

Proposed Internet-Draft S. Isaacson

~~Novell, Inc.~~

D. Taylor

Novell, Inc.

M. MacKay

Novell, Inc.

P. Zehler

Xerox Corporation

T. Hastings

Xerox Corporation

C. Manros

Xerox Corporation

R. Herriot

Sun Microsystems

November 11, 1996

~~LDPA - Lightweight Document Printing Application~~

Internet Printing Protocol

Version 0.98, November 11~~October 31~~, 1996

NOTE: Because this document has been extensively changed from its original form, it has many rough spots which will need further editing. At this time, the reader should read it for major concepts.

Status of this Memo

This document is a working version of a protocol specification. It will eventually become an Internet-Draft by following well defined IETF procedures. At that time, the following paragraphs must be included:

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) based on HTTP [still to be ratified by the IPP group]. ~~Lightweight Document Printing Application (LDPA)~~ protocol for the Internet.

This protocol is heavily influence by a subset of 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).

IPPLDPA is defined as a set of abstract data types and operations. The operations are implemented using a protocol that is HTTP based [still to be ratified by the IPP group]. ~~the Internet standard remote procedure call mechanisms defined in RFC 1831 (RPC: Remote Procedure Call Specification Version 2).~~

The IPPLDPA protocol initially covers only user operations on basic print service objects, but will cover management operation as soon as possible. ~~Authentication and some access control will be required for the CancelJob operation.~~ Additional access Control, Device Management, and Service Management will be added to the protocol as soon as possible~~are all outside the scope of this protocol.~~ Some monitoring and management is possible through ~~These areas are covered by other protocols. include methods and operations for service creation, management, and administration.~~ The SNMP Printer MIB [1] is an example of one of these. In the areas where there are no existing standards, many are being worked in other distributed service forums (management, security, etc.). As these services become more standardized, this document (and hence the protocol) can be updated to reflect the integration and relationships with these other standards.

87	Table of Contents	
88	1. Introduction	5
89	2. Distributed Printing.....	6
90	2.1 Components	7
91	2.2 Objects	7
92	2.2.1 Printer	8
93	2.2.2 Job.....	11
94	2.2.4 Initial Value Job	11
95	2.3 Object Relationships.....	12
96	2.4 Use of Naming and Directory Services.....	12
97	2.4.1 Status.....	13
98	2.4.2 Resolution.....	13
99	2.4.4 Color Supported.....	13
100	2.4.5 Maximum Speed.....	13
101	2.4.6 Maximum Speed Units.....	14
102	2.4.7 Plug and Play Device Id.....	14
103	2.4.8 Model	14
104	2.4.9 Manufacturer	14
105	2.4.10 Type	14
106	2.4.11 PDLs Supported.....	14
107	2.4.12 Sides Supported	14
108	2.5 OIDs	14
109	3. Internet Printing Model.....	14
110	3.1 Object Instances	15
111	3.2 Limits and Defaults	15
112	3.3 List Object Attribute Scoping Rules.....	15
113	4. Operations.....	16
114	4.1 Common Data Structures.....	16
115	6. User Operations	25
116	6.1 Print Operation.....	25
117	6.1.1 Print Argument.....	25
118	6.1.2 Print Result.....	27
119	6.2 Cancel Job Operation.....	28
120	6.2.1 Cancel Job Argument.....	28
121	6.2.2 Cancel Job Result.....	29
122	6.3.1 List Object Attributes Argument	29
123	6.3.1 List Object Attributes Argument	30
124	6.3.2 List Object Attributes Result	32
125	6.4 Modify Job Operation	33
126	6.4.1 Modify Job Argument	33
127	6.4.2 Modify Job Result	35
128	6.5 Resubmit Job Operation.....	35
129	6.5.1 Resubmit Job Argument.....	35
130	6.5.2 Resubmit Job Result.....	36
131	7. Object Attributes	37
132	7.1 Job Attributes	40
133	7.1.1 Job Informational Attributes	40
134	7.1.1.1 job-identifier	40
135	7.1.1.2 job-owner.....	41
136	7.1.1.3 TBD.....	43
137	7.1.1.4 job-name.....	43
138	7.1.2 Printer Selection Attributes	44
139	7.1.2.1 printer-name-requested	44
140	7.1.2.1 printer-name-requested	44

