116	5.4.4.1 GetJobs .....	20
117	5.4.4.2 Get Jobs Response .....	20
118	6. Object Attributes .....	20
119	6.1 Attribute Syntaxes .....	21
120	6.2 Job Attributes .....	22
121	6.2.1 Job Informational Attributes (Set by a Client/End User) ..	22
122	6.2.1.1 job-name (string) .....	22
123	6.2.2 Job Informational Attributes (Set by a Printer) .....	22
124	6.2.2.1 job-identifier (url) .....	22
125	6.2.2.2 job-originator (name) .....	23
126	6.2.2.3 job-originating-host (name) .....	23
127	6.2.2.4 notification-address (name) .....	23
128	6.2.2.5 job-locale (type3Locale) .....	23
129	6.2.3 Job Status Attributes (Set by Printer) .....	23
130	6.2.3.1 current-job-state (type1Enum) .....	24
131	6.2.3.2 output-device-assigned (name) .....	25
132	6.2.3.3 submission-time (dateTime) .....	26
133	6.2.3.4 job-message-from-administrator (string) .....	26
134	6.2.3.5 completion-time (dateTime) .....	26
135	6.2.3.6 job-state-reasons (1#type2Enum) .....	26
136	6.2.3.7 impressions-completed (cardinal) .....	27
137	6.2.3.8 media-sheets-completed (cardinal) .....	27
138	6.2.4 Job Sheet Attributes (Set by Client/End User) .....	27
139	6.2.5 Notification Attributes (Set by a Client/End User) .....	27
140	6.2.5.1 notification-events (#type1Enum) .....	27
141	6.2.6 Job Scheduling Attributes (Set by Client/End User) .....	28
142	6.2.6.1 job-priority (type1Enum) .....	28
143	6.2.6.2 job-print-after (dateTime) .....	28
144	6.2.6.3 job-print-off-peak (type3Enum) .....	29
145	6.2.6.4 job-retention-period (deltaTime) .....	29
146	6.2.7 Job Production Attributes (Set by Client/End User) .....	29
147	6.2.7.1 medium-select (type2Enum) .....	30
148	6.2.7.2 number-up (positiveInteger) .....	32
149	6.2.7.3 finishing (type2Enum) .....	33
150	6.2.7.4 sides (type2Enum) .....	34
151	6.2.7.5 copies (positiveInteger) .....	34
152	6.2.7.6 printer-resolution-select (positiveIntegerCross) .....	34
153	6.2.7.7 print-quality (type2Enum) .....	34
154	6.2.7.8 page-select (positiveIntegerRange) .....	34
155	6.2.7.9 files-are-one-document (Boolean) .....	34
156	6.2.7.10 files-are-interleaved (Boolean) .....	35
157	6.2.8 Attributes for Conversion of Text Files (Set by Client/End	
158	User) .....	35
159	6.2.8.1 width (cardinalUnits) .....	35
160	6.2.8.2 length (cardinalUnits) .....	35
161	6.2.8.3 left-margin (cardinalUnits) .....	35
162	6.2.8.4 right-margin (cardinalUnits) .....	35
163	6.2.8.5 top-margin (cardinalUnits) .....	35
164	6.2.8.6 bottom-margin (cardinalUnits) .....	36
165	6.2.8.7 repeated-tab-stops (cardinalUnits) .....	36
166	6.2.8.8 header-text (string) .....	36
167	6.2.8.9 footer-text (string) .....	36
168	6.2.8.10 font-size (cardinalUnits) .....	36
169	6.2.8.11 number-pages (Boolean) .....	36

170	6.2.8.12 default-code-set (type3Enum) .....	36
171	6.2.8.13 content-orientation (type2Enum) .....	36
172	6.2.9 Job Resource Attributes (Set by the program that produces	
173	or senses the PDL) .....	37
174	6.2.9.1 document-format-used (1#type2Format) .....	38
175	6.2.9.2 fonts-used (1#string) .....	38
176	6.2.9.3 code-sets-used (1#type3Enum) .....	38
177	6.2.9.4 media-used (1#type2Enum) .....	38
178	6.2.9.5 sides-used (type2Enum) .....	39
179	6.2.9.6 print-quality-used (type2Enum) .....	39
180	6.2.9.7 finishing-used (type2Enum) .....	39
181	6.2.9.8 printer-resolution-used (positiveIntegerCrossState) ..	39
182	6.2.9.9 total-job-octets (positiveInteger) .....	39
183	6.2.9.10 job-impression-count (positiveInteger) .....	39
184	6.2.9.11 job-media-sheet-count (positiveInteger) .....	39
185	6.2.9.12 job-intervening-jobs (positiveInteger) .....	39
186	6.2.10 Number of Documents (Set by Client) .....	39
187	6.2.10.1 number-of-documents (positiveInteger) .....	40
188	6.2.11 Document Data (Set by a Client/End User) .....	40
189	6.2.11.1 document-format (type2Format) .....	40
190	6.2.11.2 document-name (name) .....	40
191	6.2.11.3 document-URL (name) .....	40
192	6.2.11.4 document-content (octetString) .....	40
193	6.3 Operation Attributes (Set by Client) .....	41
194	6.3.1 operation-locale (type3Locale) .....	41
195	6.3.2 operation-notification-address (url) .....	41
196	6.3.3 operation-user-name (name) .....	41
197	6.3.4 operation-host-name (name) .....	41
198	6.4 Printer Attributes (Set by the Administrator) .....	41
199	6.4.1 printer-name (name) .....	42
200	6.4.2 printer-location (string) .....	42
201	6.4.3 printer-model (string) .....	42
202	6.4.4 printer-types (type2Enum) .....	42
203	6.4.5 printer-state (type1Enum) .....	43
204	6.4.6 printer-state-message (string) .....	44
205	6.4.7 message (string) .....	44
206	6.4.8 locale (type3Locale) .....	44
207	6.4.9 notification-events (#type2Enum) .....	45
208	6.4.10 notification-addresses (#name) .....	45
209	6.4.11 end-user-acl (#name) .....	45
210	6.4.12 maximum-printer-speed (positiveIntegerUnits) .....	45
211	6.4.13 fonts-substitutions (#stringPair) .....	45
212	6.4.14 fonts-supported (1#stringState) .....	45
213	6.4.15 media-supported (1#nameState) .....	46
214	6.4.16 document-formats-supported (1#type2FormatState) .....	46
215	6.4.17 numbers-up-supported (1#positiveIntegerState) .....	46
216	6.4.18 finishings-supported (#type2EnumState) .....	46
217	6.4.19 sides-supported (1#type2EnumState) .....	46
218	6.4.20 print-qualities-supported (1#type2EnumState) .....	47
219	6.4.21 printer-resolutions-supported	
220	(1#positiveIntegerCrossState) .....	47
221	6.4.22 code-sets-supported (1#type3EnumState) .....	47
222	6.4.23 off-peak-times-supported (#type3EnumState) .....	47
223	6.4.24 events-supported (#type2EnumState) .....	47
224	6.4.25 locales-supported (1#type3LocaleState) .....	48
225	6.4.26 job-sheets-supported (#type3EnumState) .....	48
226	6.4.27 maximum-copies (positiveInteger) .....	48
227	6.4.28 maximum-job-octets (positiveInteger) .....	48
228	6.4.29 maximum-impressions (positiveInteger) .....	48
229	6.4.30 maximum-media-sheets (positiveInteger) .....	49

230	6.4.31 maximum-job-retention-period (deltaTime)	49
231	6.4.32 maximum-end-user-priority (type1Enum)	49
232	6.4.33 queued-job-count (positiveInteger)	49
233	6.4.34 scheduling-algorithm (type3Enum)	49
234	6.5 Job Templates	49
235	6.6 Conformance	50
236	7. Security Considerations	50
237	8. References	50
238	9. Author's Address	51
239	10. Appendix A: Sample IPP Operations	53
240	10.1 Querying the printer	53
241	10.2 Print Operation - with print data included	53
242	10.3 Print Operation - with no data included	54
243	10.4 Querying the state of the job	54
244	10.5 Canceling a Job	54
245	10.6 Listing jobs on a Printer	55
246		

## 247 1. Introduction

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

## 259 2. Distributed Printing

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

### 269 2.1 Generic Print System Components

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

- 273 - End Users: End Users are humans (or agents or applications who  
 274 work on behalf of a human) who submit print jobs.
- 275 - Print clients: Print clients are computer network nodes with  
 276 which humans interact in order to manipulate the distributed  
 277 print service. A print client uses some protocol to invoke  
 278 print service operations on another node. Each operation has  
 279 arguments and results associated with it. The print client  
 280 provides arguments which add information about the operation

281 requested, and receives results which describe the status and  
282 outcome of the operation.

