

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 .....	55
2. Distributed Printing .....	55
2.1 Generic Print System Components .....	55
2.2 IPP Components .....	66
3. IPP Objects .....	66
3.1 Printer .....	77
3.2 Job .....	99
3.3 Job Template .....	99
3.4 Object Relationships .....	99
3.5 Object Identity .....	109
4. Naming .....	1010
4.1 Directory Services .....	1110
4.2 Directory Entry Schema .....	1111
4.2.1 Name .....	1211
4.2.2 Description .....	1211
4.2.3 Location .....	1211
4.2.4 Print Quality .....	1211
4.2.5 Cost .....	1211
4.2.6 This indicates a somewhat subjective evaluation of the overall cost of printing at this printer: "high", "medium", or "low".Resolution .....	1211
4.2.7 Color Supported .....	1212
4.2.8 Fonts Supported .....	1212
4.2.9 Maximum Speed .....	1312
4.2.10 Device Id .....	1312
4.2.11 Model .....	1312
4.2.12 Manufacturer .....	1312
4.2.13 Type .....	1312
4.2.14 Document Formats s Supported .....	1413
4.2.15 Sides Supported .....	1413
4.2.16 Finishings Supported .....	1413
5. IPP Operations .....	1513
5.1 IPP Operations Using HTTP .....	2213
5.1.1 HTTP Overview .....	2213
5.1.2 IPP Operation Encoding .....	2214
5.1.2.1 HTTP Request-Header Fields .....	2315
5.1.2.2 Print .....	2415
5.1.2.3 Cancel Job .....	2415
5.1.2.4 Get Attributes .....	2416
5.1.2.5 Get Jobs .....	2416
5.1.3 The Print-Job .....	2416
5.1.3.1 Print Job Object Header .....	2516
5.1.3.2 Document Header .....	2516
5.1.3.3 Document-Content Header .....	2517
5.1.3.4 Job Attributes .....	2517
5.1.3.5 Document Attributes .....	2617
5.2 Print Operation .....	2617

110	5.2.1 Print Request .....	<u>2618</u>
111	5.2.2 Print Response .....	<u>2618</u>
112	5.3 Cancel Job Operation .....	<u>2718</u>
113	5.3.1 Cancel Job Request .....	<u>2718</u>
114	5.3.2 Cancel Job Response .....	<u>2719</u>
115	5.4 Get Attributes Operations .....	<u>2719</u>
116	5.4.1 Get Attributes Request .....	<u>2719</u>
117	5.4.2 Get Attributes Response .....	<u>2719</u>
118	5.5 Get Jobs Operation .....	<u>2819</u>
119	5.5.1 Get Jobs Request .....	<u>2819</u>
120	5.5.2 Get Jobs Response .....	<u>2819</u>
121	6. Object Attributes .....	<u>2820</u>
122	6.1 Attribute Syntaxes .....	<u>2820</u>
123	6.2 Job Attributes .....	<u>2921</u>
124	6.2.1 Job Informational Attributes (Set by a Client/End User) .....	<u>2921</u>
125	6.2.1.1 job-name (string) .....	<u>3021</u>
126	6.2.2 Job Informational Attributes (Set by a Printer) .....	<u>3021</u>
127	6.2.2.1 job-identifier (url) .....	<u>3021</u>
128	6.2.2.2 job-originator (name) .....	<u>3022</u>
129	6.2.2.3 job-originating-host (name) .....	<u>3022</u>
130	6.2.2.4 notification-address (name) .....	<u>3022</u>
131	6.2.2.5 job-locale (type3Locale) .....	<u>3122</u>
132	6.2.3 Printer Selection Attributes (Set by Client/End User) ..	<u>3122</u>
133	6.2.3.1 printer-name-requested (URL) .....	<u>3123</u>
134	6.2.4 Job Status Attributes (Set by Printer) .....	<u>3123</u>
135	6.2.4.1 current-job-state (type1Enum) .....	<u>3123</u>
136	6.2.4.2 printer-assigned (name) .....	<u>3325</u>
137	6.2.4.3 submission-time (dateTime) .....	<u>3425</u>
138	6.2.4.4 job-message-from-administrator (string) .....	<u>3426</u>
139	6.2.4.5 completion-time (dateTime) .....	<u>3426</u>
140	6.2.4.6 job-state-reasons (1#type2Enum) .....	<u>3426</u>
141	6.2.4.7 impressions-completed (cardinal) .....	<u>3527</u>
142	6.2.4.8 media-sheets-completed (cardinal) .....	<u>3527</u>
143	6.2.5 Job Sheet Attributes (Set by Client/End User) .....	<u>3527</u>
144	6.2.6 Notification Attributes (Set by a Client/End User) ....	<u>3527</u>
145	6.2.6.1 notification-events (#type1Enum) .....	<u>3627</u>
146	6.2.7 Job Scheduling Attributes (Set by Client/End User) ....	<u>3628</u>
147	6.2.7.1 job-priority (type1Enum) .....	<u>3628</u>
148	6.2.7.2 job-print-after (dateTime) .....	<u>3728</u>
149	6.2.7.3 job-print-off-peak (type3Enum) .....	<u>3729</u>
150	6.2.7.4 job-retention-period (deltaTime) .....	<u>3729</u>
151	6.2.8 Job Production Attributes (Set by Client/End User) ....	<u>3829</u>
152	6.2.8.1 medium-select (type2Enum) .....	<u>3830</u>
153	6.2.8.2 number-up (positiveInteger) .....	<u>4132</u>
154	6.2.8.3 finishing (type2Enum) .....	<u>4133</u>
155	6.2.8.4 sides (type2Enum) .....	<u>4234</u>
156	6.2.8.5 copies (positiveInteger) .....	<u>4234</u>
157	6.2.8.6 printer-resolution-select (positiveIntegerCross) ...	<u>4234</u>
158	6.2.8.7 print-quality (type2Enum) .....	<u>4334</u>
159	6.2.8.8 page-select (positiveIntegerRange) .....	<u>4334</u>
160	6.2.8.9 files-are-one-document (Boolean) .....	<u>4334</u>
161	6.2.8.10 files-are-interleaved (Boolean) .....	<u>4335</u>
162	6.2.9 Attributes for Conversion of Text Files (Set by Client/End	
163	User) .....	<u>4335</u>
164	6.2.9.1 width (cardinalUnits) .....	<u>4435</u>
165	6.2.9.2 length (cardinalUnits) .....	<u>4435</u>
166	6.2.9.3 left-margin (cardinalUnits) .....	<u>4435</u>
167	6.2.9.4 right-margin (cardinalUnits) .....	<u>4435</u>
168	6.2.9.5 top-margin (cardinalUnits) .....	<u>4435</u>
169	6.2.9.6 bottom-margin (cardinalUnits) .....	<u>4436</u>

170	6.2.9.7 repeated-tab-stops (cardinalUnits)	4436
171	6.2.9.8 header-text (string)	4436
172	6.2.9.9 footer-text (string)	4436
173	6.2.9.10 font-size (cardinalUnits)	4436
174	6.2.9.11 number-pages (Boolean)	4536
175	6.2.9.12 default-code-set (type3Enum)	4536
176	6.2.9.13 content-orientation (type2Enum)	4536
177	6.2.10 Job Resource Attributes (Set by the program that produces	
178	or senses the PDL)	4537
179	6.2.10.1 document-format-used (1#type2Format)	4638
180	6.2.10.2 fonts-used (1#string)	4738
181	6.2.10.3 code-sets-used (1#type3Enum)	4738
182	6.2.10.4 media-used (1#type2Enum)	4738
183	6.2.10.5 sides-used (type2Enum)	4739
184	6.2.10.6 print-quality-used (type2Enum)	4739
185	6.2.10.7 finishing-used (type2Enum)	4739
186	6.2.10.8 printer-resolution-used (positiveIntegerCrossState)	4739
187	6.2.10.9 total-job-octets (positiveInteger)	4739
188	6.2.10.10 job-impression-count (positiveInteger)	4839
189	6.2.10.11 job-media-sheet-count (positiveInteger)	4839
190	6.2.11 Number of Documents (Set by Client)	4839
191	6.2.11.1 number-of-documents (positiveInteger)	4839
192	6.2.12 Document Data (Set by a Client/End User)	4840
193	6.2.12.1 document-format (type2Format)	4840
194	6.2.12.2 document-name (name)	4940
195	6.2.12.3 document-URL (name)	4940
196	6.2.12.4 document-content (octetString)	4940
197	6.3 Operation Attributes (Set by Client)	4940
198	6.3.1 operation-locale (type3Locale)	4941
199	6.3.2 operation-notification-address (name)	4941
200	6.3.3 operation-user-name (name)	5041
201	6.3.4 operation-host-name (name)	5041
202	6.4 Printer Attributes (Set by the Administrator)	5041
203	6.4.1 printer-name (name)	5142
204	6.4.2 printer-location (string)	5142
205	6.4.3 printer-model (string)	5142
206	6.4.4 printer-types (type2Enum)	5142
207	6.4.5 printer-state (type1Enum)	5243
208	6.4.6 printer-state-message (string)	5344
209	6.4.7 message (string)	5344
210	6.4.8 locale (type3Locale)	5344
211	6.4.9 notification-events (#type2Enum)	5345
212	6.4.10 notification-addresses (#name)	5345
213	6.4.11 end-user-acl (#name)	5445
214	6.4.12 maximum-printer-speed (positiveIntegerUnits)	5445
215	6.4.13 fonts-substitutions (#stringPair)	5445
216	6.4.14 fonts-supported (1#stringState)	5446
217	6.4.15 media-supported (1#nameState)	5446
218	6.4.16 document-formats-supported (1#type2FormatState)	5446
219	6.4.17 numbers-up-supported (1#positiveIntegerState)	5546
220	6.4.18 finishings-supported (#type2EnumState)	5546
221	6.4.19 sides-supported (1#type2EnumState)	5547
222	6.4.20 print-qualities-supported (1#type2EnumState)	5547
223	6.4.21 printer-resolutions-supported	
224	(1#positiveIntegerCrossState)	5547
225	6.4.22 code-sets-supported (1#type3EnumState)	5647
226	6.4.23 off-peak-times-supported (#type3EnumState)	5647
227	6.4.24 events-supported (#type2EnumState)	5648
228	6.4.25 locales-supported (1#type3LocaleState)	5648
229	6.4.26 job-sheets-supported (#type3EnumState)	5648

230	6.4.27 maximum-copies (positiveInteger) .....	<a href="#">5748</a>
231	6.4.28 maximum-job-octets (positiveInteger) .....	<a href="#">5748</a>
232	6.4.29 maximum-impressions (positiveInteger) .....	<a href="#">5749</a>
233	6.4.30 maximum-media-sheets (positiveInteger) .....	<a href="#">5749</a>
234	6.4.31 maximum-job-retention-period (deltaTime) .....	<a href="#">5749</a>
235	6.4.32 maximum-end-user-priority (typeEnum) .....	<a href="#">5849</a>
236	6.5 Job Templates .....	<a href="#">5849</a>
237	6.6 Conformance .....	<a href="#">5850</a>
238	7. Security Considerations .....	<a href="#">5950</a>
239	8. References .....	<a href="#">5950</a>
240	9. Author's Address .....	<a href="#">5951</a>
241	10. Appendix A: .....	<a href="#">6153</a>
242		

## 243 1. Introduction

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

## 255 2. Distributed Printing

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

### 265 2.1 Generic Print System Components

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

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

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

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

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

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

## 2.2 IPP Components

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

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

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

Clients use~~interact with these using~~ the following operations:

- Print (section xxx5.4.1)
- Cancel Job (section 5.4.2~~xxx~~)
- Get Attributes(section 5.4.3~~xxx~~)
- Get Jobs (section 5.4.4~~xxx~~)

## 3. IPP Objects

This section describes the IPP objects.