141	7.1.2.1 printer-name-requested	44
142	7.1.3 Job Status Attributes	44
143	7.1.3.1 current-job-state	45
144	7.1.3.2 printers-assigned	45
145	7.1.3.3 submission-time	46
146	7.1.3.5 job-message-from-administrator	47
147	7.1.3.6 completion-time	47
148	7.1.3.7 job-state-reasons	47
149	7.1.3.8 number-of-documents	48
150	7.1.3.9 job-submission-complete	48
151	7.1.3.9 job-submission-complete	48
152	7.1.4 Job Results Handling Attributes	49
153	7.1.4.1 job-sheets	49
154	7.1.5 Job Event Handling Attributes	49
155	7.1.5.1 notification-profile	50
156	7.1.6 Job Scheduling Instructions Attributes	50
157	7.1.6.1 job-hold	50
158	7.1.6.2 job-priority	51
159	7.1.6.4 job-retention-period	51
160	7.2 Document Attributes	52
161	7.2.1 Document Description Attributes	52
162	7.2.1.1 document-format	53
163	7.2.2 Document Production Instruction Attributes	54
164	7.2.2.2 default-medium	55
165	7.2.2.3 number-up	56
166	7.2.2.4 finishing	57
167	7.2.2.5 sides	57
168	7.2.2.6 copy-count	57
169	7.2.2.1 default-font	58
170	7.2.2.1 default-font	58
171	7.2.2.1 default-font	58
172	7.2.2.1 default-font	58
173	7.2.2.1 default-font	59
174	7.2.2 Document Production Instruction Attributes	59
175	7.2.2.1 default-font	59
176	7.2.2.1 default-font	59
177	7.2.2.1 default-font	59
178	7.2.2.1 default-font	59
179	7.2.2.1 default-font	59
180	7.2.2.1 default-font	59
181	7.2.2.1 default-font	60
182	7.2.2.1 default-font	60
183	7.2.2.1 default-font	60
184	7.2.2.1 default-font	60
185	7.2.2.1 default-font	60
186	7.2.2.1 default-font	60
187	7.2.2.1 default-font	60
188	7.2.3 Document Characteristics Attributes	61
189	7.2.3.1 fonts-used	62
190	7.2.3.1 fonts-used	62
191	7.2.3.1 fonts-used	63
192	7.2.3.2 media-used	63
193	7.2.3.2 media-used	63
194	7.2.3.2 media-used	63

195	7.2.3.2 media-used.....	63
196	7.2.3.2 media-used.....	64
197	7.2.3.2 media-used.....	64
198	7.2.3.2 media-used.....	64
199	7.2.3.2 media-used.....	64
200	7.2.3.2 media-used.....	64
201	7.2.4 Document Status Attributes.....	64
202	7.2.1 Document Description Attributes	64
203	7.2.1.1 document-format	64
204	7.2.1.2 document-content.....	65
205	7.3 Operation Attributes	65
206	7.4 Printer Attributes.....	66
207	7.4.1 printer-name	67
208	7.4.2 printer-state	67
209	7.4.2 printer-state	67
210	7.4.2 printer-state	68
211	7.4.3 message	68
212	7.4.6 fonts-supported	69
213	7.4.8 media-supported	69
214	7.4.11 document-formats-supported.....	70
215	7.4.12 numbers-up-supported.....	70
216	7.4.13 finishings-supported.....	70
217	7.4.14 sides-supported	70
218	7.4.15 job-sheets-supported.....	71
219	7.4.16 document-sheets-supported	71
220	7.4.17 maximum-copies-supported	71
221	7.4.18 notification-delivery-methods-supported	72
222	7.4.19 physical-printers-supported.....	72
223	7.4.20 Logical-printers-supported	72
224	7.4.21 events-supported	72
225	7.4.22 transfer-methods-supported.....	72
226	7.4.23 locales-supported	73
227	7.4.24 multiple-documents-supported	73
228	7.4.28 cancel-individual-document-supported.....	73
229	7.4.29 modify-individual-document-supported.....	73
230	7.5 Initial Value Job Attributes.....	73
231	7.6 Initial Value Document Attributes	73
232	7.7 Relationship to ISO/IEC 10175 Conformance Levels.....	73
233	8. Security Considerations.....	74
234	9. References.....	74
235	10. Author's Address.....	74
236	1. Introduction	

This document is a Proposed Internet-Draft. It is a specification for a protocol that can be used for distributed printing on the Internet. This protocol, Internet Printing Protocol (IPP) Lightweight Document Printing Application (LDPA), is heavily influence by based on the printing model introduced in the Document Printing Application (ISO/IEC 10175 DPA) standard. The DPA model describes a distributed printing service made up of cooperating networked entities. DPA also identifies user and administrative roles and operations. These ideas and concepts, when unified with other Internet protocols and services, realizes a distributed print service for the Internet.