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

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

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

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

## 307 2.2 IPP Components

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

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

317  
318 Clients use the following operations:

- 319 - Print (section 5.4.1)
- 320 - Cancel Job (section 5.4.2)
- 321 - Get Attributes (section 5.4.3)
- 322 - Get Jobs (section 5.4.4)

## 323 324 3. IPP Objects

325 This section describes the IPP objects.

## 3.1 Printer

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

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

Some examples of configurations containing IPP Printer object include:

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

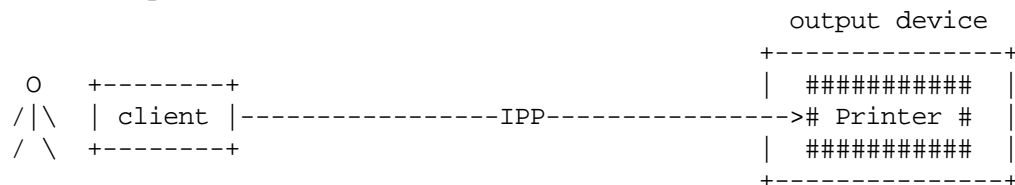
See the following figures for some examples on how to view IPP Printer objects on top of other printing system models:

Legend:

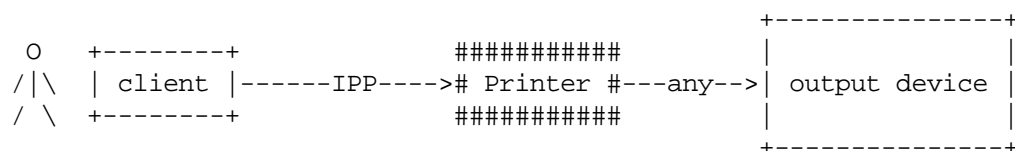
##### indicates an IPP printer object which is either embedded in an output device or is hosted in a server. An IPP printer object may or may not queue/spool.

any indicates any network protocol or direct connect, including IPP

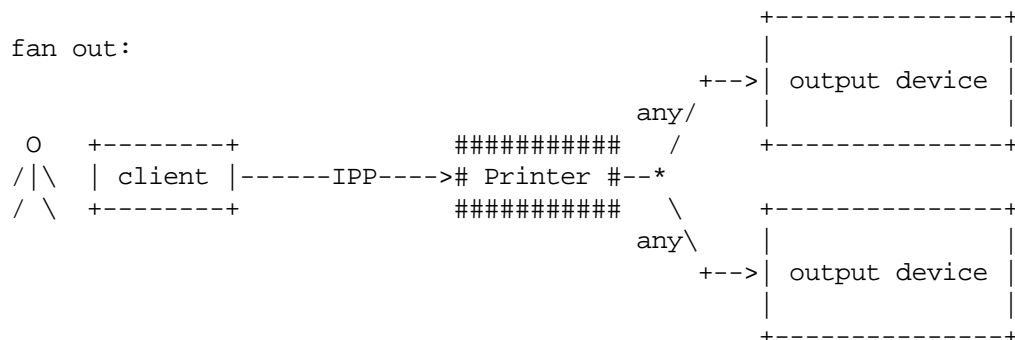
embedded printer:



hosted printer:



fan out:



### 3.2 Job

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

Job attributes are broken up into the following groups:

- 411 - Job Informational (sections 6.2.1, 6.2.2)
- 412 - Job Status (section 6.2.3)
- 413 - Job Sheet (section 6.2.4)
- 414 - Notification (section 6.2.5)
- 415 - Job Scheduling (section 6.2.6)
- 416 - Job Production (section 6.2.7)
- 417 - Conversion of Text Files (section 6.2.8)
- 418 - Job Resource (section 6.2.9)
- 419 - Number of Documents (section 6.2.10)
- 420 - Document Attributes (6.2.11)
- 421

### 422 3.3 Job Template

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

426 ISSUE: The notion of Job Template needs more work. It is  
 427 currently believed that when a client needs to present a Print  
 428 Dialog box to an end-user, it gets potential job values and  
 429 default job values from a Printer. The default values are from  
 430 the Job Template associated with the Printer named by the end-  
 431 user. If a end-user sends a job to a Printer, the Printer may set  
 432 unspecified attributes to the value of the associated Job  
 433 Template.

### 434 3.4 Object Relationships

436 Instances of objects within the system have relationships which  
 437 must be maintained persistently along with the persistent storage  
 438 of the objects themselves. A Printer can contain zero, or more  
 439 Job objects. A Job object contains one or more Documents. A  
 440 Printer object is associated with zero or more Job Template  
 441 objects.

### 442 3.5 Object Identity

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

445 The following objects have the following mandatory identifier  
 446 attributes:

447 Object	Identifier	Containing Object
448 Printer	printer-name	None
449 Job	job-identifier	Printer
450 Job Template	job-template-name	None
451		
452		

## 453 4. Naming

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

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

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

461 Another example is the following URL:

462 http://1.2.3.4:380/printer-2

463 In this case, the URL identifies the use of the HTTP protocol.  
464 The Printer is located at the node identified by the IP address of  
465 "1.2.3.4" using port 380 for the HTTP server, and "printer-2" is  
466 the name of the Printer.

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

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

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

475  
476 These look like two different Printers , but underneath they  
477 represent the same Printer object, but that Printer object has  
478 two Job Templates and each is exposed through a different URL  
479 for the same Printer object. Each one the the associated Job  
480 Templates would have two different Job Template default  
481 attribute sets.

482  
483 - The Job Template can be exposed along with the name of the  
484 Printer object directly in the URL as in:  
485 http://some.domain.com/hr-printer/resumes. In this case there  
486 is a "resumes" Job Template associated with the "hr-printer"  
487 Printer.

488

489 This specification establishes, through IANA, a new well known  
490 port, port 380, for the use of IPP over HTTP. The purpose of this  
491 new well known port would be to distinguish printing from non-  
492 printing content. While any acceptable HTTP content could be  
493 inter-mixed over HTTP well known port 80, only HTTP printing would  
494 be acceptable on port 380.

#### 495 4.1 Directory Services

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

504  
505

## 4.2 Directory Entry Schema

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

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

### 4.2.1 Name

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

### 4.2.2 Description

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

### 4.2.3 Location

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

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

### 4.2.4 Maximum Print Quality

This indicates a somewhat subjective evaluation of the overall printing quality: "high", "medium", or "low".

### 4.2.5 Cost

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

### 4.2.6 Resolution

This is the maximum resolution of the Printer in dpi.

546 The syntax shall be the same as that of the printer-resolution-  
547 select job attribute. That syntax allows a single integer to  
548 specify the maximum resolution or a pair of integers to specify  
549 the maximum resolution when the x and y dimensions differ. When  
550 two integers are specified, the first is in the x direction, ie.,  
551 the direction fo the shortest dimension of the medium, so that the  
552 value is independent of whether the Printer feeds long edge or  
553 short edge first.

#### 554 4.2.7 Color Supported

555 This is a either a BOOLEAN for either yes, color printing is  
556 supported, or no color printing is not supported or it is an  
557 enumerated value such as "highlight", "four color", "mono", etc.

#### 558 4.2.8 Fonts Supported

559 This attribute takes on a list of fonts that are supported by the  
560 printer. This is replicated from the fonts-supported attribute in  
561 the Printer object.

#### 562 4.2.9 Maximum Speed

563 This is the maximum speed of the printer ppm, ipm, lpm, or cps.  
564 They syntax and values are the same as the maximum-printer-speed  
565 Printer attribute.

#### 566 4.2.10 Device Id

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

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

#### 574 4.2.11 Make and Model

575 This is a simple text string defined by the manufacturer that  
576 contains some reference to the make and model of the entity being  
577 represented to the end-user by this Printer object. Is is  
578 recommended that the manufacturer use some regular form such as  
579 "vendor-x/super-duper-printer".

#### 580 4.2.12 Marker Type

581 This is the printing mechanism of the print device: laser, ink  
582 jet, thermal, etc. The syntax and values are the same as for the  
583 printer-types Printer attribute, except the value of the  
584 attribute shall be single-valued

#### 585 4.2.13 Document Formats Supported

586 This is a list of all of the document formats that the printer  
587 and/or its interpreter(s) support. The syntax and values are the

588 same as those for the document-formats-supported Printer  
589 attributes.

#### 590 4.2.14 Sides Supported

591 This attribute specifies the capabilities of the Printer for  
592 marking on sides of the medium. The syntax and values shall be  
593 the same as the sides-supported Printer attribute. Standard  
594 values are: 1-sided (simplex), 2-sided-long- edge (duplex), and 2-  
595 sided-short- edge (tumble).

#### 596 4.2.15 Finishings Supported

597 This attribute identifies the finishing operations supported by  
598 the Printer. The standard finishing objects are defined in the  
599 section on the finishing job attribute.

### 600 4.3 Directory Entries Using LDAP

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

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

610 To create a Printer object directory entry using LDAP:

611 1. An administrator uses a program to create an entry for the  
612 Printer object on a directory server that supports LDAP. The  
613 administrator defines the Distinguished Name (dn) and the default  
614 subjective attributes for the Printer object directory entry.  
615 Issue: Should the administrator also define default objective  
616 attributes or wait for the Printer object itself to initialize  
617 these attributes?

618 2. The Printer object invokes the ldap\_open API to open a  
619 connection to the directory server:

620 Example: ld=ldap\_open ("dir.host.name", LDAP\_PORT)

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

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

624 Example: ldap\_simple\_bind\_s (ld, dn, NULL) (which does a simple  
625 authentication without a password).

626 4. The Printer object invokes the ldap\_modify or ldap\_modify\_s API  
627 to define the objective attributes for the Printer object entry as  
628 identified by its Distinguished Name (dn).

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

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

Example: `ldap_unbind (ld)`

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

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

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

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

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

## 5. IPP Operations

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

### 5.1 HTTP Overview

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

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

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

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

## 678 5.2 IPP Operation Encoding

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

681 HTTP MESSAGE = Request | Response

682  
683 Requests and responses use the generic message format of RFC 822  
684 for transferring entities. Both messages may include optional  
685 header fields and an entity body. The entity body is separated  
686 from the headers by a null line (a line with nothing preceding the  
687 CRLF).