## 326 3.1 Printer

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

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

345 Some examples of configurations containing IPP Printer object  
 346 include:

- 347 - An output device, with no spooling capabilities, supporting
- 348 IPP
- 349 - An output device, with a built-in spooler, supporting IPP
- 350 - A print server with one or more associated output devices with
- 351 the print server supporting IPP.
- 352 - The associated output devices may or may not be capable of
- 353 spooling jobs
- 354 - The associated output devices may or may not support IPP
- 355 ~~A print server with one or more downstream print servers~~
- 356 ~~and/or output devices where the upstream print server supports~~
- 357 ~~IPP~~
- 358 ~~ISSUE: Is this previous example too much?~~

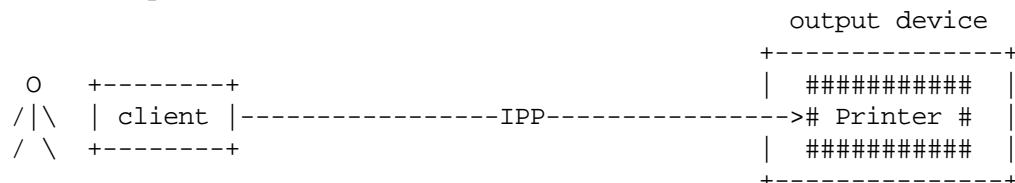
359  
 360 See the following figures for some examples on how to view IPP  
 361 Printer objects on top of other printing system models:

Legend:

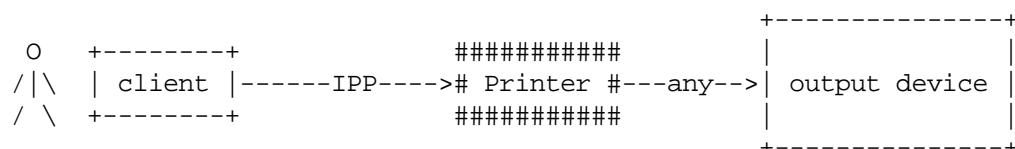
##### indicates an IPP printer object which is either embedded in an output device or is hosted in a server. An IPP printer object may or may not queue/spool.

any indicates any network protocol or direct connect, including IPP

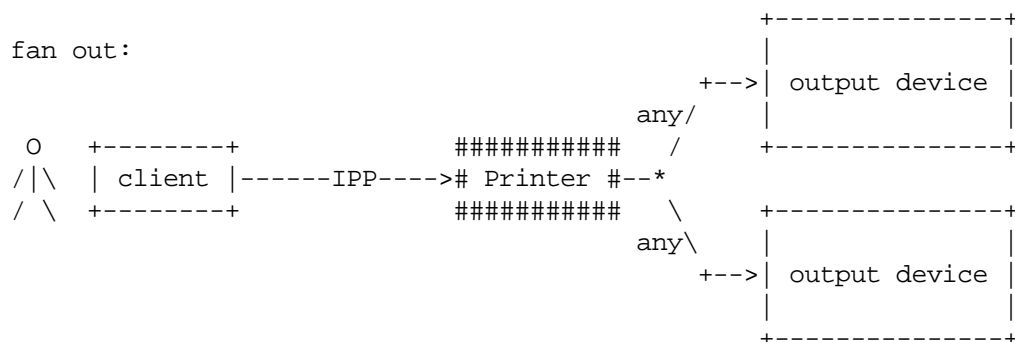
embedded printer:



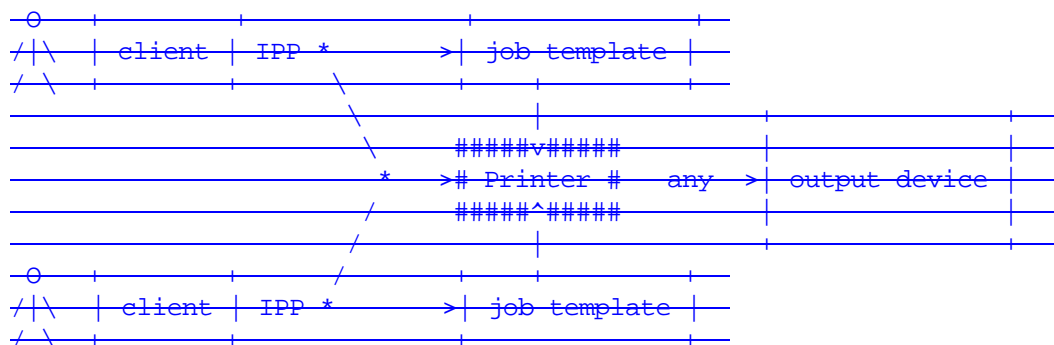
hosted printer:



fan out:



~~fan in:~~





### 3.2 Job

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

Job attributes are broken up into the following groups :~~provide information to~~

- Job Informational (sections 6.2.1, 6.2.2)
- Job Status (section 6.2.3)
- Job Sheet (section 6.2.4)
- Notification (section 6.2.5)
- Job Scheduling (section 6.2.6)
- Job Production (section 6.2.7)
- Conversion of Text Files (section 6.2.8)
- Job Resource (section 6.2.9)
- Number of Documents (section 6.2.10)
- Document Attributes (6.2.11)
- ~~— identify the print job (section xxx) —~~
- ~~— assist in selecting the Printer (section xxx) —~~
- ~~— report job status (section xxx) —~~
- ~~— assist in scheduling and processing (section xxx) —~~
- ~~— describe the documents in the job (section xxx) —~~
- ~~— produce the document (section xxx) —~~

~~ISSUE: This list needs to be fixed to match the final job attributes sections (they don't match right now).~~

### 3.3 Job Template

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

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

### 3.4 Object Relationships

Instances of objects within the system have relationships which must be maintained persistently along with the persistent storage of the objects themselves. A Printer can contain zero, or more

467 Job objects. A Job object contains one or more Documents. A  
 468 Printer object is associated with zero~~one~~ or more Job Template  
 469 objects.

### 470 3.5 Object Identity

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

473 The following objects have the following mandatory identifier  
 474 attributes:

475 Object	Identifier	Containing Object
476 Printer	printer-name	None
477 Job	job-identifier	Printer
478		
479 Job Template	job-template-name	None
480		
481		

### 482 4. Naming

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

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

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

490 Another example is the following URL:

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

492 In this case, the URL identifies the use of the HTTP protocol.  
 493 The Printer is located at the node identified by the IP address of  
 494 "1.2.3.4" using port 380 for the HTTP server, and "printer-2" is  
 495 the name of the Printer.

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

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

- 502 1) ~~1)~~ http://some.domain.com/two-sided-printer 1 and
- 503 2) ~~2)~~ 2) http://some.domain.com/draft-printer.
- 504

505 These look like two different Printers locations, but  
 506 underneath they represent the same Printer object, but that  
 507 Printer object has two Job Templates and each is exposed  
 508 through a different URL for the same Printer object. Each one  
 509 the the associated Job Templates would have using two different  
 510 Job Template default attribute sets.  
 511

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

~~ISSUE: Should IPP propose a new standard port number (say 380) for an HTTP server which has been optimized to support the IPP protocol over that HTTP implementation? IPP should work for any valid HTTP server, however, there might be some specializations can be performed for IPP operations.~~

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

#### 4.1 Directory Services

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

#### 4.2 Directory Entry Schema

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

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

~~Name, description, owner, location, address~~

## 559 4.2.1 Name

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

## 563 4.2.2 Description

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

## 566 4.2.3 Location

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

569 In order for filtered searches to be more effective, a given site  
570 may use some regular structuring within the string values such as  
571 "SITE:USA-San Jose,BUILDING:A1,FLOOR:2,ROOM:555" or "department5-  
572 2ndFloor-A5-IndianHills-Chicago-IL-USA".

573 4.2.4 Maximum Print Quality

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

576 ~~ISSUE: Does this subsume the need for Resolution and Speed?~~

## 577 4.2.5 Cost

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

## 580 4.2.6 Resolution

581 This is the maximum resolution of the Printer in dpi.

582 The syntax shall be the same as that of the printer-resolution-  
583 select job attribute. That syntax allows a single integer to  
584 specify the maximum resolution or a pair of integers to specify  
585 the maximum resolution when the x and y dimensions differ. When  
586 two integers are specified, the first is in the x direction, ie.,  
587 the direction fo the shortest dimension of the medium, so that the  
588 value is independent of whether the Printer feeds long edge or  
589 short edge first.

## 590 4.2.7 Color Supported

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

## 594 4.2.8 Fonts Supported

595 This attribute takes on a list of fonts that are supported by the  
596 printer. This is replicated from the fonts-supported attribute in  
597 the Printer object.

## 4.2.9 Maximum Speed

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

~~ISSUE: Should this be "high", "medium", and "low"??~~

~~ISSUE: Delete??~~

## 4.2.10 Device Id

~~This is the IEEE P1284 Device Id.~~ This attribute can be used for automatic driver download , database access, or ~~and~~ other automatic configuration tasks. It ~~might~~ ~~can~~ be used to generate a platform specific id such as the Windows Plug-and-Play id.

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

4.2.11 Make and Model

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

~~ISSUE: Is this needed if we use the P1284 Device Id??~~

~~4.2.12 Manufacturer~~

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

~~ISSUE: Is this needed if we use the P1284 Device Id??~~

~~ISSUE: Would the company name registered with IAN for use in specifying Internet addresses be required? Recommended?~~

4.2.12 Marker Type

This is the printing mechanism of the print device: laser, ink jet, thermal, etc. The syntax and values are the same as for the printer-types Printer attribute , except the . ~~The~~ value of the attribute shall be single-valued , ~~while the printer types attribute is multi-valued.~~ ~~A Printer that is of several types may appear multiple times in the directory.~~ ~~ISSUE: no.~~

~~ISSUE: Is this needed if we use the P1284 Device Id??~~

635 4.2.13 Document Formats ~~s~~-Supported

636 This is a list of all of the document formats that the printer  
637 and/or its interpreter(s) support. The syntax and values are the  
638 same as those for the document-formats-supported Printer  
639 attributes.

## 640 4.2.14 Sides Supported

641 This attribute specifies the capabilities of the Printer for  
642 marking on sides of the medium. The syntax and values shall be  
643 the same as the sides-supported Printer attribute. Standard  
644 values are: 1-sided (simplex), 2-sided-long- ~~binding~~ edge  
645 (duplex), and 2-sided-short- ~~binding~~ edge (tumble).

## 646 4.2.15 Finishings Supported

647 This attribute identifies the finishing operations supported by  
648 the Printer. The standard finishing objects are defined in the  
649 section on the finishing job attribute.

650 4.3 Directory Entries Using LDAP

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

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

660 To create a Printer object directory entry using LDAP:

661 1. An administrator uses a program to create an entry for the  
662 Printer object on a directory server that supports L DAP. The  
663 administrator defines the Distinguished Name (dn) and the default  
664 subjective attributes for the Printer object directory entry.  
665 Issue: Should the administrator also define default objective  
666 attributes or wait for the Printer object itself to initialize  
667 these attributes?