~~This specification obsoletes RFC 1179 "Line Printer Daemon Protocol" [10]. "lpr" was designed a long time ago with line printers in mind. It does not fit with current page oriented printing technologies and most printer vendors have made their own proprietary extensions to "lpr" to try and get by with current needs. Unfortunately, these extensions are mutually incompatible, which means that many enhanced "lpr" implementations cannot interwork. In the X/Open PSIS project [6], these differences were documented and appropriate gateway solutions between DPA and each vendor (Digital, HP, IBM, SCO, Sun, and Xerox) described. LDPA introduces several improvements over RFC 1179:~~

- ~~— It uses Object Identifiers (OIDs) for interoperability and extensibility.~~
- ~~— It models the more complex, multiple content object page oriented languages and printers.~~
- ~~— It is based on a well known and well tested ISO standard.~~
- ~~— It supports desktop printing models as well as current market trends for distributed systems which are as diverse as they are spread throughout the globe.~~

This document assumes a distributed computing environment where print service users (clients, applications, drivers, etc.) cooperate and interact with print service providers (servers, printers, gateways, etc.) to realize the print service.

~~The actual protocol is yet HTTP based [still to be ratified by the IPP group]. consists of abstract data types representing the distributed print service and its components as well as operations which define and give semantics to the interaction between these components. The operations defined for this protocol are defined as RFC 1831 [2] compliant remote procedure calls. These operations are defined in sections 4, 5, and 6. The objects and their attributes are defined in section 7.~~

~~NOTE: The abstract data types could be defined using a syntax such as RPC language [2] or ASN.1 [8]. The actual encoding of the abstract data types could be realized using either XDR [3] or BER [9]. The actual syntax and encoding mechanism must be finalized during the standards process. Only the operations and attribute semantics are defined at this time. This document does not yet contain the small subset of syntax definitions for the proposed attributes. These can be added as a short appendix to this document. For the operations, this document currently shows operations defined in both ASN.1 and XDR. The full ASN.1 for ISO/IEC 10175 parts 1 and 3 (attributes, syntaxes, and operations) can be found at "ftp://ftp.pwg.org/pub/pwg/snmpmib/dpa". A proposal for the XDR version of the operation, attributes, and syntaxes can be found at "ftp://ftp.pwg.org/pub/pwg/netprint/ldpa".~~

2. Distributed Printing

The distributed printing service is defined as a collection of coordinating and cooperating entities in a distributed computing environment. The model assumed by this protocol is potentially an n-tier client/server model, but the model will be optimized for the normal cases of a 1-tier model(client to printer) and a 2-tier (client to server to printer). Users of the 1-tier and 2-tier models should not be aware of any extra complexity to support 3 or

~~more tiers. A service requester (client) makes service requests of service providers (servers). A given instance of a service provider (server) may in turn be a service requester (client) of some other service via its service providers (servers).~~

A client is able to access the services offered by a server by invoking one or more operations associated with the server. Each operation has associated arguments and results. The arguments provide additional data which is passed from the client to the server. The results return the status and outcome of the desired operation back to the client from the server.

2.1 Components

In the distributed printing service the entities or components are:

- One or more humans or agents acting on behalf of humans. Humans (or their agents) act in the role of Users, Operators, Managers, or Administrators.
- One or more clients. Clients are computer network nodes with which end users interact in order to manipulate the distributed print service. A client implements the IPPLDPA protocol.
- One or more print service providers (servers). An instance of a print service provider implements the IPPLDPA protocol by receiving and performs and respondings to IPPLDPA operations. ~~A given instance of a print service provider is also can be a "client" of yet another instance of a print service provider. Some of these clients use the IPP protocol; others use some other protocol. The last print service provider in the n-tier chain is a "client" to a print engine. There are several different types of print service providers which are defined later in this document. A print service provider can either be physical or logical (physical if it represents a physical printer or some other document production device, or logical if it represents one or more print service providers each of which may be logical or physical).~~

This LDPA specification only defines the operation used by Users. The operations used by Operators, Managers, and Administrators may be added if there is time~~are not within the scope of this standard.~~

2.2 Objects

To accomplish the action(s) requested via an operation, the print service provider manages and manipulates data objects. These are simply convenient collections of data that may represent other objects (real life or computer system) elsewhere. A client supplies arguments in the form of attribute values for some of these objects. A server informs the client of the status or outcome of an operation by also providing attribute values for the objects involved in the operations. 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.]