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

695  
696 Response = Status-line  
697       \* (General-Header  
698       | Request-Header  
699       | Entity-Header)  
700       CRLF  
701       [ Entity-Body ]

702  
703 All IPP headers conform to the syntax

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

705  
706 IPP/1.0 defines the octet sequence CRLF as the end-of-line marker  
707 for all protocol elements except the entity-body.

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

### 713 5.2.1 HTTP Request-Header Fields

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

720  
721 HTTP Entity-Header = Content-Encoding

```

722         | Content-Length
723         | Content-Type
724         | extension-header
725

```

726 The Content-Length field must always be a valid length, This means  
 727 that for any Print Operations based on HTTP 1.0, the entire  
 728 content must be generated before this header can be built. HTTP  
 729 1.1 provides the notion of "chunks" which will allow the content  
 730 to be generated dynamically as the data is sent.

731  
 732 Content-Type will always be "Application/IPP".

#### 733 5.2.1.1 IPP Request-Line

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

```

737
738     IPP Request-Line =      Operation token  IPP/1.0  CRLF
739
740     Operation token   =      Print | CancelJob | GetAttributes |
741                             GetJobs
742

```

#### 743 5.2.2 HTTP Response-Header Fields

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

##### 748 5.2.2.1 IPP Status-Line

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

```

752
753     IPP Status-Line = IPP/1.0 Status-Code Reason-Phrase CRLF

```

#### 754 5.3 The Print Job

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

```

761
762     Print Job = Print-Job-Object-Header      ;section (5.3.1)
763                [Job Attributes]             ;section (5.3.4)
764                *(Documents)
765
766
767     Document = Document-Header              ;section (5.3.2)
768                [Document attributes]        ;section (5.3.5)
769                [Content-Header]             ;section (5.3.3)

```

770 content]

771

### 772 5.3.1 Print Job Object Header

773 Print-Job-Object Header = Content-Encoding

774 | Content-Length

775 | Content-Type

776 | extension-header

777

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

### 780 5.3.2 Document Header

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

786 Document Header = Content-Encoding

787 | Content-Length

788 | Content-Type

789 | extension-header

790

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

### 793 5.3.3 Document-Content Header

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

799

800 Document-Content-Header = Content-Encoding

801 | Content-Length

802 | Content-Type

803 | extension-header

804

805 Content-Type is defined as :

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

807

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

810 Content-Type: Postscript/2.0

811

812 Other header fields are as defined by HTTP 1.0.

### 813 5.3.4 Job Attributes

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

816  
 817       Job-Attribute = attribute name ":" Attribute value CRLF  
 818  
 819       Attribute value = Value | \*(Value "," Value)  
 820  
 821

#### 822 5.3.5 Document Attributes

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

825  
 826       Document-Attribute = attribute name ":" Attribute value CRLF  
 827  
 828       Attribute value = Value | \*(Value "," Value)  
 829

### 830 5.4 Operation Semantics

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

#### 833 5.4.1 Print Operation

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

836       Note that the Printer name is not needed since it is the target of  
 837       the entire operation. A Print Job contains the information needed  
 838       by the Print object to print a document or set of documents. When  
 839       the print operation is invoked, the Entity-Body in the HTTP  
 840       request includes an IPP Print Job. The concrete syntax of the  
 841       Print Job is defined in section 6.2.

##### 842 5.4.1.1 Print Request

844       The following abstract data types are part of the Print Request.

Job and Document Attributes	A set of Job object and Document attributes as defined in section 6.2
Document Contents	Document content is optional and not included when a URL is provided to point to the content.

##### 845 846 847 5.4.1.2 Print Response

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

849

Job-Identifier	A URL Used for all other operations on this Job.
Job Status	Current-job-state
Printer State	Printer-state
Message	Optional message

Errors	Optional Error Information
--------	----------------------------

850  
851  
852

## 5.4.2 Cancel Job Operation

853       This operation allows a user to cancel one specific Print Job any  
854       time after the print job has been established on the Printer  
855       Object. Some pages may be printed before a job is terminated if  
856       printing has already started when the Cancel Job operation is  
857       received.

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

## 860   5.4.2.1 CancelJob Request

861       The following abstract data types are part of the Cancel Job  
862       Request.

863

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

864  
865

## 5.4.2.2 CancelJob Response

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

868

Job Status	Optional Job status information
------------	---------------------------------

Errors	Optional Error Information
--------	----------------------------

869

## 870   5.4.3 Get Attributes Operation

871       This operation allows an end-user to obtain information from the  
872       Print object concerning jobs, printers, and print queues, based on  
873       ISO 10175. The entity-body of the Get Attributes operation  
874       contains the set of attributes that the requester is interested  
875       in. However, the attribute values may be null and are ignored by  
876       the server. The attribute list is returned in the response with  
877       the appropriate attribute values filled in. If no attribute list  
878       is supplied, then all attributes defined for that object are  
879       returned.

## 880   5.4.3.1 GetAttributes Request

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

Selector	Job-Identifier (URL) or Printer URL
----------	----------------------------------------

Requested Attributes	A set of attributes in which the requestor is interested
-------------------------	-------------------------------------------------------------

883

## 5.4.3.2 GetAttributes Response

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

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

## 5.4.4 Get Jobs Operation

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

## 5.4.4.1 GetJobs

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

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

## 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. For each Job URL the following attributes are returned: job-total-octets and position-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 interspersed 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).

## 913 6.1 Attribute Syntaxes

914 The sections below reference the following syntax items:

string	arbitrary ASCII strings, no control characters, except <SPACE>.
stringPair	string ":" string
stringState	string state
name	arbitrary ASCII strings, no control characters, and no <SPACE> characters.
URL	Universal Resource Locator
dateTime	date and time in RFC 822 format
deltaTime	[hours ":" ] minutes
cardinal	0 .. n represented as ASCII digits
type1Enum	standard names, must revise the IPP standard to add a new name. No private names are allowed.
type2Enum	standard names, but an implementor can add new TBDs 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.
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.
type2EnumState	type2Enum state
type3EnumState	type3Enum state
state	type1Enum
Boolean	tokens: yes, y, true, or t and no, n, false, or f.
positiveInteger	1 .. n represented as ASCII digits
positiveIntegerCross	positiveInteger [ "x" positiveInteger ]
positiveIntegerCross State	positiveIntegerCross state
positiveIntegerRange	positiveInteger ":" positiveInteger
positiveIntegerUnits	positiveInteger units
positiveIntegerState	positiveInteger state
units	"ppm"   "ipm"   "spm"   "cps"   "lpm"
type3Locale	type3Country ":" type3Language ":" type3CodeSet
type3Country	type3Enum
type3Language	type3Enum
type3CodeSet	type3Enum
type2Format	name [ "/" version ]
version	name
type3LocaleState	type3Locale state

915

916 Also, the following conventions are used:

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

918     "1#" in front of a data syntax means one or more

919

920

## 921 6.2 Job Attributes

922     A job object contains a set of job attributes and one or more  
923     documents. A client shall create a job and send it to a server  
924     using the Print operation. A client may use a job template  
925     associated with the selected printer in order to initialize the  
926     job.

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

### 929 6.2.1 Job Informational Attributes (Set by a Client/End User)

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

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

#### 934 6.2.1.1 job-name (string)

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

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

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

### 941 6.2.2 Job Informational Attributes (Set by a Printer)

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

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

#### 946 6.2.2.1 job-identifier (url)

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

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

## 952 6.2.2.2 job-originator (name)

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

## 958 6.2.2.3 job-originating-host (name)

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

## 963 6.2.2.4 notification-address (name)

964 This address specifies the email address of the client. The client  
965 specifies this attribute in the operation-notification-address  
966 attribute which the Printer in turn uses to set this attribute.

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

971 Note: The only type of notification is email.

972 Issue: Can the email address be inferred with job-originator and  
973 the originating-host?

## 974 6.2.2.5 job-locale (type3Locale)

975 This attribute identifies the locale of the job. The Printer sets  
976 this attribute from the value of the operation-locale.

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

979 The type3EnumTrip consists of 3 colon separated type 3 enums. The  
980 first shall be the two-character country code from ISO 639. The  
981 second shall be the two-character language code from ISO 3166. The  
982 third is the code-set from the IANA Code Set Registry.

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

## 984 6.2.3 Job Status Attributes (Set by Printer)

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

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

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

## 993 6.2.3.1 current-job-state (typeName)

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

Unknown	The job state is not known, or is indeterminate.
held	The job is waiting to be released for scheduling for any number of reasons as specified by the value of the job's job-state-reasons attribute.
pending	The job's job-submission-complete attribute is TRUE since the server has received a print-request with the job-submission-complete parameter TRUE and the job is waiting to start processing on a printer.
processing	The server is processing the job, or has made the job ready for printing, but the output device is not yet printing it, either because the job hasn't reached the output device or because the job is queued in the output device or some other spooler, awaiting the output device to print it.

Or

The server has completed processing the job and the output device is currently printing the job on at least one printer. That is, 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)

paused	The job has been paused as a result of a PauseJob operation.
interrupted	The job was interrupted by the InterruptJob request for an intervening job, and shall resume processing automatically once the intervening job has completed.
terminating	The job has been cancelled by a CancelJob request or aborted by the server and is in the process of terminating. The job's job-state-reasons attribute contains the reasons that the job is being terminated.

retained        The job is being retained at the server as a result of the job's job-retention-period being non-zero. The job has (1) completed successfully or with warnings or errors, (2) been aborted while printing by the server, or (3) been cancelled by the CancelJob request before or during processing. The job's job-state-reasons attribute contains the reasons that the job has been retained. While in the retained state, all of the job's document data (and resources, if any) shall be retained by the server; thus a job in the retained state could be reprinted, using some means outside the scope of 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 cancelled by the CancelJob request,  
                  AND the job's:  
                  (1) job-retention-period was zero or has expired, or  
                  (2) job-discard-time has arrived.  
                  The job's job-state-reasons attribute contains the reason(s) that the job has been completed. While in the completed state, a job's document data (and resources if any) need not be retained by the server; thus a job in the completed state could not be reprinted. The length of time that a job may be in this state, before transitioning to unknown, is implementation-dependent. However, servers that implement the completed job-state shall retain, as a minimum, the following attributes for any job in the completed state: job-identifier, job-owner, job-name, current-job-state, printers-assigned, and job-state-reasons.