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

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

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

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

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

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

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

5. The Printer object invokes the ldap\_unbind API to close the connection to the directory server.

Example: ldap\_unbind (ld)

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

To locate a Printer object entry using LDAP, a program can use the ldap\_search or ldap\_search API s 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.

~~The following symbols are used in the tables below:~~

~~P — perform the operation directly~~

~~PF — perform the operation; forward to Output Device sometimes~~

~~UA — unsupported in an Output Device unless it supports queuing~~

~~U — unsupported operation~~

~~IPP defines the following end user operations:~~

<del>Operation</del>	<del>Print Server</del>	<del>Output Device</del>
<del>Print</del>	<del>PF</del>	<del>P</del>
<del>Cancel Job</del>	<del>PF</del>	<del>P</del>
<del>Get Attributes</del>	<del>PF</del>	<del>P</del>
<del>Get Jobs</del>	<del>PF</del>	<del>P</del>

### 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

715 used for any task through extension of its request methods  
 716 (commands).

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

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

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

## 737 5.2 IPP Operation Encoding

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

740 HTTP MESSAGE = Request | Response

741  
 742 Requests and responses use the generic message format of RFC 822  
 743 for transferring entities. Both messages may include optional  
 744 header fields and an entity body. The entity body is separated  
 745 from the headers by a null line (a line with nothing preceding the  
 746 CRLF).

747  
 748 Request = Request-line  
 749 \* (General-Header  
 750 Request-Header  
 751 Entity-Header)  
 752 CRLF  
 753 [ Entity-Body ]

754  
 755 Response = Status-line  
 756 \* (General-Header  
 757 Request-Header  
 758 Entity-Header)  
 759 CRLF  
 760 [ Entity-Body ]

761  
 762 All IPP headers conform to the syntax

763 IPP Header = field name ":" [field-value] CRLF.  
 764



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

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

#### 5.2.1 HTTP Request-Header Fields

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

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

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

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

##### 5.2.1.1 IPP Request-Line

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

IPP Request-Line = Operation token IPP/1.0 CRLF

Operation token = Print | CancelJob | GetAttributes | GetJobs

#### 5.2.2 HTTP Response -Header Fields

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

##### 5.2.2.1 IPP Status-Line

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

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

### 813   5.3 The Print Job

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

820  
 821       Print Job = Print-Job-Object-Header                   ;section ( 5.3.1)  
 822                         [Job Attributes]                   ;section ( 5.3.4)  
 823                         \*(Documents)  
 824                           
 825                           
 826                         Document = Document-Header           ;section ( 5.3.2)  
 827   [Document attributes] ;section ( 5.3.5)  
 828   [Content-Header       ;section ( 5.3.3)  
 829   content]  
 830                         

#### 831   5.3.1 Print Job Object Header

832       Print-Job-Object Header = Content-Encoding  
 833   Content-Length  
 834   Content-Type  
 835   extension-header  
 836   

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

#### 839   5.3.2 Document Header

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

845       Document Header = Content-Encoding  
 846   Content-Length  
 847   Content-Type  
 848   extension-header  
 849   

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

#### 852   5.3.3 Document-Content Header

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

858  
 859       Document-Content-Header =                   Content-Encoding

	<u>Content-Length</u>
	<u>Content-Type</u>
	<u>extension-header</u>

Content-Type is defined as :

Content-Type = Data Stream Format "/" Version

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

Content-Type: Postscript/2.0

Other header fields are as defined by HTTP 1.0.

#### 5.3.4 Job Attributes

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

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

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

#### 5.3.5 Document Attributes

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

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

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

### 5.4 Operation Semantics

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

#### 5.4.1 Print Operation

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

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

##### 5.4.1.1 Print Request

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

<u>Job and Document Attributes</u>	<u>A set of Job object and Document attributes as defined in section 6.2</u>
------------------------------------	--

<u>Document Contents</u>	<u>Document content is optional and not included when a URL is provided to point to the content.</u>
--------------------------	--

#### 5.4.1.2 Print Response

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

<u>Job-Identifier</u>	<u>A URL Used for all other operations on this Job.</u>
-----------------------	---

<u>Job Status</u>	<u>Current-job-state</u>
-------------------	--------------------------

<u>Printer State</u>	<u>Printer-state</u>
----------------------	----------------------

<u>Message</u>	<u>Optional message</u>
----------------	-------------------------

<u>Errors</u>	<u>Optional Error Information</u>
---------------	-----------------------------------

#### 5.4.2 Cancel Job Operation

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

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

##### 5.4.2.1 CancelJob Request

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

<u>Message</u>	<u>Optional message to the operator.</u>
----------------	--

##### 5.4.2.2 CancelJob Response

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

<u>Job Status</u>	<u>Optional Job status information</u>
-------------------	--

<u>Errors</u>	<u>Optional Error Information</u>
---------------	-----------------------------------

### 5.4.3 Get Attributes Operation

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

#### 5.4.3.1 GetAttributes Request

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

<u>Selector</u>	<u>Job-Identifier (URL) or Printer URL</u>
-----------------	--

<u>Requested Attributes</u>	<u>A set of attributes in which the requestor is interested</u>
---------------------------------	---

#### 5.4.3.2 GetAttributes Response

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

<u>Result Attributes</u>	<u>The requested attributes of the object</u>
------------------------------	---

<u>Errors</u>	<u>Optional error information</u>
---------------	-----------------------------------

### 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:

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

#### 5.4.4.2 Get Jobs Response

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

963

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

964

965

966

967

968 ~~5.5 IPP Operations Using HTTP~~

969 ~~All IPP operations are defined using HTTP as the underlying~~  
970 ~~communication protocol.~~

971 ~~5.4.5 HTTP Overview~~

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

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

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

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

997 ~~5.4.5 IPP Operation Encoding~~

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

1000 ~~HTTP MESSAGE = Request | Response~~

1001

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

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

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

~~All IPP headers conform to the syntax~~

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

~~IPP/1.0 defines the octet sequence CR LF as the end of line marker for all protocol elements except the entity body. In this document, the sequence CR LF is shown as CRLF.~~

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

#### ~~5.4.4.3 HTTP Request Header Fields~~

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

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

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

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

1054       ~~The http method token indicates the method to be performed on the~~  
 1055       ~~resource identified by the Request URL. The method is case~~  
 1056       ~~sensitive. The http methods used will be "Post" and "Get".~~

#### 1057   ~~5.4.4.3 Print~~

1058       ~~The Print operation allows a user to submit a Print Job to the~~  
 1059       ~~print server. A Print Job contains the information needed by the~~  
 1060       ~~Print object to print a document or set of documents. When the~~  
 1061       ~~print operation is invoked, the Entity Body included in the HTTP~~  
 1062       ~~request is an IPP Print Job. The concrete syntax of the Print Job~~  
 1063       ~~is defined in section xxx. The response to a print request~~  
 1064       ~~includes the Job Identifier (a URL) assigned by the Printer.~~

#### 1065   ~~5.4.4.3 Cancel Job~~

1066       ~~This method allows a user to cancel one specific Print Job any~~  
 1067       ~~time after the print job has been established on the Printer~~  
 1068       ~~Object. Some pages may be printed before a job is terminated if~~  
 1069       ~~printing has already started when the Cancel Job operation is~~  
 1070       ~~received.~~

1071       ~~The Cancel HTTP request will be sent to the URL identifying the~~  
 1072       ~~job to be canceled.~~

#### 1073   ~~5.4.4.3 Get Attributes~~

1074       ~~This operation allows a user to obtain information from the Print~~  
 1075       ~~object concerning jobs, printers, and print queues, based on ISO~~  
 1076       ~~10175. The entity body of the Get Attributes operation contains~~  
 1077       ~~the set of attributes that the requester is interested in.~~  
 1078       ~~However, the attribute values may be null and are ignored by the~~  
 1079       ~~server. The attribute list is returned in the response with the~~  
 1080       ~~appropriate attribute values filled in. If no attribute list is~~  
 1081       ~~supplied, then all attributes defined for that object are~~  
 1082       ~~returned.~~

#### 1083   ~~5.4.4.3 Get Jobs~~

1084       ~~The Get Jobs operation allows a client to retrieve attributes of~~  
 1085       ~~the specified job.~~

#### 1086   ~~5.4.5 The Print Job~~

1087       ~~The entity body of a print request will contain a Print Job, as~~  
 1088       ~~defined below. The headers defined here are IPP headers, but~~  
 1089       ~~follow the same syntax as the basic HTTP headers.~~

1090  
 1091       ~~Print Job = Print Job Object Header                   section (1.2.1)~~  
 1092       ~~[Job Attributes]                               section (1.2.4)~~  
 1093       ~~\*(Documents)~~  
 1094  
 1095       ~~Job Attribute = Attribute name : Attribute value CRLF~~  
 1096  
 1097       ~~Document = Document Header                           section (1.2.2)~~  
 1098       ~~[Document attributes]                        section (1.2.5)~~  
 1099       ~~[Content Header                               section (1.2.3)~~



~~content]~~  
~~5.4.4.3 Print Job Object Header~~  
~~Print Job Object Header = Content Encoding~~  
~~Content Length~~  
~~Content Type~~  
~~extension header~~  
~~Content Type is always "IPP Print Object". Other header fields~~  
~~are as defined for HTTP 1.0.~~  
~~5.4.4.3 Document Header~~  
~~The document header allows the insertion of multiple documents~~  
~~within a job. At this point only a limited number of document~~  
~~attributes are defined. However, this structure allows the~~  
~~addition of other attributes which can be specified on a document~~  
~~boundary.~~  
~~Document Header = Content Encoding~~  
~~Content Length~~  
~~Content Type~~  
~~extension header~~  
~~Content type is always "IPP Document". Other header fields are as~~  
~~defined in HTTP 1.0.~~  
~~5.4.4.3 Document Content Header~~  
~~The document content header provides additional meta information~~  
~~about the document. The document content header is an optional~~  
~~field and would not be present if the document was pointed to by a~~  
~~document URL attribute. It is composed of a number of document~~  
~~header fields as follows:~~  
~~Document Content Header = Content Encoding~~  
~~Content Length~~  
~~Content Type~~  
~~extension header~~  
~~Content Type is defined as:~~  
~~Content Type = Data Stream Format "/" Version~~  
~~Thus, for example, if the document to be printed was a Postscript~~  
~~Level 2 document, the Content Type would be specified as:~~  
~~Content Type: Postscript/2.0~~  
~~Other header fields are as defined by HTTP 1.0.~~  
~~5.4.4.3 Job Attributes~~  
~~Job attributes are defined in section xxx. Attributes will always~~  
~~be sent as~~

1147 ~~Job Attribute = attribute name ":" Attribute value CRLF~~

1148 ~~Attribute value = Value | \*(Value "\", " Value)~~

1150 ~~Attribute value = Value | \*(Value "\", " Value)~~

#### 1153 ~~5.4.4.3 Document Attributes~~

1154 ~~Document attributes are defined in section yyy. At this point a~~  
 1155 ~~limited number of attribute may be specified on a document basis.~~  
 1156 ~~The syntax for a document attribute is~~