The objects which are relevant to this protocol are:

- Printer (contains server, queue and printer concepts)
- Job (contains job and document concepts, a document object may be added in the future)
- ~~- Document~~
- ~~- Initial Value Job~~
- ~~- Initial Value Document~~
- Job Template (contains Initial Value Job and Initial Value Document)

~~IPPLDPA~~ defines the operations that interact with and affect the real-life objects represented by the protocol's object definition.

2.2.1 Printer

[Note: it is not clear when 'Printer' refers to the hardware and when it refers to a software printer object.]

This document shall use the following terms:

- Output Device: printer hardware,
- Print Server: a program that augments one or more Output Devices.
- Printer: the software realization of a printer implemented in an Output Device or a Printer Server.

One of the most significant components within the distributed printing service is a Printer. ~~A Printer is an instance of a print service provider that provides access to both logical and Physical output devices.~~ A Printer object is a composition of some of the functionality that has traditionally been tied to other components within the printing system. A Printer can support the functionality of spooling, job management, device management, server, as well as more traditional device components.

A Printer can be in one of two authorization modes:

Public Access: The Printer is not restricted with any access control checks. The authorization allows anyone. The Printer uses a simple, name only (no password or credential) form of binding. [What does this last sentence mean?]

ISSUE: does Public access really mean no authentication. I would expect not. Otherwise, a person can cancel anyone's jobs.

Controlled Access: The Printer may have some restrictions based on some authentication and authorization scheme. The Printer uses some form of credential based binding. [What does this last sentence mean?]

A Printer object represents an instance of a print service provider which implements the ~~IPPLDPA~~ protocol. This allows the Printer to provide a common interface for all types of disparate and diverse physical devices or as well as a gateway interface for other non-Internet based printing systems.

To a print service user, a Printer has the "looks and feel" of a any typical physical printer. Jobs are submitted to and managed

at the Printer. The Printer can accept or reject submitted jobs based on job attributes which are sent along with the print job. The Printer tracks all jobs that have been submitted to it. The Printer can be modified to indicated a corresponding behavior change at the device level (either manually or automatically). In the Controlled Access mode, the Printer has an identity with a security or credential service.

~~The Printer can be a service provider for any of the following configurations:~~

An object that an end-user views as a Printer can be implemented with either of the following configurations.

- an Output Device which supports the IPP protocol. This Output Device may or may not have a job queue, and if it has a job queue it may be of very limited size. An administrator configures this Printer to receive jobs directly from a client.
- a Print Server and one or more downstream Output Devices. The Print Server supports incoming IPP protocol and uses either IPP or some other protocol to communicate with downstream Output Devices. A Print Server augments downstream Output Devices by supporting a large job queue. An administrator configures the downstream Output Devices to receive jobs from the Print Server. When there is only one downstream Output Device, the Printer object in the Print Server has the same values as those in the downstream Output Device. Note that if the downstream Output Device supports the IPP protocol, then its Printer Object and the Print Server's Printer Object are identical, attribute for attribute. When there are two or more downstream Output Devices, the Printer object in the Print Server has the union of values in the downstream Output Devices.

The print system shall also support gateways. Such gateways shall translate incoming IPP protocol to some other protocol and shall translate some other protocol to IPP protocol. Gateways shall pass operations through with minimal delay. Further description of them is beyond the scope of this document.

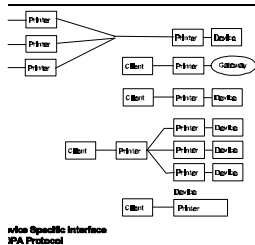
ISSUE: where does file conversion occur when there is no associated application, e.g. text or HTML to PostScript. To keep the model simple, the Printer defined above should only queue and print jobs. Conversion of text, HTML should occur within some other server which forwards the job to a Printer.