997

998        The IPP protocol supports all values for job states, but Printers  
 999        are need only support those states which are appropriate for the  
 1000        particular implementation.

1001      6.2.3.2 output-device-assigned (name)

1002        This attribute identifies the Output Device to which the Printer  
 1003        has assigned this job.

1004        If an Output Device implements a Printer, the Printer does not set  
 1005        this attribute.

1006        If a Print Server implements a Printer, the value shall be empty  
 1007        until the Printer assigns an Output Device to the job..

1008 The value of the job's printer-assigned attribute shall remain  
 1009 after the job has completed, so that end users can determine the  
 1010 Output Device on which the job was printed.

#### 1011 6.2.3.3 submission-time (dateTime)

1012 This attribute indicates the time at which this job was accepted  
 1013 by the Printer. If the Printer does not support the notion of  
 1014 time, the attribute is not stored as part of the job object.

#### 1015 6.2.3.4 job-message-from-administrator (string)

1016 This attribute provides a message from an operator, system  
 1017 administrator or "intelligent" process to indicate to the end user  
 1018 the reasons for modification or other management action taken on a  
 1019 job.

#### 1020 6.2.3.5 completion-time (dateTime)

1021 This attribute indicates the time at which this job completed.  
 1022 This time is useful for jobs which are retained after printing.  
 1023 If the Printer does not support the notion of time, the attribute  
 1024 is not stored as part of the Job object.

#### 1025 6.2.3.6 job-state-reasons (1#type2Enum)

1026 This attribute identifies the reason or reasons that the job is in  
 1027 the state that it is in (e.g., held, terminating, retained,  
 1028 completed, etc.). The printer shall indicate the particular  
 1029 reason(s) by setting the value of the job-state-reasons attribute.  
 1030 It is valid for the printer to set the value of the job-state-  
 1031 reasons attribute to the empty set.

1032 The following standard values are defined:

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

1033

1034

1035   6.2.3.7 impressions-completed (cardinal)

1036       This attribute contains the number of impressions that the Printer  
 1037       has completed printing. If the Printer cannot report this number,  
 1038       the Printer leaves this attribute unspecified.

1039   6.2.3.8 media-sheets-completed (cardinal)

1040       This attribute contains the number of media-sheets that the  
 1041       Printer has completed printing. If the Printer cannot report this  
 1042       number, the Printer leaves this attribute unspecified.

1043   6.2.4 Job Sheet Attributes (Set by Client/End User)

1044       The client shall specify these attributes to control the printing  
 1045       of job sheets.

1046       The client may also specify job sheet attributes in: Get-  
 1047       Attributes and Get-Jobs.

1048       job-sheets (type3Enum) This attribute determines what type of job-  
 1049       sheets the Printer shall print with the job.

1050       The standard values are: none, and default-sheet.

1051       The value "none" means that the Printer shall print no job sheets.  
 1052       The value "default-sheet" means that the Printer shall print the  
 1053       job sheets defined by an administrator. If the administrator's  
 1054       policy is not to support none, the Printer shall use the default-  
 1055       sheet value if the client supplies the "none" value.

1056   6.2.5 Notification Attributes (Set by a Client/End User)

1057       The client shall specify these attributes to indicate events that  
 1058       the client is interested in.

1059       The client may also specify notification attributes in: Get-  
 1060       Attributes and Get-Jobs.

1061   6.2.5.1 notification-events (#type1Enum)

1062       This attribute specifies the events about which the end user want  
 1063       to be notified.

1064       This attribute will support four events classes: none, job-  
 1065       completion, job-problems and printer-problems.

1066 If this attribute contains the event none, the Printer shall not  
1067 notify. This value is useful if an administrator has set up a  
1068 notification Printer default but the end user does not which  
1069 notification. If the none value and other values are supplied,  
1070 the Printer shall ignore the none value.

1071 This attribute will support only one delivery method, namely  
1072 email. The attribute notification-address specifies the email  
1073 address.

1074 If this attribute contains the event job-completion, the Printer  
1075 shall notify the client when the job containing this attribute  
1076 completes.

1077 If this attribute contains the event job-problem, the Printer  
1078 shall notify the client when the job containing this attribute has  
1079 a problem while the job is printing. Problems include: paper jam  
1080 and out-of-paper.

1081 If this attribute contains the event printer-problem, the Printer  
1082 shall notify the client when the job containing this attribute has  
1083 a problem while the job is printing or waiting to print. Problems  
1084 include: paper jam and out-of-paper.

#### 1085 6.2.6 Job Scheduling Attributes (Set by Client/End User)

1086 The client shall specify these attributes to provide the Printer  
1087 with information for the scheduling a print-job.

1088 The client may also specify these attributes in: Get-Attributes  
1089 and Get-Jobs.

##### 1090 6.2.6.1 job-priority (typeName)

1091 This attribute specifies a priority for scheduling the print-job.  
1092 Printers that employ a priority-based scheduling algorithm use  
1093 this attribute.

1094 There are three standard values: high, default, and low. Among  
1095 those jobs that are ready to print, a Printer shall print all such  
1096 jobs with a high priority before printing those with a default or  
1097 low priority, and a Printer shall print all such jobs with a  
1098 default priority before printing those with a low priority.

1099 If the client does not specify this attribute, the Printer assumes  
1100 that the end user places no constraints concerning priority on the  
1101 scheduling of the print-job, and it has a priority value of  
1102 default

1103 An operator can modify a job to have any priority. An end-user is  
1104 restricted by the value of the maximum-end-user-priority Printer  
1105 attribute.

##### 1106 6.2.6.2 job-print-after (dateTime)

1107 This attribute specifies the calendar date and time of day after  
1108 which the print-job shall become a candidate for printing.

1109 If the value of this attribute is in the future, the Printer shall  
1110 set the value of the job's current-job-state to held and add the  
1111 job-print-after-specified value to the job's job-state-reasons  
1112 attribute and shall not schedule the print-job for printing until  
1113 the specified date and time has passed. When the specified date  
1114 and time arrives, the Printer shall remove the job-print-after-  
1115 specified value from the job's job-state-reason attribute and, if  
1116 no other reasons remain, shall change the job's current-job-state  
1117 to pending so that the job becomes a candidate for being scheduled  
1118 to print.

1119 If this attribute is unspecified or the value is in the past, the  
1120 job shall be a candidate for scheduling immediately.

#### 1121 6.2.6.3 job-print-off-peak (type3Enum)

1122 This attribute specifies the off-peak period during which the  
1123 print-job shall become a candidate for printing.

1124 Standard values are: "evening", "night", "weekend", "second-  
1125 shift", "third-shift".

1126 If this attribute is specified, it contains a value with which an  
1127 administrator has associated allowable print times. An  
1128 administrator is encouraged to pick names that suggest the type of  
1129 off-peak period.

1130 If this attribute is unspecified, the job shall be a candidate for  
1131 scheduling immediately.

#### 1132 6.2.6.4 job-retention-period (deltaTime)

1133 The retention time is expressed in hours and minutes, e.g. 6:00 (6  
1134 hours), or 20 (20 minutes).

1135 This attribute specifies the minimum period of time following the  
1136 completion of job processing and printing that the server shall  
1137 keep job attributes and document data. The Printer may keep these  
1138 attributes and data longer than the value of the job-retention-  
1139 period attribute.

1140 Issue: There is some discussion about whether or not this should  
1141 be removed from the spec?

#### 1142 6.2.7 Job Production Attributes (Set by Client/End User)

1143 The client shall specify these attributes to affect the rendering,  
1144 production and finishing of the documents in the job. Similar  
1145 types of instructions may also be contained in the document to be  
1146 printed.

1147 If there is a conflict between the value of one of these  
1148 attributes, and a corresponding instruction in the document  
1149 (either implicit or explicit), the value of the attribute shall  
1150 take precedence over the document instruction.

1151 Job Production and Resource Attributes each address a similar set  
1152 of features but they have different uses.

1153 A job production attribute provides a client with a way to request  
 1154 some feature that is not embedded within the document data. After  
 1155 some program has merged the production attributes into the  
 1156 document data After the information from these attributes has been  
 1157 folded into the document data (possibly during a translation  
 1158 process of the document data), these attributes are no longer  
 1159 relevant and shall can be discarded from a job. Instead, the  
 1160 resource attributes specify the resources needed to print the job  
 1161 as modified by the job production attributes.

1162 Note: until companies that supply interpreters for PDL's, such as  
 1163 PostScript and PCL allow a way to specify overrides for internal  
 1164 job production instructions, a Printer may not be able to  
 1165 implement these attributes for some PDL's.

1166 A job resource attribute tells a Printer what features the job  
 1167 needs. A program that translates document data to a Printer's PDL,  
 1168 and/or merges production attributes into the document data should  
 1169 add job resource attributes to a job.