1157 ~~Document Attribute = attribute name ":" Attribute value CRLF~~

1158 ~~Attribute value = Value | \*(Value "\", " Value)~~

#### 1161 ~~5.5 Print Operation~~

1162 ~~When an end user submits a job, the client submits a Print~~  
 1163 ~~Request according to the syntax and semantics of this standard and~~  
 1164 ~~receives a Print Response according to this standard. The end~~  
 1165 ~~user or submitting application selects a Printer which implies a~~  
 1166 ~~Job Template.~~

1167 ~~[Further work needs to done to define the above concept.]~~

#### 1168 ~~5.4.5 Print Request~~

1169 ~~The following abstract data types are part of the Print Request.~~  
 1170 ~~Note: The Printer name is not needed since it is a URL and it is~~  
 1171 ~~the target of the entire operation.~~

<del>Job and</del>	<del>A set of Job object and Document attributes as</del>
<del>Document</del>	<del>defined in section xxx</del>
<del>Attributes</del>	
<del>Job Return</del>	<del>The set of Job attributes to return in the</del>
<del>Attributes</del>	<del>response</del>
<del>Printer Return</del>	<del>The set of Printer attributes to return in the</del>
<del>Attributes</del>	<del>response</del>
<del>Document</del>	<del>Note: What if there are multiple documents and</del>
<del>Contents</del>	<del>each has a different size? How does this map on</del>
	<del>top of the HTTP header that has one size? Does</del>
	<del>it require multiple HTTP operations?</del>

#### 1174 ~~5.4.5 Print Response~~

1175 ~~The following abstract data types are part of the Print Response:~~

1176 ~~Job Id Used for all other operations on this Job.~~

~~Job Status Job state information~~

~~Printer State Optional Printer state information~~

~~Message~~                      ~~Optional message Note: Is this needed?~~

~~Errors~~                      ~~Optional Error Information~~

## ~~5.5 Cancel Job Operation~~

### ~~5.4.5 Cancel Job Request~~

~~The following abstract data types are part of the Cancel Job Request. Note: The Job Id is not needed as data within the operation since the Job URL is the target of the entire operation.~~

~~Document Number~~      ~~Optional document number of the document to Issue: We don't need this if we do not allow canceling of a single document in a multi-document job, right?~~

~~Message~~                      ~~Optional message to the operator.~~

### ~~5.4.5 Cancel Job Response~~

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

~~Job Status~~                      ~~Optional Job status information~~

~~Errors~~                      ~~Optional Error Information~~

## ~~5.5 Get Attributes Operations~~

### ~~5.4.5 Get Attributes Request~~

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

~~Selector~~                      ~~A Job Id or Printer Name URL ISS UE: Is this just the target URL of the operation and it is not needed here within the operation?~~

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

### ~~5.4.5 Get Attributes Response~~

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

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

~~Errors~~                      ~~Optional error information~~

1200 ~~5.5 Get Jobs Operation~~1201 ~~5.4.5 Get Jobs Request~~

1202 ~~The following abstract data types are part of the Get Jobs~~  
 1203 ~~Request:~~

1204     ~~Filtering~~           ~~A lightweight filtering mechanism, such as all~~  
                           ~~jobs versus a particular end user's jobs.~~  
    ~~Requested~~           ~~A set of job attributes in which the requestor~~  
    ~~Attributes~~          ~~is interested~~

1205

1206

1207 ~~5.4.5 Get Jobs Response~~

1208 ~~The following abstract data types are part of the Get Jobs~~  
 1209 ~~Response:~~

1210

~~Result~~               ~~Attribute set containing the returned results.~~  
    ~~Attributes~~  
    ~~Errors~~              ~~Optional Error Information~~

1211

## 1212 6. Object Attributes

1213 This section describes the attributes, syntaxes, and values that  
 1214 are part of IPP. The sections below show the objects and their  
 1215 associated attributes which are included within the scope of this  
 1216 protocol. The text in these sections has been heavily influenced  
 1217 by the ISO/IEC 10175 DPA (Final, June 1996).

1218

1219

1220

## 1221 6.1 Attribute Syntaxes

1222 The sections below reference the following syntax items:

string	arbitrary ASCII strings, no control characters, except <SPACE>. <del>TBD</del>
stringPair	string ":" string
stringState	string state
name	arbitrary ASCII strings, no control characters, and no <SPACE> characters. <del>TBD</del>
URL	Universal Resource Locator <del>TBD</del>
dateTime	date and time in RFC 822 format <del>TBD</del>
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. <del>TBD</del>
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

Type3Enum	private (un-registered) with a suitable distinguishing prefix, such as -xxx- where xxx is the company name registered with IANA.
	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. <del>TBD</del>
type2EnumState	type2Enum state
type3EnumState	type3Enum state
state	<a href="#">type1Enum</a> <del>TBD</del>
Boolean	tokens: yes, y, true, or t and no, n, false, or f. <del>TBD</del>
positiveInteger	1 .. n represented as ASCII digits <del>TBD</del>
positiveIntegerCross	positiveInteger [ "x" positiveInteger ]
positiveIntegerCross	positiveIntegerCross state
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

Also, the following conventions are used:

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

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

## 6.2 Job Attributes

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

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

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

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

1240       The client may also specify these attributes in the operations:  
1241       Get-Attributes, and Get-Jobs.

#### 1242   6.2.1.1 job-name (string)

1243       This attribute supplies a human readable string for naming the  
1244       print-job.

1245       This attribute is intended for to be printed on a start sheet,  
1246       returned in a Get-Jobs result, or used in notification messages.

1247       If the client does not specify this attribute, a Printer shall set  
1248       it to the name of the file of the first document in the job.

#### 1249   6.2.2 Job Informational Attributes (Set by a Printer)

1250       The Print shall add all of these attributes to a job to provide  
1251       information to identify a print-job.

1252       The client may specify these attributes in the operations: Get-  
1253       Attributes and Get-Jobs, but not in Print.

##### 1254   6.2.2.1 job-identifier (url)

1255       This attribute provides the job-identifier for this job on the  
1256       Printer. The Printer shall generate a job-identifier value as a  
1257       URL..

1258       The value of the job-identifier attribute shall be returned by the  
1259       Printer as part of the PrintResult in the Print operation.

##### 1260   6.2.2.2 job-originator (name)

1261       This attribute specifies the name of the person submitting the  
1262       print job. The Printer shall set this attribute to the most  
1263       authentic name that it can obtain from the client. The operation-  
1264       user-name attribute is intended to be a source of the most  
1265       authentic name.

##### 1266   6.2.2.3 job-originating-host (name)

1267       This attribute identifies the originating host of the job. The  
1268       Printer shall set this attribute to the value of the operation-  
1269       host-name which is intended to be the most authentic host name of  
1270       the client.

##### 1271   6.2.2.4 notification-address (name)

1272       This address specifies the email address of the client. The client  
1273       specifies this attribute in the operation-notification-address  
1274       attribute which the Printer in turn uses to set this attribute.

1275       The Printer shall use this attribute as the address for sending  
1276       messages to a job submit ~~ter~~er when an event occurs that the end user  
1277       has registered an interest in or when certain other events occur,  
1278       such as Cancel-Job.

1279       Note: The only type of notification is email.

1280 | ~~IssueSSUE: Can the email address be inferred with job-originator~~  
 1281 | ~~and the originating-host ?~~

#### 1282 | 6.2.2.5 job-locale (type3Locale)

1283 | This attribute identifies the locale of the job. The Printer sets  
 1284 | this attribute from the value of the operation-locale.

1285 | The Printer shall use this attribute to determine the locale for  
 1286 | notification messages that it sends.

1287 | The type3EnumTrip consists of 3 colon separated type 3 enums. The  
 1288 | first shall be the two-character country code from ISO 639. The  
 1289 | second shall be the two-character language code from ISO 3166. The  
 1290 | third is the code-set from the IANA Code Set Registry.

1291 | ~~IssueSSUE: Is there a more standard syntax for locale?~~

#### 1292 | ~~6.2.3 Printer Selection Attributes (Set by Client/End User)~~

1293 | ~~The client shall specify this attribute to select a particular~~  
 1294 | ~~Printer.~~

1295 | ~~The client may also specify these attributes in the operations:~~  
 1296 | ~~Get Attributes, and Get Jobs.~~

1297 | ~~Issue: this attribute may be implicit in the specified URL in the~~  
 1298 | ~~Print operation.~~

#### 1299 | ~~6.2.2.6 printer name requested (URL)~~

1300 | ~~This attribute identifies the printer that the client requests~~  
 1301 | ~~for printing the job.~~

1302 | ~~ISSUE: We decided to delete this attribute~~

#### 1303 | 6.2.3 Job Status Attributes (Set by Printer)

1304 | The Printer shall add these attributes to a job when a client  
 1305 | submits a job, and the Printer shall assign appropriate values to  
 1306 | each such job-status attribute.

1307 | The Printer uses these attributes to specify the job status  
 1308 | before, during and after the processing of the print-job by the  
 1309 | Printer.

1310 | The client may specify job-status attributes in: Get-Attributes  
 1311 | and Get-Jobs, but not Print.

#### 1312 | 6.2.3.1 current-job-state (type1Enum)

1313 | This attribute identifies the current state of the job with the  
 1314 | following values: unknown, pre-processing, pending, processing,  
 1315 | printing, held, terminating, retained, completed.

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

<del>Pre-processing</del>	<del>The job has been created on the server by the create job sub operation of the print request, but a print request with a TRUE value for the job-submission-complete component of the PrintArgument has not yet been received and no document has started processing. The job maybe in the process of being checked by the server for attributes, defaults being applied, a printer being selected, etc.</del>
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

	<u>The server has completed processing the job and the output device is currently printing the job on at least one printer. That is, a print engine 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)</u>
<del>printing</del>	<del>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).</del>
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.

1316

1317        The IPP protocol supports all values for job states, but Printers  
 1318        are need only support those states which are appropriate for the  
 1319        particular implementation.

1320        6.2.3.2 output-device ~~printer~~-assigned (name)

1321        This attribute identifies the Output Device to which the Printer  
 1322        has assigned this job.

1323        If an Output Device implements a Printer, the Printer does not set  
 1324        this attribute.

1325        If a Print Server implements a Printer, the value shall be empty  
 1326        until the Printer assigns an Output Device to the job..

1327 ~~ISSUE: Is this attribute appropriate for a model in which we are~~  
 1328 ~~hiding the downstream Printer. The printers assigned value shall~~  
 1329 ~~not be the same as the printer requested by the end user.~~

1330 The value of the job's printer-assigned attribute shall remain  
 1331 after the job has completed, so that end users can determine the  
 1332 Output Device on which the job was printed.

#### 1333 6.2.3.3 submission-time (dateTime)

1334 This attribute indicates the time at which this job was accepted  
 1335 by the Printer. If the Printer does not support the notion of  
 1336 time, the attribute is not stored as part of the job object.

#### 1337 6.2.3.4 job-message-from-administrator (string)

1338 This attribute provides a message from an operator, system  
 1339 administrator or "intelligent" process to indicate to the end user  
 1340 the reasons for modification or other management action taken on a  
 1341 job.

#### 1342 6.2.3.5 completion-time (dateTime)

1343 This attribute indicates the time at which this job completed.  
 1344 This time is useful for jobs which are retained after printing.  
 1345 If the Printer does not support the notion of time, the attribute  
 1346 is not stored as part of the Job object.