- ~~— The Printer can be a Physical Printer which represents and controls a print device using a device specific interface (possibly embedded).~~
- ~~— The Printer can be a Gateway to some other printing system (for example LPD).~~
- ~~— The Printer can be a Logical Printer which feeds other Printers (either Physical or Logical or Gateway).~~
- The following describes the allowed chains of printer types using RFC 822 syntax:
- normal usage:
- ipp-end-user [PrintServer] OutputDevice
- outbound gateway:

```

489 ipp-end-user Gateway AnotherPrintSystem
490
491 inbound gateway:
492 AnotherPrintSystem Gateway [PrintServer] OutputDevice
493
494
495 Although other arrangements of these printer types is possible,
496 they are not useful.
497
498
499 Figure 1 shows the some of the typical configurations of Printers:
500

```



~~The following questions may arise:~~

~~rinter Configurations~~

2. How can I achieve both "fan out" and "fan in" of client to Printers and from Printers to physical print devices?

This model currently contains a mechanism for defaulting in two places: during GUI initialization in the client and during printing of the job in the Output Device. There may be a need to associated defaulting (i.e. a job template) with a Printer Object so that an administrator can enforce value that are left undefined. This is currently an open issue that will not be resolved for version 1.0.

For "fan out", an administrator ~~may~~ sets up a ~~Logical Printer Print Server to which can~~ fan out to one or more downstream Output Devices ~~Printers~~ for load balancing and reliability.

3. How can a Printer be configured to service multiple spooling ~~Logical~~ Printers?

The system administrator establishes whether an end-user has direct access to an Output Device or indirect access via a Print Server. which Logical and Physical printers spool and which do not. If multiple ~~spooling Logical Print Servers Printers~~ are used to feed the same Output Device Printer, their schedulers are not coordinated. Although this configuration is possible, it is not recommended.

2.2.2 Job

A Job object is used to model a job. A job can consist of one ~~or more~~ documents. There are certain job attributes that pertain to the running and scheduling of the entire job (all documents). There are other job attributes that define global behavior or defaults for all contained documents~~jobs~~.

Still other job attributes pertain to each document within a job. There are no separate document objects, no attributes that pertain to one document in a job but not to others, except for the attribute that specifies the location of the documents.

In future versions, jobs will be able to contain more than one document, and documents will become separate object with attributes that override corresponding job attributes.

~~2.2.3 Document~~

~~A Document object is used to model a document. There are attributes that describe the contents of the document as well as the processing and handling of the document.~~

2.2.4 ~~Initial Value~~ Job Template

An ~~Initial Value~~ Job Template object is used to model job defaults. These are essentially job attributes that a client uses to initialize a newly created job object. ~~are used as default attributes for each job that is submitted.~~

~~2.2.5 Initial Value Document~~

~~An Initial Value Document is used to model document defaults.
These are essentially document attributes that are used as default
attributes for each document that is submitted as part of a job.~~

2.3 Object Relationships

Instances of objects within the system have relationships which must be maintained persistently along with the persistent storage of the objects themselves.

An instance of a print service provider is a Printer. The Printer is represented via a Printer object. A Printer can contain zero, one, or more Job objects. A Job ~~object~~_{object} contains one or more Documents~~s~~_{objects}. The following relationships are examples:

- "Document object D1 belongs to Job object J1", or
- "Job object J1 belongs to Printer object P1".

2.4 Use of Naming and Directory Services

[There is more work that need to be done to define the name service in an HTTP context. A name is a URL and is probably resolved at Print Servers and Output Devices.]

Any distributed service uses some sort of naming and/or directory service for (X.500, NDS, DCE naming, DNS). It is outside the scope of this protocol to define which name service to use or what the protocol is for using that name service, but the following discussion helps to clarify how the name service is intended to be used.

To distributed printing system, the instances of print service providers are represented by objects of type Printer. These same instances are also registered to the name/directory service. There is one entry in the name service for each Printer.

That is, instances of print service providers are represented to ~~IPPLDPA~~ as Printer objects. These objects represent their real-life counterparts, the print service provider (software, hardware, or firmware). However, for directory lookup, there is an entry in the naming service that also represents the Printer.

It is important to remember that a Printer object represents the current status and configuration information of a certain print service provider. The Printer object contains attributes and values that describe the characteristics and capabilities of the ~~logical or physical~~ print device. However, a few of the most important attributes from the Printer object are duplicated in the entry in the directory. These attributes are used for filtered directory lookups. The results of these searches enable a user to select an appropriate printer. It is the responsibility of the Printer itself to keep these attributes consistent and accurate. This requirement frees the directory or some directory agent from continually polling registered entities for configuration changes.