1170 For example, a job production attribute medium-select with the  
 1171 value of "letter" requests that a job be printed on letter paper,  
 1172 but gives no information about what resources the job needs. A  
 1173 resource production attribute media-used with the values of  
 1174 "letter" and "ledger" tell a Printer that the job needs letter and  
 1175 ledger paper, but gives no information about which pages use each  
 1176 medium.

1177

1178 The client may also specify document production-instruction  
 1179 attributes in: Get-Attributes and GetJobs.

#### 1180 6.2.7.1 medium-select (type2Enum)

1181 This attribute identifies the medium that the Printer shall use  
 1182 for all pages of the document regardless of what media are  
 1183 specified within the document.

1184 The values for medium include medium-names, medium-sizes, input-  
 1185 trays and electronic forms so that one attribute specifies the  
 1186 media.

1187 Standard values are defined (taken from ISO DPA and the Printer  
 1188 MIB)

1189

default	The default medium for the output device
iso-a4-white	Specifies the ISO A4 white medium
iso-a4-coloured	Specifies the ISO A4 coloured medium
iso-a4-transparent	Specifies the ISO A4 transparent medium
iso-a3-white	Specifies the ISO A3 white medium
iso-a3-coloured	Specifies the ISO A3 coloured medium
iso-a5-white	Specifies the ISO A5 white medium
iso-a5-coloured	Specifies the ISO A5 coloured medium

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

1190

1191 The following standard values are defined for North American  
 1192 media:

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

1193

1194 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

1195

1196 The following standard values are defined for the less commonly  
 1197 used media (white-only):

executive-white	Specifies the white executive medium
folio-white	Specifies the folio white medium
invoice-white	Specifies the white invoice medium
ledger-white	Specifies the white ledger medium
quarto-white	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-l0-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

1198

1199 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

1200

1201 6.2.7.2 number-up (positiveInteger)

1202 This attribute specifies the number of source page-images to  
1203 impose upon a single side of an instance of a selected medium.

1204 In general, only certain numeric values are valid for this  
1205 attribute, depending upon the Printer implementation to which the  
1206 print-request is directed. Typical supported values are 2 and 4.  
1207 If this attribute is unspecified or has a value of 1, then the  
1208 Printer does not apply any number-up transformation to the pages.

1209 This attribute primarily controls the translation, scaling and  
 1210 rotation of page images, but a site may choose to add  
 1211 embellishments, such as borders to each logical page. If  
 1212 embellishments are added, especially for the number-up = 1 case,  
 1213 these are controlled through some other mechanism or attribute.  
 1214 The user expects that if number-up is absent or equal to 1 then no  
 1215 other imposition embellishments are added via this attribute.

1216

#### 1217 6.2.7.3 finishing (type2Enum)

1218 This attribute identifies the finishing operation that the Printer  
 1219 should apply to each copy of the printed document. Examples  
 1220 include stapling, saddle-stitching, hole-drilling, binding with  
 1221 tape, etc.

1222 Standard values for this attribute are:

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.  
 none                    Perform no finishing. See 9.1.2

1223

1224    6.2.7.4 sides (type2Enum)

1225                    This attribute specifies whether the document should be printed in  
 1226                    one of three ways: 1-sided (simplex), 2-sided-long-binding-edge  
 1227                    (duplex), 2-sided-short-binding-edge (tumble).

1228                    The standard values are: 1-sided, 2-sided-long-edge, 2-sided-  
 1229                    short-edge.

1230    6.2.7.5 copies (positiveInteger)

1231                    This attribute specifies the number of copies of the job to be  
 1232                    printed. If this attribute is unspecified, its default value is 1  
 1233                    copy.

1234    6.2.7.6 printer-resolution-select (positiveIntegerCross)

1235                    This attribute specifies the resolution that the Printer should  
 1236                    use.

1237                    The syntax allows a single integer to specify the resolution or a  
 1238                    pair of integers to specify the resolution when the x and y  
 1239                    dimensions differ. When two integers are specified, the first is  
 1240                    in the x direction, ie., in the direction fo the shortest  
 1241                    dimension of the medium, so that the value is independent of  
 1242                    whether the printer feeds long edge or short edge first..

1243    6.2.7.7 print-quality (type2Enum)

1244                    This attribute specifies the print quality that the Printer should  
 1245                    use.

1246                    The standard values are:

1247                    draft            Lowest quality available on the printer  
 1248                    normal          Normal or intermediate quality on the printer  
 1249                    high             Highest quality available on the printer

1250

1251

1252    6.2.7.8 page-select (positiveIntegerRange)

1253                    This attribute specifies the pages in the document that the  
 1254                    Printer shall use. This attribute is unlikely to be useful for  
 1255                    jobs with more than one document or in Job Templates. If this  
 1256                    attribute is unspecified, then the Printer prints all pages in a  
 1257                    document.

1258    6.2.7.9 files-are-one-document (Boolean)

1259                    This attribute is relevant only if a job consists of two or more  
 1260                    documents. It controls finishing operations, job-sheet placement,  
 1261                    and the order of documents when the copies attribute exceeds 1.

1262 If the files for the job are a and b and this attribute is true,  
1263 then files a and b are treated as a single document for finishing  
1264 operations. Also, there will be no slip sheets between files a and  
1265 b. If more than one copy is made, the ordering must be a, b, a,  
1266 b, .... The attribute files-are-interleaved is ignored.

1267 If the files for the job are a and b and this attribute is false  
1268 or unspecified, then each file is treated as a single document for  
1269 finishing operations. Also, a client may specify that a slip sheet  
1270 be between files a and b. If more than one copy is made, and the  
1271 attribute files-are-interleaved false or unspecified, the ordering  
1272 is a, a, b, b, .... If more than one copy is made, and the  
1273 attribute files-are-interleaved true, the ordering is a, b, a, b,  
1274 ....

#### 1275 6.2.7.10 files-are-interleaved (Boolean)

1276 This attribute is used in conjunction with files-are-one-document  
1277 (q.v.).

1278

#### 1279 6.2.8 Attributes for Conversion of Text Files (Set by Client/End 1280 User)

1281 The client shall specify these attributes to control formatting  
1282 for text documents or HTML documents. If the client does not  
1283 specify any of these attributes, a Printer shall use its own  
1284 defaults.

1285 A client need not specify these attributes for other types of  
1286 documents, such as PostScript or PCL.

##### 1287 6.2.8.1 width (cardinalUnits)

1288 This attribute specifies the media width for the document in  
1289 characters.

##### 1290 6.2.8.2 length (cardinalUnits)

1291 This attribute specifies the media length for the document in  
1292 characters.

##### 1293 6.2.8.3 left-margin (cardinalUnits)

1294 This attribute specifies the left-margin for the document in  
1295 characters.

##### 1296 6.2.8.4 right-margin (cardinalUnits)

1297 This attribute specifies the right-margin for the document in  
1298 characters.

##### 1299 6.2.8.5 top-margin (cardinalUnits)

1300 This attribute specifies the top-margin for the document in lines.

## 1301 6.2.8.6 bottom-margin (cardinalUnits)

1302 This attribute specifies the bottom-margin for the document in  
1303 lines.

## 1304 6.2.8.7 repeated-tab-stops (cardinalUnits)

1305 This attribute specifies the tab stops for the document in  
1306 characters.

## 1307 6.2.8.8 header-text (string)

1308 This attribute specifies the header text for the document.

## 1309 6.2.8.9 footer-text (string)

1310 This attribute specifies the footer text for the document.

## 1311 6.2.8.10 font-size (cardinalUnits)

1312 This attribute specifies the font-size in points for text in the  
1313 document. The value of this attribute affects the size of the  
1314 other text attributes.

1315 If this attribute is omitted, the Printer shall assume a value of  
1316 10. A value of 10 with a fixed pitch font, shall produce 12  
1317 characters per inch in the horizontal direction and with 6 lines  
1318 per inch in the vertical direction.

## 1319 6.2.8.11 number-pages (Boolean)

1320 This attribute specifies that the pages should be numbered in the  
1321 document.

1322 default-font (string) This attribute specifies the font to use for  
1323 all text in the document.

## 1324 6.2.8.12 default-code-set (type3Enum)

1325 This attribute specifies the code-set in which the document is  
1326 encoded.

## 1327 6.2.8.13 content-orientation (type2Enum)

1328 This attribute specifies the orientation of the document.

1329 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.

1330

1331      6.2.9 Job Resource Attributes (Set by the program that produces or  
 1332      senses the PDL)

1333      A program described below shall add these attributes, which  
 1334      describe the resources needed to print the job.

1335      A Printer may use these attributes to validate and schedule the  
 1336      print-job without interpreting the contents of the document. This  
 1337      provides the opportunity for a Printer to support a broad set of  
 1338      document formats yet still support fast efficient scheduling and  
 1339      validation of each job.

1340

1341      The client/end user shall not specify these attributes. Instead,  
 1342      it is the duty of the program that translates the document to the  
 1343      printer's PDL (or analyzes it) to add these attributes and their  
 1344      values to the job. Such a program may execute at a number of  
 1345      different points in time:

1346      1. The program produces a final form document and stores it in  
 1347      a file before the end-user submits the print job.

1348      2. The program produces a final form document data stream when  
 1349      the end-user specifies "Print" to the application program  
 1350      (Windows GDI driver).

1351      3. The program translates a revisable or final form document  
 1352      into a PDL that the printer understands.