#### 1347 6.2.3.6 job-state-reasons (1#type2Enum)

1348 This attribute identifies the reason or reasons that the job is in  
 1349 the state that it is in (e.g., held, terminating, retained,  
 1350 completed, etc.). The printer shall indicate the particular  
 1351 reason(s) by setting the value of the job-state-reasons attribute.  
 1352 It is valid for the printer to set the value of the job-state-  
 1353 reasons attribute to the empty set.

1354 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.

1355

1356

1357 6.2.3.7 impressions-completed (cardinal)

1358 This attribute contains the number of impressions that the Printer

1359 has completed printing. If the Printer cannot report this number,

1360 the Printer leaves this attribute unspecified.

1361 6.2.3.8 media-sheets-completed (cardinal)

1362 This attribute contains the number of media-sheets that the

1363 Printer has completed printing. If the Printer cannot report this

1364 number, the Printer leaves this attribute unspecified.

1365 6.2.4 Job Sheet Attributes (Set by Client/End User)

1366 The client shall specify these attributes to control the printing

1367 of job sheets.

1368 The client may also specify job sheet attributes in: Get-

1369 Attributes and Get-Jobs.

1370 job-sheets (type3Enum) This attribute determines what type of job-

1371 sheets the Printer shall print with the job.

1372 The standard values are: none, and default-sheet.

1373 The value "none" means that the Printer shall print no job sheets.

1374 The value "default-sheet" means that the Printer shall print the

1375 job sheets defined by an administrator. If the administrator's

1376 policy is not to support none, the Printer shall use the default-

1377 sheet value if the client supplies the "none" value.

1378 6.2.5 Notification Attributes (Set by a Client/End User)

1379 The client shall specify these attributes to indicate events that

1380 the client is interested in.

1381 The client may also specify notification attributes in: Get-

1382 Attributes and Get-Jobs.

## 1383 6.2.5.1 notification-events (#typeEnum)

1384 This attribute specifies the events about which the end user want  
1385 to be notified.

1386 This attribute will support four events classes: none, job-  
1387 completion, job-problems and printer-problems.

1388 If this attribute contains the event none, the Printer shall not  
1389 notify. This value is useful if an administrator has set up a  
1390 notification Printer default but the end user does not which  
1391 notification. If the none value and other values are supplied,  
1392 the Printer shall ignore the none value.

1393 This attribute will support only one delivery method, namely  
1394 email. The attribute notification-address specifies the email  
1395 address.

1396 If this attribute contains the event job-completion, the Printer  
1397 shall notify the client when the job containing this attribute  
1398 completes.

1399 If this attribute contains the event job-problem, the Printer  
1400 shall notify the client when the job containing this attribute has  
1401 a problem while the job is printing. Problems include: paper jam  
1402 and out-of-paper.

1403 If this attribute contains the event printer-problem, the Printer  
1404 shall notify the client when the job containing this attribute has  
1405 a problem while the job is printing or waiting to print. Problems  
1406 include: paper jam and out-of-paper.

1407

1408 ~~ISSUE: Email is quite deficient for timely notification to an end~~  
1409 ~~user who receives a lot of email, but there are no other choices.~~  
1410 ~~The internet community needs to solve this problem, perhaps with~~  
1411 ~~an extremely urgent email. Intranet usage may have other~~  
1412 ~~solutions.~~

## 1413 6.2.6 Job Scheduling Attributes (Set by Client/End User)

1414 The client shall specify these attributes to provide the Printer  
1415 with information for the scheduling a print-job.

1416 The client may also specify these attributes in: Get-Attributes  
1417 and Get-Jobs.

1418

## 1419 6.2.6.1 job-priority (typeEnum)

1420 This attribute specifies a priority for scheduling the print-job.  
1421 Printers that employ a priority-based scheduling algorithm use  
1422 this attribute.

1423 There are three standard values: high, default, and low. Among  
1424 those jobs that are ready to print, a Printer shall print all such

1425 jobs with a high priority before printing those with a default or  
1426 low priority, and a Printer shall print all such jobs with a  
1427 default priority before printing those with a low priority.

1428 If the client does not specify this attribute, the Printer assumes  
1429 that the end user places no constraints concerning priority on the  
1430 scheduling of the print-job, and it has a priority value of  
1431 default

1432 An operator can modify a job to have any priority. An end-user is  
1433 restricted by the value of the maximum-end-user-priority Printer  
1434 attribute.

#### 1435 6.2.6.2 job-print-after (dateTime)

1436 This attribute specifies the calendar date and time of day after  
1437 which the print-job shall become a candidate for printing.

1438 If the value of this attribute is in the future, the Printer shall  
1439 set the value of the job's current-job-state to held and add the  
1440 job-print-after-specified value to the job's job-state-reasons  
1441 attribute and shall not schedule the print-job for printing until  
1442 the specified date and time has passed. When the specified date  
1443 and time arrives, the Printer shall remove the job-print-after-  
1444 specified value from the job's job-state-reason attribute and, if  
1445 no other reasons remain, shall change the job's current-job-state  
1446 to pending so that the job becomes a candidate for being scheduled  
1447 to print.

1448 If this attribute is unspecified or the value is in the past, the  
1449 job shall be a candidate for scheduling immediately.

#### 1450 6.2.6.3 job-print-off-peak (type3Enum)

1451 This attribute specifies the off-peak period during which the  
1452 print-job shall become a candidate for printing.

1453 Standard values are: "evening", "night", "weekend", "second-  
1454 shift", "third-shift".

1455 If this attribute is specified, it contains a value with which an  
1456 administrator has associated allowable print times. An  
1457 administrator is encouraged to pick names that suggest the type of  
1458 off-peak period.

1459 If this attribute is unspecified, the job shall be a candidate for  
1460 scheduling immediately.

#### 1461 6.2.6.4 job-retention-period (deltaTime)

1462 The retention time is expressed in hours and minutes, e.g. 6:00 (6  
1463 hours), or 20 (20 minutes).

1464 This attribute specifies the minimum period of time following the  
1465 completion of job processing and printing that the server shall  
1466 keep job attributes and document data. The Printer may keep these  
1467 attributes and data longer than the value of the job-retention-  
1468 period attribute.

1469 IssueSSUE: There is some discussion about whether or not this  
1470 should be removed from the spec? Should this be deleted?

#### 1471 6.2.7 Job Production Attributes (Set by Client/End User)

1472 The client shall specify these attributes to affect the rendering,  
1473 production and finishing of the documents in the job. Similar  
1474 types of instructions may also be contained in the document to be  
1475 printed.

1476 If there is a conflict between the value of one of these  
1477 attributes, and a corresponding instruction in the document  
1478 (either implicit or explicit), the value of the attribute shall  
1479 take precedence over the document instruction.

1480 Job Production and Resource Attributes each address a similar set  
1481 of features but they have different uses.

1482 A job production attribute provides a client with a way to request  
1483 some feature that is not embedded within the document data. After  
1484 some program has merged the production attributes into the  
1485 document data After the information from these attributes has been  
1486 folded into the document data (possibly during a translation  
1487 process of the document data), these attributes are no longer  
1488 relevant and shall can be discarded from a job. Instead, the  
1489 resource attributes specify the resources needed to print the job  
1490 as modified by the job production attributes.

1491 Note: until companies that supply interpreters for PDL's, such as  
1492 PostScript and PCL allow a way to specify overrides for internal  
1493 job production instructions, a Printer may not be able to  
1494 implement these attributes for some PDL's.

1495 A job resource attribute tells a Printer what features the job  
1496 needs. A program that translates document data to a Printer's PDL,  
1497 and/or merges production attributes into the document data should  
1498 add job resource attributes to a job.

1499 For example, a job production attribute medium-select with the  
1500 value of "letter" requests that a job be printed on letter paper,  
1501 but gives no information about what resources the job needs. A  
1502 resource production attribute media-used with the values of  
1503 "letter" and "ledger" tell a Printer that the job needs letter and  
1504 ledger paper, but gives no information about which pages use each  
1505 medium.

1506

1507 The client may also specify document production-instruction  
1508 attributes in: Get-Attributes and GetJobs.

##### 1509 6.2.7.1 medium-select (type2Enum)

1510 This attribute identifies the medium that the Printer shall use  
1511 for all pages of the document regardless of what media are  
1512 specified within the document.

1513 The values for medium include medium-names, medium-sizes, input-  
 1514 trays and electronic forms so that one attribute specifies the  
 1515 media.

1516 Standard values are defined (taken from ISO DPA and the Printer  
 1517 MIB)

1518

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

1519

1520 The following standard values are defined for North American  
 1521 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

1522

1523 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

1524

1525 The following standard values are defined for the less commonly  
 1526 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-10-white	Specifies the ISO A10 white medium
iso-b0-white	Specifies the ISO B0 white medium
iso-b1-white	Specifies the ISO B1 white medium
iso-b2-white	Specifies the ISO B2 white medium
iso-b3-white	Specifies the ISO B3 white medium
iso-b6-white	Specifies the ISO B6 white medium
iso-b7-white	Specifies the ISO B7 white medium
iso-b8-white	Specifies the ISO B8 white medium
iso-b9-white	Specifies the ISO B9 white medium
iso-b10-white	Specifies the ISO B10 white medium
jis-b0-white	Specifies the JIS B0 white medium
jis-b1-white	Specifies the JIS B1 white medium
jis-b2-white	Specifies the JIS B2 white medium
jis-b3-white	Specifies the JIS B3 white medium
jis-b6-white	Specifies the JIS B6 white medium
jis-b7-white	Specifies the JIS B7 white medium
jis-b8-white	Specifies the JIS B8 white medium
jis-b9-white	Specifies the JIS B9 white medium
jis-b10-white	Specifies the JIS B10 white medium

1527

1528 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

1529

1530 6.2.7.2 number-up (positiveInteger)

1531 This attribute specifies the number of source page-images to  
 1532 impose upon a single side of an instance of a selected medium.

1533 In general, only certain numeric values are valid for this  
 1534 attribute, depending upon the Printer implementation to which the  
 1535 print-request is directed. Typical supported values are 2 and 4.  
 1536 If this attribute is unspecified or has a value of 1, then the  
 1537 Printer does not apply any number-up transformation to the pages.

1538 This attribute primarily controls the translation, scaling and  
 1539 rotation of page images, but a site may choose to add  
 1540 embellishments, such as borders to each logical page. If  
 1541 embellishments are added, especially for the number-up = 1 case,  
 1542 these are controlled through some other mechanism or attribute.  
 1543 The user expects that if number-up is absent or equal to 1 then no  
 1544 other imposition embellishments are added via this attribute.

1545 ~~ISSUE: should there be a separate attribute to control~~  
 1546 ~~embellishments, especially for the 1-up case?~~

1547 6.2.7.3 finishing (type2Enum)

1548 This attribute identifies the finishing operation that the Printer  
 1549 should apply to each copy of the printed document. Examples  
 1550 include stapling, saddle-stitching, hole-drilling, binding with  
 1551 tape, etc.

1552 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

1553

## 1554 6.2.7.4 sides (type2Enum)

1555 This attribute specifies whether the document should be printed in  
 1556 one of three ways: 1-sided (simplex), 2-sided-long-binding-edge  
 1557 (duplex), 2-sided-short-binding-edge (tumble).

1558 The standard values are: 1-sided, 2-sided-long-edge, 2-sided-  
 1559 short-edge.

## 1560 6.2.7.5 copies (positiveInteger)

1561 This attribute specifies the number of copies of the job to be  
 1562 printed. If this attribute is unspecified, its default value is 1  
 1563 copy.