The following attributes are in the directory entry:

Fully Distinguished Name (the name within the directory's name space)

667 Description
 668 Location
 669 Owner
 670 Address
 671 Status
 672 Resolution
 673 Color Supported
 674 Maximum Speed
 675 Maximum Speed Units
 676 Device Id
 677 Model
 678 Manufacturer
 679 Type
 680 PDLs Supported
 681 Sides Supported
 682

683 The final set of name service entry attributes needs to be
 684 finalized and rationalized with the PSIS name service
 685 recommendations [6] as well as implementation experience.
 686

687 2.4.1 Status

688
 689
 690 [Such a dynamic value seem like it could be a problem in some
 691 name service entries.]

692 The printer status field in the directory entry is really a
 693 "summary" attribute of the true printer state. The following
 694 mapping takes place between the Printer Status attribute in the
 695 directory entry and the printer-state attribute in the Printer
 696 object:
 697

698 "Not Connected"
 699 STATE_NOT_CONNECTED
 700 STATE_PAUSED_NOT_CONNECTED
 701 "Shutdown"
 702 STATE_SHUTDOWN
 703 "Active"
 704 STATE_IDLE
 705 STATE_PAUSED
 706 STATE_PRINTING
 707 "Stopped"
 708 STATE_STOPPED
 709 STATE_PAUSED_STOPPED
 710

711 Even though the Printer may not be up and running, the directory
 712 entry still exists in the directory. In this case, the directory
 713 entry represents the fact that it may begin running at some future
 714 time.
 715

716 2.4.2 Resolution

717
 718 This is a single valued, maximum resolution in either the
 719 horizontal or vertical direction of the print device in dpi.
 720

721 2.4.4 Color Supported

722
 723 This is a BOOLEAN for either yes, color printing is supported, or
 724 no color printing is not supported.
 725

726 2.4.5 Maximum Speed

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

2.4.6 Maximum Speed Units

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

2.4.7 Plug and Play Device Id

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

2.4.8 Model

This is a simple text string defined by the manufacturer.

2.4.9 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.

2.4.10 Type

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

2.4.11 PDLs Supported

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

2.4.12 Sides Supported

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

2.5 OIDs

[OIDs do not belong in this model until we establish the protocol]

~~This protocol makes use of Object Identifiers (OIDs). All OIDs used in this protocol are defined encoded using the OBJECT IDENTIFIER ASN.1 syntax and the BER encoding of OBJECT IDENTIFIER.~~

~~This specification does not introduce any new OIDs. The following rules are used:~~

~~— Since LDPA is a small subset of DPA, for all attributes and values which are OIDs and are defined as within the scope of this specification, the OIDs from DPA will be used.~~

~~— For any extensions to this specification that fall within DPA operations or semantics, OIDs from DPA will be used.~~

~~— For any vendor specific extensions, OIDs from the appropriate enterprise arcs in the OID tree will be used.~~

3. Internet Printing Model

3.1 Object Instances

All instances of all objects have an identifier attribute that makes them unique so that they can be unambiguously referenced. In the object-oriented model, these are the globally unique object references which are created by factories or constructors.

The following objects have the following mandatory identifier attributes:

Object	Identifier	Containing Object
Printer	printer-name	None
Job	job-identifier	Printer
Document	document-sequence-number	Job
Initial Value Job	TBD	Printer
Job Template	job-template-name	None
Initial Value	TBD	Printer
Document		

3.2 Limits and Defaults

This ~~IPPLDPA~~ specification does not include any mechanism for specifying for enforcing "limits" or any other kinds constraints. However, defaults are achieved through the implementation of Job Template. ~~Initial Value Job and Initial Value Document objects.~~

3.3 ~~List Object Attribute Scoping Rules~~

~~The LIST OBJECT ATTRIBUTES operation is used for various reasons. The first of which, is to list contained jobs under a given Printer. Listing jobs works in this manner, according to the designator in the LOA (LIST OBJECT ATTRIBUTES) request:~~

- ~~1. LIST_OF_ORDERED_JOBS~~
~~—Lists scheduled jobs. Does not include retained jobs.~~
- ~~2. List objects (job class) with specified instances. Includes retained jobs.~~
~~—Lists specified jobs. If a job is not found as a current job, LOA looks for it as a retained job.~~
- ~~3. List objects (job class) without specified instances. Includes retained jobs; retained jobs are listed after current jobs.~~
~~—If the client is bound to a printer agent, lists all jobs for that printer agent.~~
~~—If only retained jobs are desired, the retained job state may be specified in a filter.~~