1353      If any of these attributes is unspecified, the Printer shall  
 1354      assume that the all resources required by the document of the type  
 1355      specified by the missing attributes are ready, ie., are available  
 1356      to the Printer and/or output device without human intervention.  
 1357      These attributes may be unspecified if the translation program

1358 fails to provides such values, or if no translation occurs (e.g.  
1359 the document is a PostScript document.

1360 Note: The Printer does not use these attributes during the actual  
1361 printing of a document.

1362 Note: these attributes allow more than one value wherever it is  
1363 possible for a job to specify more than one value of the  
1364 corresponding job attribute, possibly by embedded instructions.

1365 The client may specify these attributes in: Get-Attributes and  
1366 Get-Jobs.

1367 See the section on job production attributes for an explanation of  
1368 how the job resource attributes differ from the job production  
1369 attributes.

#### 1370 6.2.9.1 document-format-used (1#type2Format)

1371 This attribute identifies the document formats needed to print the  
1372 document(s) in this job.

1373 A format consists of two elements, a name and a version. The  
1374 latter element is optional.

1375 The syntax is for type2Format:

1376 name [ "/" version ]

1377 Examples include: PostScript, PostScript/2.0 and PCL/5e

1378 Note: The version component is optional.

#### 1379 6.2.9.2 fonts-used (1#string)

1380 This attribute identifies the font resources used in the  
1381 document(s) in the job.

#### 1382 6.2.9.3 code-sets-used (1#type3Enum)

1383 This attribute identifies the code-sets used in the document(s) in  
1384 the Job. This attribute is relevant only for files that are not in  
1385 ASCII, such as text files and possibly PCL files. PostScript files  
1386 are always ASCII. Normally there is at most 1 code-set.

1387 Standard values are defined in the section specifying the default-  
1388 code-set attribute.

#### 1389 6.2.9.4 media-used (1#type2Enum)

1390 This attribute identifies the media, media-sizes, input-trays or  
1391 electronic forms needed to print the document(s) in the job.

1392 Standard values for this attribute are defined in the section  
1393 specifying the medium-select attribute.

1394

## 1395 6.2.9.5 sides-used (type2Enum)

1396 This attribute specifies whether a job needs 1-sided, 2-sided-  
1397 long-binding-edge, or 2-sided-short-binding-edge printing.

1398 Standard values for this attribute are defined in the section  
1399 specifying the sides attribute.

## 1400 6.2.9.6 print-quality-used (type2Enum)

1401 This attribute specifies what print quality the job needs.

1402 Standard values for this attribute are defined in the section  
1403 specifying the print-quality attribute.

## 1404 6.2.9.7 finishing-used (type2Enum)

1405 This attribute specifies what finishing the job needs.

1406 Standard values for this attribute are defined in the section  
1407 specifying the finishing attribute.

## 1408 6.2.9.8 printer-resolution-used (positiveIntegerCrossState)

1409 This attribute specifies what resolution the job needs.

## 1410 6.2.9.9 total-job-octets (positiveInteger)

1411 This attribute specifies the total size of the job in octets. This  
1412 attribute is the first of three that a translation program can use  
1413 to specify the size of a job.

## 1414 6.2.9.10 job-impression-count (positiveInteger)

1415 This attribute specifies the total size of the job in impressions.

## 1416 6.2.9.11 job-media-sheet-count (positiveInteger)

1417 This attribute specifies the total size of the job in media-  
1418 sheets.

## 1419 6.2.9.12 job-intervening-jobs (positiveInteger)

1420 This attribute indicates the number of jobs that are "ahead" of  
1421 this job in the current scheduled order. For efficiency, it is  
1422 only necessary to calculate this value when an operation is  
1423 performed that requests this attribute.

## 1424 6.2.10 Number of Documents (Set by Client)

1425 This group contains a single attribute which specifies the number  
1426 of documents in the job.

1427 The client shall specify this attribute in Print and may specify  
1428 this attribute in: Get-Attributes and Get-Jobs.

## 1429 6.2.10.1 number-of-documents (positiveInteger)

1430 This attribute specifies the number of documents in the job. Each  
1431 document shall contain its own set of document content attributes  
1432 described below.

## 1433 6.2.11 Document Data (Set by a Client/End User)

1434 This group of attributes describes the document data for the job.  
1435 These attributes also include the document data or reference it.

1436 All job attributes in other sections of this document occur only  
1437 once per job and apply to all documents in a job.

1438 The client may specify document-data attributes in Print. The  
1439 client must specify either the document-URL or document-content in  
1440 Print.

1441 Except for document-content, the client may specify document-data  
1442 attributes in: Get-Attributes, and Get-Jobs.

## 1443 6.2.11.1 document-format (type2Format)

1444 This attribute identifies the document format of this document.

1445 If the client does not specify this attribute, then the Printer  
1446 shall attempt to determine the format in order to decide if the  
1447 document data needs to be translated. The version component is  
1448 optional.

## 1449 6.2.11.2 document-name (name)

1450 This attribute contains the name of the document used by the  
1451 client to initially identify the document.

## 1452 6.2.11.3 document-URL (name)

1453 This attribute contains the URL of the document if the client  
1454 specified the document with a URL.

1455 If this attribute is specified, then document-content shall be  
1456 unspecified.

## 1457 6.2.11.4 document-content (octetString)

1458 This attribute contains the actual contents of the document.

1459 If this attribute is specified, then document-URL shall be  
1460 unspecified.

1461 This attribute shall be used during the transmission of the Print  
1462 operation over a network. A Printer shall save the document data  
1463 to a file and reference it with the document-URL or document-path  
1464 attribute. A Get-Attribute or Get-Jobs operation shall always find  
1465 that this attribute is unspecified.

## 1466 6.3 Operation Attributes (Set by Client)

1467 NOTE: These attributes have just been introduced and they are not  
1468 as stable as the attributes in the other sections. Some work is  
1469 still needed to show the relationship between these attributes,  
1470 job attributes, printer attributes, and authentication and  
1471 authorization.

1472 The client shall set these attributes and associate them with an  
1473 operation rather than an object.

1474 It is intended that a client program rather than an end-user has  
1475 control over the setting of these values so that they cannot be  
1476 easily forged.

## 1477 6.3.1 operation-locale (type3Locale)

1478 This attribute identifies the locale of the client. The Printer  
1479 uses this attribute to determine the locale of messages in the  
1480 result of the operation or in errors returned by the operation.

1481 The standard values are defined in the section on the job-locale  
1482 attribute.

1483 If an operation does not specify this attribute, the Printer shall  
1484 assume that the operation has the same locale as the Printer.

## 1485 6.3.2 operation-notification-address (url)

1486 This attribute identifies the both the address and mechanism for  
1487 delivery of events. If the URL has a "mailto:" scheme, then email  
1488 is used and the rest of the URL is used as the email address. If  
1489 the URL has a "http:" scheme, then an HTTP APPEND method is used  
1490 to add HTML formatted events to the end of a specified HTML file.

## 1491 6.3.3 operation-user-name (name)

1492 This attribute identifies the most authenticated end user name  
1493 that the client can supply. This name identifies the end user  
1494 performing the operation.

1495 This value shall be set by the system rather than the end-user in  
1496 order to minimize the chance of forgery.

## 1497 6.3.4 operation-host-name (name)

1498 This attribute identifies the most authenticated host name that  
1499 the client can supply. This name identifies the host from which  
1500 the operation comes.

1501 This value shall be set by the system rather than the end-user in  
1502 order to minimize the chance of forgery.

## 1503 6.4 Printer Attributes (Set by the Administrator)

1504 A printer object may be realized in either a Print Server or  
1505 Output Device. Note: How these attribute are set by an  
1506 Administrator is outside the scope of this specification.

1507 A Printer Object in an Output Device contains a set of printer  
 1508 object attributes that represent an Output Device capable of  
 1509 rendering a document in visible form. Examples include electronic  
 1510 and electro-mechanical printers such as laser printers, ink-jet  
 1511 printers, and various kinds of impact printers, but may include  
 1512 other types of output devices such as microfiche imagers and  
 1513 plotters as well.

1514 A Printer Object in a Print Server that supplies queuing,  
 1515 spooling, and scheduling for an Output device that does not queue  
 1516 or spool.

1517 A Printer Object in a Print Server contains a set of printer  
 1518 object attributes that are the union of the Printer objects in the  
 1519 downstream Output Devices. This object extends the capabilities  
 1520 of an Output Device. For example, an administrator might define a  
 1521 single Print Server to represent all of the Output Devices of the  
 1522 same type and capability in a single location, associated with a  
 1523 particular server. A end user would normally send a print-job to  
 1524 a Print Server , and allow the Print Server to assign the job to a  
 1525 particular Output Device based on the relative load and  
 1526 availability of the printers under its control, thus providing a  
 1527 load balancing service. However, nothing precludes an  
 1528 administrator from configuring a print system so that a end user  
 1529 can send a print-job directly to an Output Device .

1530 A Print Server, in the most common case, controls exactly one  
 1531 downstream Output Device. The Print Server's Printer object has  
 1532 attributes whose values are the same as those of the Printer  
 1533 object in the downstream Output Device.

1534 The attributes defined in this section provide information about  
 1535 a particular Printer.

#### 1536 6.4.1 printer-name (name)

1537 This attribute uniquely identifies the printer on its host.

#### 1538 6.4.2 printer-location (string)

1539 This attribute identifies the location of this printer.

#### 1540 6.4.3 printer-model (string)

1541 This attribute identifies the make and model of the printer.

#### 1542 6.4.4 printer-types (type2Enum)

1543 This attribute identifies the marking technology of the printer.

1544 The standard value for this attribute are the descriptive names  
 1545 specified by ISO DPA which have corresponding enum symbolic and  
 1546 numeric values assigned by the Printer MIB (RFC 1759).. These  
 1547 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

1548

## 1549 6.4.5 printer-state (type1Enum)

1550 This attribute identifies the current state of the printer. The  
 1551 protocol support all values for printer states, however a Printer  
 1552 shall only generate the printer states which are appropriate for  
 1553 the particular implementation.