## 1564 6.2.7.6 printer-resolution-select (positiveIntegerCross)

1565 This attribute specifies the resolution that the Printer should  
 1566 use.

1567 The syntax allows a single integer to specify the resolution or a  
 1568 pair of integers to specify the resolution when the x and y  
 1569 dimensions differ. When two integers are specified, the first is  
 1570 in the x direction, ie., in the direction of the shortest  
 1571 dimension of the medium, so that the value is independent of  
 1572 whether the printer feeds long edge or short edge first..

## 1573 6.2.7.7 print-quality (type2Enum)

1574 This attribute specifies the print quality that the Printer should  
1575 use.

1576 The standard values are:

1577	draft	Lowest quality available on the printer
1578	normal	Normal or intermediate quality on the printer
1579	high	Highest quality available on the printer

1580

1581

## 1582 6.2.7.8 page-select (positiveIntegerRange)

1583 This attribute specifies the pages in the document that the  
1584 Printer shall use. This attribute is unlikely to be useful for  
1585 jobs with more than one document or in Job Templates. If this  
1586 attribute is unspecified, then the Printer prints all pages in a  
1587 document.

## 1588 6.2.7.9 files-are-one-document (Boolean)

1589 This attribute is relevant only if a job consists of two or more  
1590 documents. It controls finishing operations, job-sheet placement,  
1591 and the order of documents when the copies attribute exceeds 1.

1592 If the files for the job are a and b and this attribute is true,  
1593 then files a and b are treated as a single document for finishing  
1594 operations. Also, there will be no slip sheets between files a and  
1595 b. If more than one copy is made, the ordering must be a, b, a,  
1596 b, .... The attribute files-are-interleaved is ignored.

1597 If the files for the job are a and b and this attribute is false  
1598 or unspecified, then each file is treated as a single document for  
1599 finishing operations. Also, a client may specify that a slip sheet  
1600 be between files a and b. If more than one copy is made, and the  
1601 attribute files-are-interleaved false or unspecified, the ordering  
1602 is a, a, b, b, .... If more than one copy is made, and the  
1603 attribute files-are-interleaved true, the ordering is a, b, a, b,  
1604 ....

## 1605 6.2.7.10 files-are-interleaved (Boolean)

1606 This attribute is used in conjunction with files-are-one-document  
1607 (q.v.).

1608 | ~~ISSUE: Should the files are one document and files are interleaved~~  
1609 | ~~be combined into a single enum attribute?~~

1610 6.2.8 Attributes for Conversion of Text Files (Set by Client/End  
1611 User)

1612 The client shall specify these attributes to control formatting  
1613 for text documents or HTML documents. If the client does not  
1614 specify any of these attributes, a Printer shall use its own  
1615 defaults.

1616       A client need not specify these attributes for other types of  
1617       documents, such as PostScript or PCL.

#### 1618   6.2.8.1 width (cardinalUnits)

1619       This attribute specifies the media width for the document in  
1620       characters.

#### 1621   6.2.8.2 length (cardinalUnits)

1622       This attribute specifies the media length for the document in  
1623       characters.

#### 1624   6.2.8.3 left-margin (cardinalUnits)

1625       This attribute specifies the left-margin for the document in  
1626       characters.

#### 1627   6.2.8.4 right-margin (cardinalUnits)

1628       This attribute specifies the right-margin for the document in  
1629       characters.

#### 1630   6.2.8.5 top-margin (cardinalUnits)

1631       This attribute specifies the top-margin for the document in lines.

#### 1632   6.2.8.6 bottom-margin (cardinalUnits)

1633       This attribute specifies the bottom-margin for the document in  
1634       lines.

#### 1635   6.2.8.7 repeated-tab-stops (cardinalUnits)

1636       This attribute specifies the tab stops for the document in  
1637       characters.

#### 1638   6.2.8.8 header-text (string)

1639       This attribute specifies the header text for the document.

#### 1640   6.2.8.9 footer-text (string)

1641       This attribute specifies the footer text for the document.

#### 1642   6.2.8.10 font-size (cardinalUnits)

1643       This attribute specifies the font-size in points for text in the  
1644       document. The value of this attribute affects the size of the  
1645       other text attributes.

1646       If this attribute is omitted, the Printer shall assume a value of  
1647       10. A value of 10 with a fixed pitch font, shall produce 12  
1648       characters per inch in the horizontal direction and with 6 lines  
1649       per inch in the vertical direction.

## 1650 6.2.8.11 number-pages (Boolean)

1651 This attribute specifies that the pages should be numbered in the  
1652 document.

1653 default-font (string) This attribute specifies the font to use for  
1654 all text in the document.

## 1655 6.2.8.12 default-code-set (type3Enum)

1656 This attribute specifies the code-set in which the document is  
1657 encoded.

## 1658 6.2.8.13 content-orientation (type2Enum)

1659 This attribute specifies the orientation of the document.

1660 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.

1661

1662 6.2.9 Job Resource Attributes (Set by the program that produces or  
1663 senses the PDL)

1664 A program described below shall add these attributes, which  
1665 describe the resources needed to print the job.

1666 A Printer may use these attributes to validate and schedule the  
1667 print-job without interpreting the contents of the document. This  
1668 provides the opportunity for a Printer to support a broad set of

document formats yet still support fast efficient scheduling and validation of each job.

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

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

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

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

If any of these attributes is unspecified, the Printer shall assume that the all resources required by the document of the type specified by the missing attributes are ready, ie., are available to the Printer and/or output device without human intervention. These attributes may be unspecified if the translation program fails to provides such values, or if no translation occurs (e.g. the document is a PostScript document).

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

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

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

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

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

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

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

The syntax is for type2Format:

name [ "/" version ]

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

Note: The version component is optional. ~~ISSUE: do we want the version to be optional?~~

## 1711 6.2.9.2 fonts-used (1#string)

1712 This attribute identifies the font resources used in the  
1713 document(s) in the job.

## 1714 6.2.9.3 code-sets-used (1#type3Enum)

1715 This attribute identifies the code-sets used in the document(s) in  
1716 the Job. This attribute is relevant only for files that are not in  
1717 ASCII, such as text files and possibly PCL files. PostScript files  
1718 are always ASCII. Normally there is at most 1 code-set.

1719 Standard values are defined in the section specifying the default-  
1720 code-set attribute.

## 1721 6.2.9.4 media-used (1#type2Enum)

1722 This attribute identifies the media, media-sizes, input-trays or  
1723 electronic forms needed to print the document(s) in the job.

1724 Standard values for this attribute are defined in the section  
1725 specifying the medium-select attribute.

1726  
1727 6.2.9.5 sides-used (type2Enum)

1728 This attribute specifies whether a job needs 1-sided, 2-sided-  
1729 long-binding-edge, or 2-sided-short-binding-edge printing.

1730 Standard values for this attribute are defined in the section  
1731 specifying the sides attribute.

## 1732 6.2.9.6 print-quality-used (type2Enum)

1733 This attribute specifies what print quality the job needs.

1734 Standard values for this attribute are defined in the section  
1735 specifying the print-quality attribute.

## 1736 6.2.9.7 finishing-used (type2Enum)

1737 This attribute specifies what finishing the job needs.

1738 Standard values for this attribute are defined in the section  
1739 specifying the finishing attribute.

## 1740 6.2.9.8 printer-resolution-used (positiveIntegerCrossState)

1741 This attribute specifies what resolution the job needs.

## 1742 6.2.9.9 total-job-octets (positiveInteger)

1743 This attribute specifies the total size of the job in octets. This  
1744 attribute is the first of three that a translation program can use  
1745 to specify the size of a job.

## 1746 6.2.9.10 job-impression-count (positiveInteger)

1747 This attribute specifies the total size of the job in impressions.

## 1748 6.2.9.11 job-media-sheet-count (positiveInteger)

1749 This attribute specifies the total size of the job in media-  
1750 sheets.

1751 6.2.9.12 job-intervening-jobs (positiveInteger)

1752 This attribute indicates the number of jobs that are "ahead" of  
1753 this job in the current scheduled order. For efficiency, it is  
1754 only necessary to calculate this value when an operation if  
1755 performed that requests this attribute.

## 1756 6.2.10 Number of Documents (Set by Client)

1757 This group contains a single attribute which specifies the number  
1758 of documents in the job.

1759 The client shall specify this attribute in Print and may specify  
1760 this attribute in: Get-Attributes and Get-Jobs.

## 1761 6.2.10.1 number-of-documents (positiveInteger)

1762 This attribute specifies the number of documents in the job. Each  
1763 document shall contain its own set of document content attributes  
1764 described below.

## 1765 6.2.11 Document Data (Set by a Client/End User)

1766 This group of attributes describes the document data for the job.  
1767 These attributes also include the document data or reference it.

1768 All job attributes in other sections of this document occur only  
1769 once per job and apply to all documents in a job.

1770 The client may specify document-data attributes in Print. The  
1771 client must specify either the document-URL or document-content in  
1772 Print.

1773 Except for document-content, the client may specify document-data  
1774 attributes in: Get-Attributes, and Get-Jobs.

## 1775 6.2.11.1 document-format (type2Format)

1776 This attribute identifies the document format of this document.

1777 If the client does not specify this attribute, then the Printer  
1778 shall attempt to determine the format in order to decide if the  
1779 document data needs to be translated. The version component is  
1780 optional.

1781 ~~ISSUE: do we want the version to be optional?~~



## 1782 6.2.11.2 document-name (name)

1783 This attribute contains the name of the document used by the  
1784 client to initially identify the document.

## 1785 6.2.11.3 document-URL (name)

1786 This attribute contains the URL of the document if the client  
1787 specified the document with a URL.

1788 If this attribute is specified, then document-content shall be  
1789 unspecified.

## 1790 6.2.11.4 document-content (octetString)

1791 This attribute contains the actual contents of the document.

1792 If this attribute is specified, then document-URL shall be  
1793 unspecified.

1794 This attribute shall be used during the transmission of the Print  
1795 operation over a network. A Printer shall save the document data  
1796 to a file and reference it with the document-URL or document-path  
1797 attribute. A Get-Attribute or Get-Jobs operation shall always find  
1798 that this attribute is unspecified.

## 1799 6.3 Operation Attributes (Set by Client)

1800 NOTE: These attributes have just been introduced and they are not  
1801 as stable as the attributes in the other sections. Some work is  
1802 still needed to show the relationship between these attributes,  
1803 job attributes, printer attributes, and authentication and  
1804 authorization.

1805 The client shall set these attributes and associate them with an  
1806 operation rather than an object.

1807 It is intended that a client program rather than an end-user has  
1808 control over the setting of these values so that they cannot be  
1809 easily forged.

## 1810 6.3.1 operation-locale (type3Locale)

1811 This attribute identifies the locale of the client. The Printer  
1812 uses this attribute to determine the locale of messages in the  
1813 result of the operation or in errors returned by the operation.

1814 The standard values are defined in the section on the job-locale  
1815 attribute.

1816 If an operation does not specify this attribute, the Printer shall  
1817 assume that the operation has the same locale as the Printer.

1818 6.3.2 operation-notification-address ( url~~name~~ )

1819 This attribute identifies the both the address and mechanism for  
1820 delivery of events . If the URL has a "mailto:" scheme, then email  
1821 is used and the rest of the URL is used as the email-address . If

1822 the URL has a "http:" scheme, then an HTTP APPEND method is used  
1823 to add HTML formatted events to the end of a specified HTML file.  
1824 of the client. The Printer uses this attribute to determine the  
1825 email address for any notifications that occur in the Printer.

1826 ~~ISSUE: can this address be determined from the next two~~  
1827 ~~attributes: operation user name and operation host name?~~

### 1828 6.3.3 operation-user-name (name)

1829 This attribute identifies the most authenticated end user name  
1830 that the client can supply. This name identifies the end user  
1831 performing the operation.

1832 This value shall be set by the system rather than the end-user in  
1833 order to minimize the chance of forgery.

### 1834 6.3.4 operation-host-name (name)

1835 This attribute identifies the most authenticated host name that  
1836 the client can supply. This name identifies the host from which  
1837 the operation comes.

1838 This value shall be set by the system rather than the end-user in  
1839 order to minimize the chance of forgery.

## 1840 6.4 Printer Attributes (Set by the Administrator)

1841 A printer object may be realized in either a Print Server or  
1842 Output Device. Note: How these attribute are set by an  
1843 Administrator is outside the scope of this specification.

1844 A Printer Object in an Output Device contains a set of printer  
1845 object attributes that represent an Output Device capable of  
1846 rendering a document in visible form. Examples include electronic  
1847 and electro-mechanical printers such as laser printers, ink-jet  
1848 printers, and various kinds of impact printers, but may include  
1849 other types of output devices such as microfiche imagers and  
1850 plotters as well.

1851 A Printer Object in a Print Server that supplies queuing,  
1852 spooling, and scheduling for an Output device that does not queue  
1853 or spool.

1854 A Printer Object in a Print Server contains a set of printer  
1855 object attributes that are the union of the Printer objects in the  
1856 downstream Output Devices. This object extends the capabilities  
1857 of an Output Device. For example, an administrator might define a  
1858 single Print Server to represent all of the Output Devices of the  
1859 same type and capability in a single location, associated with a  
1860 particular server. A end user would normally send a print-job to  
1861 a Print Server, and allow the Print Server to assign the job to a  
1862 particular Output Device based on the relative load and  
1863 availability of the printers under its control, thus providing a  
1864 load balancing service. However, nothing precludes an  
1865 administrator from configuring a print system so that a end user  
1866 can send a print-job directly to an Output Device.

1867 A Print Server, in the most common case, controls exactly one  
 1868 downstream Output Device. The Print Server's Printer object has  
 1869 attributes whose values are the same as those of the Printer  
 1870 object in the downstream Output Device.

1871 The attributes defined in this section provide information about  
 1872 a particular Printer.

1873 6.4.1 printer-name (name)

1874 This attribute uniquely identifies the printer on its host.

1875 6.4.2 printer-location (string)

1876 This attribute identifies the location of this printer.

1877 6.4.3 printer-model (string)

1878 This attribute identifies the make and model of the printer.

1879 6.4.4 printer-types (type2Enum)

1880 This attribute identifies the marking technology of the printer.

1881 The standard value for this attribute are the descriptive names  
 1882 specified by ISO DPA which have corresponding enum symbolic and  
 1883 numeric values assigned by the Printer MIB (RFC 1759).. These  
 1884 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 ion-deposition	other photographic ion deposition
E-beam	E-beam
typesetter	typesetter

1885

1886 ~~ISSUE: Should they be from the printer MIB instead. In the printer~~  
 1887 ~~MIB hyphens do not exist. Instead the first letter after a hyphen~~  
 1888 ~~is upper case. THIS ISSUE IS CLOSED: We will use the xxx yyy zzz~~  
 1889 ~~format rather than the xxxYyyZzz format.~~

#### 1890 6.4.5 printer-state (type1Enum)

1891 This attribute identifies the current state of the printer. The  
 1892 protocol support all values for printer states, however a Printer  
 1893 shall only generate the printer states which are appropriate for  
 1894 the particular implementation.

1895 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- The server has scheduled a job on the  
to-printer 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.

1896

## 1897 6.4.6 printer-state-message (string)

1898 This attribute specifies a message that gives further information  
1899 about the current printer state. .

## 1900 6.4.7 message (string)

1901 This attribute provides a message from an operator, system  
1902 administrator or "intelligent" process to indicate to the end user  
1903 information or status of the printer, such as why it is  
1904 unavailable or when it is expected to be available. .

## 1905 6.4.8 locale (type3Locale)

1906 This attribute specifies the locale that the Printer operates in.

1907 The standard values are defined in the section on the job-locale  
1908 attribute.

## 1909 6.4.9 notification-events (#type2Enum)

1910 This attribute specifies the events on whose occurrence the  
1911 Printer should notify those addresses specified by the  
1912 notification-addresses attribute.

1913 If the attribute is unspecified or empty, the Printer does not  
1914 perform notification, though the Printer still checks the job's  
1915 notification-events attribute.

1916 In this attribute, job-problem and printer-problem have the same  
1917 meaning.

1918 The standard values are defined in the section on the job's  
1919 notification-events attribute.

## 1920 6.4.10 notification-addresses (#name)

1921 This attribute specifies the email addresses to which the Printer  
1922 should send messages when events specified by the notification-  
1923 events attribute occur.

1924 If the attribute is unspecified or empty, the Printer does not  
1925 perform notification, though the Printer still checks the job's  
1926 notification-events attribute.

## 1927 6.4.11 end-user-acl (#name)

1928 This attribute specifies the end users who are allowed to print on  
1929 the Printer.

1930 If the attribute is unspecified or empty, the Printer allows  
1931 anyone to print.

1932 ~~ISSUE: this does not fully solve the internet authorization~~  
1933 ~~problem because of authentication issues.~~

## 1934 6.4.12 maximum-printer-speed (positiveIntegerUnits)

1935 This attribute indicates the maximum printer speed of the Printer.  
1936 A job cannot control a Printer's speed, but a Printer Browser can  
1937 use printer speed as a criteria.

1938 The standard units are a type2Enum and are: ppm, ipm, spm, lpm,  
1939 cps.

## 1940 6.4.13 fonts-substitutions (#stringPair)

1941 This attribute specifies an appropriate substitute for a font that  
1942 is advertised as supported in the fonts-supported attribute, even  
1943 though the Printer doesn't actually have the font available.

1944 This attribute consists of a set of font pairs: a font name and  
1945 the font to use instead.

## 1946 6.4.14 fonts-supported (1#stringState)

1947 This attribute identifies the font resources supported by this  
1948 printer and indicates the state of readiness for each font.

1949 The standard names are defined in the section on default-font.

1950 Each item in the list contains the pair consisting of a font name  
1951 and a state indicating the font's readiness state.

## 1952 6.4.15 media-supported (1#nameState)

1953 This attribute identifies the media, media-sizes, input trays, and  
1954 electronic forms supported by this printer, and indicates the  
1955 state of readiness for each medium resource.

1956 There may be just two states: ready and needs-installing, or there  
1957 may be a third state: needs-purchasing.

1958 The standard names are defined in the section on the section on  
1959 the medium-select.

## 1960 6.4.16 document-formats-supported (1#type2FormatState)

1961 This attribute identifies the document-formats, including the  
1962 document-format-versions, supported by the Printer. This set  
1963 includes both the formats that are native to the Printer and  
1964 those formats that the Printer can translate to one that is

1965 native to the Printer. From the client's point of view, this set  
1966 contains all formats in which documents can be submitted to this  
1967 Printer.

1968 Proprietary document format identifiers, and versions are assigned  
1969 by the owners of those formats.

1970 The state of readiness for each format is also included, though  
1971 all formats should normally always be ready.

1972 6.4.17 numbers-up-supported (1#positiveIntegerState)

1973 This attribute identifies the number-up values supported by this  
1974 printer..

1975 The state of readiness for each number-up value is also included,  
1976 though all number-up conversions should always be ready.

1977 6.4.18 finishings-supported (#type2EnumState)

1978 This attribute identifies the finishing operations supported by  
1979 this Printer and states of readiness for each finishing.

1980 The standard finishing objects are defined in the section on the  
1981 finishing Job attribute.

1982 6.4.19 sides-supported (1#type2EnumState)

1983 This attribute indicates the values of the sides attribute  
1984 supported by this printer and the states of readiness of each  
1985 value.

1986 The standard values are defined in the section on the sides  
1987 attribute.

1988 6.4.20 print-qualities-supported (1#type2EnumState)

1989 This attribute indicates the values of the printer-quality  
1990 attribute supported by this printer and the states of readiness  
1991 for each print-quality value.

1992 The standard values are defined in the printer-quality attribute.

1993 6.4.21 printer-resolutions-supported (1#positiveIntegerCrossState)

1994 This attribute indicates the values of the printer-resolution-  
1995 select attribute supported by this printer and their states of  
1996 readiness.

1997 The state of readiness for each printer resolution is also  
1998 included, though normally all printer-resolutions should always be  
1999 ready.

2000 The syntax is discussed in the section on the printer-resolution-  
2001 select attribute.

## 2002 6.4.22 code-sets-supported (1#type3EnumState)

2003 This attribute indicates the values of the default-code-set  
2004 attribute supported by this printer and the states of readiness  
2005 for each code-set.

2006 The standard values are defined in the default-code-set attribute.

## 2007 6.4.23 off-peak-times-supported (#type3EnumState)

2008 This attribute indicates the values of the job-print-off-peak  
2009 attribute supported by this printer and the states of readiness  
2010 for each value.

2011 If this attribute is unspecified, then the Printer has no off-peak  
2012 periods.

2013 The standard values are defined in the section on the job-print-  
2014 off-peak attribute.

2015 Note: this document does not define how an administrator  
2016 associates the off-peak names with actual time periods.

## 2017 6.4.24 events-supported (#type2EnumState)

2018 This attribute indicates the values of the job and printer  
2019 notification-events attribute supported by this Printer and the  
2020 states of readiness for each value.

2021 If this attribute is unspecified, then the Printer does not  
2022 support notification.

2023 The standard values are defined in the section on the  
2024 notification-events attribute.

## 2025 6.4.25 locales-supported (1#type3LocaleState)

2026 This attribute indicates the values of the job-locale attribute  
2027 supported by this Printer and the states of readiness for each  
2028 value.

2029 The standard values are defined in the section on the job-locale  
2030 attribute.

## 2031 6.4.26 job-sheets-supported (#type3EnumState)

2032 This attribute identifies the job-sheet values supported by this  
2033 printer, and the state of readiness for each job-sheet.

2034 To allow no job sheets, the system administrator shall include the  
2035 value none as a value for this attribute. The client specifies  
2036 that there are no job sheets by using the value none as the value  
2037 of the job-sheets attribute.

2038 If the job-sheets attribute is not specified or contains a value  
2039 which the Printer does not support, then the server shall select  
2040 from among the values of this attribute. The server shall not



2041 select the value none unless it is the only value specified for  
2042 the job-sheets-supported attribute.

2043 NOTE - It is preferable for the server to produce some job  
2044 jobsheet, even if not the desired one, rather than produce none at  
2045 all or reject the job.

2046  
2047 6.4.27 maximum-copies (positiveInteger)

2048 This attribute indicates the maximum number of copies of a  
2049 document that can be rendered by this printer in a single print-  
2050 job.

2051 If the attribute is unspecified or has a value of 0, there is no  
2052 limit on the maximum number of copies for this Printer.