1554 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 part of ISO/IEC 10175.

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

1555

1556 6.4.6 printer-state-message (string)

1557 This attributes specifies a message that gives further information  
 1558 about the current printer state. .

1559 6.4.7 message (string)

1560 This attribute provides a message from an operator, system  
 1561 administrator or "intelligent" process to indicate to the end user  
 1562 information or status of the printer, such as why it is  
 1563 unavailable or when it is expected to be available. .

1564 6.4.8 locale (type3Locale)

1565 This attribute specifies the locale that the Printer operates in.

1566 The standard values are defined in the section on the job-locale  
 1567 attribute.

## 1568 6.4.9 notification-events (#type2Enum)

1569 This attribute specifies the events on whose occurrence the  
1570 Printer should notify those addresses specified by the  
1571 notification-addresses attribute.

1572 If the attribute is unspecified or empty, the Printer does not  
1573 perform notification, though the Printer still checks the job's  
1574 notification-events attribute.

1575 In this attribute, job-problem and printer-problem have the same  
1576 meaning.

1577 The standard values are defined in the section on the job's  
1578 notification-events attribute.

## 1579 6.4.10 notification-addresses (#name)

1580 This attribute specifies the email addresses to which the Printer  
1581 should send messages when events specified by the notification-  
1582 events attribute occur.

1583 If the attribute is unspecified or empty, the Printer does not  
1584 perform notification, though the Printer still checks the job's  
1585 notification-events attribute.

## 1586 6.4.11 end-user-acl (#name)

1587 This attribute specifies the end users who are allowed to print on  
1588 the Printer.

1589 If the attribute is unspecified or empty, the Printer allows  
1590 anyone to print.

## 1591 6.4.12 maximum-printer-speed (positiveIntegerUnits)

1592 This attribute indicates the maximum printer speed of the Printer.  
1593 A job cannot control a Printer's speed, but a Printer Browser can  
1594 use printer speed as a criteria.

1595 The standard units are a type2Enum and are: ppm, ipm, spm, lpm,  
1596 cps.

## 1597 6.4.13 fonts-substitutions (#stringPair)

1598 This attribute specifies an appropriate substitute for a font that  
1599 is advertised as supported in the fonts-supported attribute, even  
1600 though the Printer doesn't actually have the font available.

1601 This attribute consists of a set of font pairs: a font name and  
1602 the font to use instead.

## 1603 6.4.14 fonts-supported (1#stringState)

1604 This attribute identifies the font resources supported by this  
1605 printer and indicates the state of readiness for each font.

1606       The standard names are defined in the section on default-font.

1607       Each item in the list contains the pair consisting of a font name  
1608       and a state indicating the font's readiness state.

1609   6.4.15 media-supported (1#nameState)

1610       This attribute identifies the media, media-sizes, input trays, and  
1611       electronic forms supported by this printer, and indicates the  
1612       state of readiness for each medium resource.

1613       There may be just two states: ready and needs-installing, or there  
1614       may be a third state: needs-purchasing.

1615       The standard names are defined in the section on the section on  
1616       the medium-select.

1617   6.4.16 document-formats-supported (1#type2FormatState)

1618       This attribute identifies the document-formats, including the  
1619       document-format-versions, supported by the Printer. This set  
1620       includes both the formats that are native to the Printer and  
1621       those formats that the Printer can translate to one that is  
1622       native to the Printer. From the client's point of view, this set  
1623       contains all formats in which documents can be submitted to this  
1624       Printer.

1625       Proprietary document format identifiers, and versions are assigned  
1626       by the owners of those formats.

1627       The state of readiness for each format is also included, though  
1628       all formats should normally always be ready.

1629   6.4.17 numbers-up-supported (1#positiveIntegerState)

1630       This attribute identifies the number-up values supported by this  
1631       printer..

1632       The state of readiness for each number-up value is also included,  
1633       though all number-up conversions should always be ready.

1634   6.4.18 finishings-supported (#type2EnumState)

1635       This attribute identifies the finishing operations supported by  
1636       this Printer and states of readiness for each finishing.

1637       The standard finishing objects are defined in the section on the  
1638       finishing Job attribute.

1639   6.4.19 sides-supported (1#type2EnumState)

1640       This attribute indicates the values of the sides attribute  
1641       supported by this printer and the states of readiness of each  
1642       value.

1643       The standard values are defined in the section on the sides  
1644       attribute.

## 1645 6.4.20 print-qualities-supported (1#type2EnumState)

1646 This attribute indicates the values of the printer-quality  
1647 attribute supported by this printer and the states of readiness  
1648 for each print-quality value.

1649 The standard values are defined in the printer-quality attribute.

## 1650 6.4.21 printer-resolutions-supported (1#positiveIntegerCrossState)

1651 This attribute indicates the values of the printer-resolution-  
1652 select attribute supported by this printer and their states of  
1653 readiness.

1654 The state of readiness for each printer resolution is also  
1655 included, though normally all printer-resolutions should always be  
1656 ready.

1657 The syntax is discussed in the section on the printer-resolution-  
1658 select attribute.

## 1659 6.4.22 code-sets-supported (1#type3EnumState)

1660 This attribute indicates the values of the default-code-set  
1661 attribute supported by this printer and the states of readiness  
1662 for each code-set.

1663 The standard values are defined in the default-code-set attribute.

## 1664 6.4.23 off-peak-times-supported (#type3EnumState)

1665 This attribute indicates the values of the job-print-off-peak  
1666 attribute supported by this printer and the states of readiness  
1667 for each value.

1668 If this attribute is unspecified, then the Printer has no off-peak  
1669 periods.

1670 The standard values are defined in the section on the job-print-  
1671 off-peak attribute.

1672 Note: this document does not define how an administrator  
1673 associates the off-peak names with actual time periods.

## 1674 6.4.24 events-supported (#type2EnumState)

1675 This attribute indicates the values of the job and printer  
1676 notification-events attribute supported by this Printer and the  
1677 states of readiness for each value.

1678 If this attribute is unspecified, then the Printer does not  
1679 support notification.

1680 The standard values are defined in the section on the  
1681 notification-events attribute.

## 1682 6.4.25 locales-supported (1#type3LocaleState)

1683 This attribute indicates the values of the job-locale attribute  
1684 supported by this Printer and the states of readiness for each  
1685 value.

1686 The standard values are defined in the section on the job-locale  
1687 attribute.

## 1688 6.4.26 job-sheets-supported (#type3EnumState)

1689 This attribute identifies the job-sheet values supported by this  
1690 printer, and the state of readiness for each job-sheet.

1691 To allow no job sheets, the system administrator shall include the  
1692 value none as a value for this attribute. The client specifies  
1693 that there are no job sheets by using the value none as the value  
1694 of the job-sheets attribute.

1695 If the job-sheets attribute is not specified or contains a value  
1696 which the Printer does not support, then the server shall select  
1697 from among the values of this attribute. The server shall not  
1698 select the value none unless it is the only value specified for  
1699 the job-sheets-supported attribute.

1700 NOTE - It is preferable for the server to produce some job  
1701 jobsheet, even if not the desired one, rather than produce none at  
1702 all or reject the job.

## 1703 1704 6.4.27 maximum-copies (positiveInteger)

1705 This attribute indicates the maximum number of copies of a  
1706 document that can be rendered by this printer in a single print-  
1707 job.

1708 If the attribute is unspecified or has a value of 0, there is no  
1709 limit on the maximum number of copies for this Printer.

## 1710 6.4.28 maximum-job-octets (positiveInteger)

1711 This attribute indicates that the Printer shall accept a job only  
1712 if its size in octets is less than the value specified by this  
1713 attribute.

1714 If the attribute is unspecified or has a value of 0, there is no  
1715 limit on the size of a job in octets.

## 1716 6.4.29 maximum-impressions (positiveInteger)

1717 This attribute indicates that the Printer shall accept a job only  
1718 if its size in impression is less than the value specified by this  
1719 attribute.

1720 If the attribute is unspecified or has a value of 0, there is no  
1721 limit on the size of a job in impressions.

## 1722 6.4.30 maximum-media-sheets (positiveInteger)

1723 This attribute indicates that the Printer shall accept a job only  
1724 if its size in media-sheets is less than the value specified by  
1725 this attribute.

1726 If the attribute is unspecified or has a value of 0, there is no  
1727 limit on the size of a job in media-sheets.

## 1728 6.4.31 maximum-job-retention-period (deltaTime)

1729 This attribute indicates that when the Printer accepts a job, the  
1730 retention period must not exceed the value of this attribute.  
1731 Otherwise, the Printer sets the job's retention-period to the  
1732 value of this attribute.

1733 If this attribute is unspecified, then the Printer places no limit  
1734 on the retention time.

## 1735 6.4.32 maximum-end-user-priority (type1Enum)

1736 This attribute indicates that when the Printer accepts a job, the  
1737 job-priority must not exceed the value of this attribute.  
1738 Otherwise, the Printer sets the job's job-priority to the value of  
1739 this attribute.