2053 6.4.28 maximum-job-octets (positiveInteger)

2054 This attribute indicates that the Printer shall accept a job only  
2055 if its size in octets is less than the value specified by this  
2056 attribute.

2057 If the attribute is unspecified or has a value of 0, there is no  
2058 limit on the size of a job in octets.

2059 6.4.29 maximum-impressions (positiveInteger)

2060 This attribute indicates that the Printer shall accept a job only  
2061 if its size in impression is less than the value specified by this  
2062 attribute.

2063 If the attribute is unspecified or has a value of 0, there is no  
2064 limit on the size of a job in impressions.

2065 6.4.30 maximum-media-sheets (positiveInteger)

2066 This attribute indicates that the Printer shall accept a job only  
2067 if its size in media-sheets is less than the value specified by  
2068 this attribute.

2069 If the attribute is unspecified or has a value of 0, there is no  
2070 limit on the size of a job in media-sheets.

2071 6.4.31 maximum-job-retention-period (deltaTime)

2072 This attribute indicates that when the Printer accepts a job, the  
2073 retention period must not exceed the value of this attribute.  
2074 Otherwise, the Printer sets the job's retention-period to the  
2075 value of this attribute.

2076 If this attribute is unspecified, then the Printer places no limit  
2077 on the retention time.

2078 | ~~ISSUE: Should this be deleted?~~

## 2079 6.4.32 maximum-end-user-priority (type1Enum)

2080 This attribute indicates that when the Printer accepts a job, the  
2081 job-priority must not exceed the value of this attribute.  
2082 Otherwise, the Printer sets the job's job-priority to the value of  
2083 this attribute.

2084 If this attribute is unspecified, then the Printer places no limit  
2085 on the job-priority time.

2086 The standard values are defined in the section on the job-priority  
2087 attribute.

2088 6.4.33 queued-job-count (positiveInteger)

2089 This attribute contains a count of the number of jobs that are  
2090 either pending and/or processing.

2091 6.4.34 scheduling-algorithm (type3Enum)

2092 This attribute indicates the current scheduling algorithm for this  
2093 Printer: "none", "shortest-job-first", "time-received", etc.

## 2094 6.5 Job Templates

2095 The attributes for a Job Template can be any of the Job object  
2096 attributes defined in the sections:

2097 Job Sheet Attributes  
2098 Notification Attributes  
2099 Job Scheduling Attributes  
2100 (except job-print-after)  
2101 Job Production Attributes  
2102 (except page-select)  
2103 Attributes for Conversion of Text Files  
2104  
2105

## 2105 6.6 Conformance

2106 A conforming implementation shall implement all operations,  
2107 objects and attributes defined in this document. IPP is explicitly  
2108 designed to be extensible. This means that in addition to the  
2109 attributes defined in this specification, specific implementation  
2110 instances may support not only the basic protocol as defined in  
2111 this specification, but might add vendor specific extensions.

2112 Also, for the core set of attributes listed in this specification,  
2113 it is not required that a conforming server support all (standard)  
2114 values of all supported attributes. For example, it is not  
2115 required that a printer implement all finishing methods indicated  
2116 by the standard values.

2117 The explicit requirement of the term "supported", with respect to  
2118 one of the attributes that deal with printer functions or  
2119 resources, is that the server shall recognize the attribute and  
2120 those values that are supported, and shall be able to respond to a  
2121 query about which values that printer does, in fact, support.

2122 Additional attributes can be proposed to be registered by going  
2123 through the type 2 enum process which will register their  
2124 specification after approval with IANA.

## 2125 7. Security Considerations

2126 This protocol does not identify any new authentication mechanisms.  
2127 The authentication mechanisms built into HTTP (such as SSL and  
2128 SHTTP) are recommended.

2129 This protocol does define a simple authorization mechanism by  
2130 introducing the "end-user-acl" attribute as part of the Printer  
2131 object. This ACL attribute is a multi-valued list of all of the  
2132 authenticated names of end-users. This protocol does not specify  
2133 what the domain is for names in this ACL attribute.

2134 | Issue~~SSUE~~: Will it always be possible for a Printer to obtain a  
2135 meaningful authenticated name that the Printer can match against  
2136 the end-user-acl, or will some other mechanism be necessary, such  
2137 as a password?

## 2138 8. References

- 2139 [1] Smith, R., Wright, F., Hastings, T., Zilles, S., and  
2140 Gyllenskog, J., "Printer MIB", RFC 1759, March 1995.  
2141
- 2142 [2] Berners-Lee, T, Fielding, R., and Nielsen, H., "Hypertext  
2143 Transfer Protocol - HTTP/1.0", RFC 1945, August 1995.  
2144
- 2145 [3] Crocker, D., "Standard for the Format of ARPA Internet Text  
2146 Messages", RFC 822, August 1982.  
2147
- 2148 [4] Postel, J., "Instructions to RFC Authors", RFC 1543, October  
2149 1993.  
2150
- 2151 [5] ISO/IEC 10175 Document Printing Application (DPA), Final,  
2152 June 1996.  
2153
- 2154 [6] Herriot, R. (editor), X/Open A Printing System  
2155 Interoperability Specification (PSIS), August 1995.  
2156
- 2157 [7] Kirk, M. (editor), POSIX System Administration - Part 4:  
2158 Printing Interfaces, POSIX 1387.4 D8, 1994.  
2159
- 2160 [8] Borenstein, N., and Freed, N., "MIME (Multi-purpose Internet  
2161 Mail Extensions) Part One: Mechanism for Specifying and  
2162 Describing the Format of Internet Message Bodies", RFC 1521,  
2163 September, 1993.  
2164
- 2165 [9] Braden, S., "Requirements for Internet Hosts - Application  
2166 and Support", FRC 1123, October, 1989,  
2167
- 2168 [10] McLaughlin, L. III, (editor), "Line Printer Daemon Protocol"  
2169 RFC 1179, August 1990.  
2170  
2171

## 2172 9. Author's Address

2173 Scott A. Isaacson

2174 Novell, Inc.  
2175 122 E 1700 S  
2176 Provo, UT 84606  
2177  
2178 Phone: 801-861-7366  
2179 Fax: 801-861-4025  
2180 EMail: scott\_isaacson@novell.com  
2181  
2182 Tom Hastings  
2183 Xerox Corporation  
2184 701 S. Aviation Blvd.  
2185 El Segundo, CA 90245  
2186  
2187 Phone: 310-333-6413  
2188 Fax: 310-333-5514  
2189 EMail: hasting@cp10.es.xerox.com  
2190  
2191 Robert Herriot  
2192 Sun Microsystems Inc.  
2193 2550 Garcia Ave., MPK-17  
2194 Mountain View, CA 94043  
2195  
2196 Phone: 415-786-8995  
2197 Fax: 415-786-7077  
2198 Email: robert.herriot@eng.sun.com  
2199  
2200 Roger deBry  
2201 HUC/003G  
2202 IBM Corporation  
2203 P.O. Box 1900  
2204 Boulder, CO 80301-9191  
2205  
2206 Phone: (303) 924-4080  
2207 Fax: (303) 924-9889  
2208 Email: debry@vnet.ibm.com  
2209  
2210 Other Contributors  
  
2211 Devon Taylor, Novell, Inc.  
2212 Mike MacKay, Novell, Inc.  
2213 Peter Zehler, Xerox, Corp.  
2214 Keith Carter, IBM Corporation  
2215 Carl-Uno Manros, Xerox, Corp.  
2216 <add the list of IPP attendees and participants>  
2217  
  
2218

10. Appendix A: Sample IPP Operations

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

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

10.1 Querying the printer

Client some.domain.com

----- >

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

GetAttributes IPP/1.0

Printer-state :

Sides-supported :

Media-supported :

Document-formats-supported :

<-----

http/1.0 201 "Created" (a response)

IPP/1.0 xxx "attribute list returned "

Printer-state : running

Sides-supported : 1-sided

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

Document-formats-supported : Postscript/2.0

10.2 Print Operation - with print data included

Client some.domain.com

----- >

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

Print IPP/1.0

Print-Job-Object Header

Job-name : My Job

Medium : iso-a4-white

Notification-events : Job-completion

Notification-address : joe@pc.domain.com

Document Header

Document-name : Letter to Mom

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

Document in Postscript level 2 format

<-----

http/1.0 200 "accepted"

2265 IPP/1.0 xxx "print job accepted and queued "  
 2266 Job-Identifier : some.domain.com/printer-1/0037  
 2267 Current-job-state : pending  
 2268 Printer-state : running  
 2269

### 2270 10.3 Print Operation - with no data included

2271 Client some.domain.com  
 2272  
 2273 ----- >  
 2274

2275 Post http://some.domain.com/printer-1 http/1.0  
 2276 Print IPP/1.0  
 2277 Print-Job-Object Header  
 2278 Job-name : My Job  
 2279 Medium : iso-a4-white  
 2280 Notification-events : Job-completion  
 2281 Notification-address : joe@some.domain.com  
 2282 Document Header  
 2283 Document-name : Letter to Mom  
 2284 Document-URL : joe@pc.domain.com/Docs/To-mom.ps  
 2285  
 2286 <-----

2287 http/1.0 200 "accepted"  
 2288 IPP/1.0 xxx "print job accepted and queued "  
 2289 Job-Identifier : some.domain.com/printer-1/0037  
 2290 Current-job-state : pending  
 2291 Printer-state : running  
 2292

### 2293 10.4 Querying the state of the job

2294 In this example, no attributes are specified, so all job  
 2295 attributes are returned .

2296 Client some.domain.com  
 2297  
 2298 ----- >

2299 Post http://some.domain.com/printer-1/0037 http/1.0  
 2300 GetAttributes IPP/1.0  
 2301  
 2302 <-----

2303 http/1.0 201 "Created" (a response)  
 2304 IPP/1.0 xxx "attribute list returned "  
 2305 Job-Name : My Job  
 2306 Job-Originator : Joe@some.domain.com  
 2307 Job-originating-host : pc.domain.com  
 2308 Notification-address : joe@pc.domain.com  
 2309 Job-locale : xx:xx:xx  
 2310 Current-job-status : printing  
 2311 Printer-assigned : printer-1  
 2312 Submission-time : 1214  
 2313 Media-sheets-completed : 2  
 2314

### 2315 10.5 Canceling a Job

2316 Client some.domain.com

```

2316 | ----- >
2317 | Post: http://some.domain.com/printer-1/0037
2318 | CancelJob IPP/1.0
2319 |
2320 | <-----
2321 |
2322 | http/1.0 200 "okay"
2323 | Current-job-state : terminating
2324 |
2325 | 10.6 Listing jobs on a Printer
2326 |
2327 | List jobs on print er-1, only return job sizes. Jobs are returned
2328 | in the order they are scheduled for printing. A Job-identifier
2329 | attribute preceeds the attributes returned for each job to delimit
2330 | job boundaries.
2331 |
2332 |
2333 | Client some. domain.com
2334 |
2335 | ----- >
2336 | Post http/1.0 some.domain.com/printer-1
2337 | GetJobs IPP/1.0
2338 | total-job-octets :
2339 |
2340 | <-----
2341 |
2342 | http/1.0 201 "Created" (a response)
2343 | IPP/1.0 xxx "created an attribute list "
2344 | Job-identifier : 0033
2345 | total-job-octets : 4567
2346 | Job-identifier : 0034
2347 | total-job-octets : 12345
2348 | Job-identifier : 0035
2349 | total-job-octets : 12356

```