1740 If this attribute is unspecified, then the Printer places no limit  
1741 on the job-priority time.

1742 The standard values are defined in the section on the job-priority  
1743 attribute.

## 1744 6.4.33 queued-job-count (positiveInteger)

1745 This attribute contains a count of the number of jobs that are  
1746 either pending and/or processing.

## 1747 6.4.34 scheduling-algorithm (type3Enum)

1748 This attribute indicates the current scheduling algorithm for this  
1749 Printer: "none", "shortest-job-first", "time-received", etc.

## 1750 6.5 Job Templates

1751 The attributes for a Job Template can be any of the Job object  
1752 attributes defined in the sections:

1753 Job Sheet Attributes  
1754 Notification Attributes  
1755 Job Scheduling Attributes  
1756 (except job-print-after)  
1757 Job Production Attributes  
1758 (except page-select)  
1759 Attributes for Conversion of Text Files  
1760

## 1761 6.6 Conformance

1762 A conforming implementation shall implement all operations,  
1763 objects and attributes defined in this document. IPP is explicitly  
1764 designed to be extensible. This means that in addition to the  
1765 attributes defined in this specification, specific implementation  
1766 instances may support not only the basic protocol as defined in  
1767 this specification, but might add vendor specific extensions.

1768 Also, for the core set of attributes listed in this specification,  
1769 it is not required that a conforming server support all (standard)  
1770 values of all supported attributes. For example, it is not  
1771 required that a printer implement all finishing methods indicated  
1772 by the standard values.

1773 The explicit requirement of the term "supported", with respect to  
1774 one of the attributes that deal with printer functions or  
1775 resources, is that the server shall recognize the attribute and  
1776 those values that are supported, and shall be able to respond to a  
1777 query about which values that printer does, in fact, support.

1778 Additional attributes can be proposed to be registered by going  
1779 through the type 2 enum process which will register their  
1780 specification after approval with IANA.

## 1781 7. Security Considerations

1782 This protocol does not identify any new authentication mechanisms.  
1783 The authentication mechanisms built into HTTP (such as SSL and  
1784 SHTTP) are recommended.

1785 This protocol does define a simple authorization mechanism by  
1786 introducing the "end-user-acl" attribute as part of the Printer  
1787 object. This ACL attribute is a multi-valued list of all of the  
1788 authenticated names of end-users. This protocol does not specify  
1789 what the domain is for names in this ACL attribute.

1790 Issue: Will it always be possible for a Printer to obtain a  
1791 meaningful authenticated name that the Printer can match against  
1792 the end-user-acl, or will some other mechanism be necessary, such  
1793 as a password?

## 1794 8. References

1795 [1] Smith, R., Wright, F., Hastings, T., Zilles, S., and  
1796 Gyllenskog, J., "Printer MIB", RFC 1759, March 1995.  
1797

1798 [2] Berners-Lee, T., Fielding, R., and Nielsen, H., "Hypertext  
1799 Transfer Protocol - HTTP/1.0", RFC 1945, August 1995.  
1800

1801 [3] Crocker, D., "Standard for the Format of ARPA Internet Text  
1802 Messages", RFC 822, August 1982.  
1803

1804 [4] Postel, J., "Instructions to RFC Authors", RFC 1543, October  
1805 1993.  
1806

1807 [5] ISO/IEC 10175 Document Printing Application (DPA), Final,  
1808 June 1996.

- [6] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August 1995.
- [7] Kirk, M. (editor), POSIX System Administration - Part 4: Printing Interfaces, POSIX 1387.4 D8, 1994.
- [8] Borenstein, N., and Freed, N., "MIME (Multi-purpose Internet Mail Extensions) Part One: Mechanism for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, September, 1993.
- [9] Braden, S., "Requirements for Internet Hosts - Application and Support", FRC 1123, October, 1989,
- [10] McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.

## 9. Author's Address

Scott A. Isaacson  
Novell, Inc.  
122 E 1700 S  
Provo, UT 84606  
  
Phone: 801-861-7366  
Fax: 801-861-4025  
EMail: scott\_isaacson@novell.com

Tom Hastings  
Xerox Corporation  
701 S. Aviation Blvd.  
El Segundo, CA 90245  
  
Phone: 310-333-6413  
Fax: 310-333-5514  
EMail: hastings@cp10.es.xerox.com

Robert Herriot  
Sun Microsystems Inc.  
2550 Garcia Ave., MPK-17  
Mountain View, CA 94043  
  
Phone: 415-786-8995  
Fax: 415-786-7077  
Email: robert.herriot@eng.sun.com

Roger deBry  
HUC/003G  
IBM Corporation  
P.O. Box 1900  
Boulder, CO 80301-9191  
  
Phone: (303) 924-4080  
Fax: (303) 924-9889  
Email: debry@vnet.ibm.com

Other Contributors

1867        Devon Taylor, Novell, Inc.  
1868        Mike MacKay, Novell, Inc.  
1869        Peter Zehler, Xerox, Corp.  
1870        Keith Carter, IBM Corporation  
1871        Carl-Uno Manros, Xerox, Corp.  
1872        <add the list of IPP attendees and participants>  
1873

1874

## 1875 10. Appendix A: Sample IPP Operations

1876 The following examples illustrate typical flows using the IPP  
 1877 protocol. In these examples, the IPP Printer object named  
 1878 "printer-1" is located at the node identified by the DNS name  
 1879 "some.domain.com". AJjob Template has been defined for printer-1  
 1880 which establishes the print defaults.

1881 For brevity in the following flows, none of the HTTP headers are  
 1882 shown. CRLF sequences are not shown.

## 1883 10.1 Querying the printer

1884 Client some.domain.com

1885 ----->

1886 Post http://some.domain.com/printer-1 http/1.0

1887 GetAttributes IPP/1.0

1888 Printer-state :

1889 Sides-supported :

1890 Media-supported :

1891 Document-formats-supported :

1892

1893 <-----

1894

1895 http/1.0 201 "Created" (a response)

1896 IPP/1.0 xxx "attribute list returned"

1897 Printer-state : running

1898 Sides-supported : 1-sided

1899 Media-supported : iso-a4-white, iso-b4-white

1900 Document-formats-supported : Postscript/2.0

1901

1902

## 1903 10.2 Print Operation - with print data included

1904 Client some.domain.com

1905 ----->

1906 Post http://some.domain.com/printer-1 http/1.0

1907 Print IPP/1.0

1908 Print-Job-Object Header

1909 Job-name : My Job

1910 Medium : iso-a4-white

1911 Notification-events : Job-completion

1912 Notification-address : joe@pc.domain.com

1913 Document Header

1914 Document-name : Letter to Mom

1915 Document-Content Header (content type = Postscript/2.0)

1916 Document in Postscript level 2 format

1917

1918

1919 <-----

1920 http/1.0 200 "accepted"

```

1921         IPP/1.0 xxx "print job accepted and queued"
1922         Job-Identifier : some.domain.com/printer-1/0037
1923         Current-job-state : pending
1924         Printer-state : running
1925
1926 10.3 Print Operation - with no data included

1927 Client                                     some.domain.com
1928
1929 ----->
1930
1931 Post http://some.domain.com/printer-1 http/1.0
1932   Print IPP/1.0
1933   Print-Job-Object Header
1934     Job-name : My Job
1935     Medium : iso-a4-white
1936     Notification-events : Job-completion
1937     Notification-address : joe@some.domain.com
1938   Document Header
1939     Document-name : Letter to Mom
1940     Document-URL : joe@pc.domain.com/Docs/To-mom.ps
1941
1942 <-----

1943 http/1.0 200 "accepted"
1944   IPP/1.0 xxx "print job accepted and queued"
1945   Job-Identifier : some.domain.com/printer-1/0037
1946   Current-job-state : pending
1947   Printer-state : running
1948
1949 10.4 Querying the state of the job

1950 In this example, no attributes are specified, so all job
1951 attributes are returned.

1952 Client                                     some.domain.com
1953
1954 ----->
1955
1956 Post http://some.domain.com/printer-1/0037 http/1.0
1957   GetAttributes IPP/1.0
1958
1959 <-----

1958 http/1.0 201 "Created" (a response)
1959   IPP/1.0 xxx "attribute list returned"
1960   Job-Name : My Job
1961   Job-Originator : Joe@some.domain.com
1962   Job-originating-host : pc.domain.com
1963   Notification-address : joe@pc.domain.com
1964   Job-locale : xx:xx:xx
1965   Current-job-status : printing
1966   Printer-assigned : printer-1
1967   Submission-time : 1214
1968   Media-sheets-completed : 2
1969
1970 10.5 Canceling a Job

1971 Client                                     some.domain.com

```

```
1972      ----- >
1973      Post: http://some.domain.com/printer-1/0037
1974          CancelJob IPP/1.0
1975
1976      <-----
1977
1978      http/1.0 200 "okay"
1979      Current-job-state : terminating
1980
1981 10.6 Listing jobs on a Printer
1982
1983      List jobs on printer-1, only return job sizes. Jobs are returned
1984      in the order they are scheduled for printing. A Job-identifier
1985      attribute preceeds the attributes returned for each job to delimit
1986      job boundaries.
1986
1987      Client                                some.domain.com
1988
1989      ----- >
1989      Post http/1.0 some.domain.com/printer-1
1990          GetJobs IPP/1.0
1991          total-job-octets :
1992
1993      <-----
1994
1995      http/1.0 201 "Created" (a response)
1996          IPP/1.0 xxx "created an attribute list"
1997          Job-identifier : 0033
1998          total-job-octets : 4567
1999          Job-identifier : 0034
2000          total-job-octets : 12345
2001          Job-identifier : 0035
2002          total-job-octets : 12356
2003
```