~~The second reason that the LIST OBJECT ATTRIBUTES operation is used is to query the database to find out about contained object relationships such as "What are the initial value objects for a given Printer?". The rules for these types of operations are:~~

- ~~1. List objects without specified instances.~~
~~—Lists all contained objects~~

4. Operations

~~IPPLDPA~~ defines the following⁷ end user operations:

The following symbols are used in the tables below:

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

Operation	Print Server	Output Device
Bind		
Unbind		
- Print	PF	P
- Cancel Job	PF	P
- <u>Get Attributes</u>	PF	P
- List Object Attributes		
- <u>Get Jobs</u>	PF	P

Lower priority (version 2) end user operations are:

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

Management operations are (we want these in version 1.0):

Operation	Print Server	Output Device
- <u>Clean Queue</u>	PF	UA
- <u>Disallow Queuing</u>	P	UA
- <u>Allow Queuing</u>	P	UA
- <u>Pause Printing</u>	P	P
- <u>Resume Printing</u>	P	P
- <u>Promote Job</u>	PF	UA
- <u>Shutdown Printer</u>	P	P
- <u>Startup Printer</u>	P	P
- <u>Create Printer</u>	P	U
- <u>Delete Printer</u>	P	U
- <u>Set Attribute</u>	P	P
- <u>Get Local Attributes</u>	P	P

4.1 Common Data Structures

This section describes the common data structures that are used by two or more operations.

4.1.1 ~~XDR~~

~~/*~~
~~// Note: Text is stored in XDR structures in Unicode, to~~
~~eliminate~~
~~// problems in comparing disparate forms of text.~~


```

883     // Unicode characters are always kept in low high byte order
884     // structures.
885     //
886     // The Text structure is defined as opaque, for efficiency
887     // in marshalling / unmarshalling operations.
888     // This means that the array item count contains the number
889     // of bytes, rather than the number of 16-bit characters.
890     */
891     //
892     typedef opaque Text<>;
893
894
895     /* ----- job identifier ----- */
896     struct PrtContainedObjectId {
897         Text printerName;
898         nuint32 localIdentifier;
899     };
900     typedef PrtContainedObjectId PrtContainedObjectIdSet<>;
901
902     /* ----- document identifier ----- */
903     struct DocumentIdentifier {
904         PrtContainedObjectId jobIdIdentifier;
905         nuint32 documentNumber;
906     };
907
908
909     /*
910     // Often times it is necessary for an object to have an
911     // attribute whose value is the "identifier" of another object.
912     // These attributes used an attribute syntax as defined below
913     */
914
915     enum ObjectIdentificationEnum {
916         OBJ_ID_PRT_CONTAINED_OBJ_ID = 0,
917         OBJ_ID_DOCUMENT_IDENTIFIER = 1,
918         OBJ_ID_OBJECT_IDENTIFIER = 2,
919         OBJ_ID_OBJECT_NAME = 3,
920         OBJ_ID_NAME_OR_OID = 4,
921         OBJ_ID_SIMPLE_NAME = 5,
922         OBJ_ID_PRT_CONFIG_OBJ_ID = 6
923     };
924     typedef enum ObjectIdentificationEnum ObjectIdentificationEnum;
925
926     struct ObjectIdentification {
927         ObjectIdentificationEnum designator;
928         union {
929             PrtContainedObjectId prtContainedObjectId;
930             DocumentIdentifier documentIdentifier;
931             ObjectIdentifier objectIdentifier;
932             DistinguishedNameString objectName;
933             NameOrOid nameOrOid;
934             Text simpleName;
935             PrtConfigObjectId prtConfigObjectId;
936         } ObjectIdentification_u;
937     };
938     typedef struct ObjectIdentification ObjectIdentification;
939
940     typedef AttributeValue AttributeValueSet<>;
941
942     /*

```

```

943  /** NOTE:
944  /** Sending an empty sequence for values allows an attribute
945  /** to be set as if it was not specified. This is primarily
946  /** for use in the modify function.
947  */
948
949  /* attribute */
950  struct Attribute {
951      ObjectIdentifier attributeId;
952      AttributeValueSet valueSet;
953      nuint32 qualifier;
954  };
955
956  typedef Attribute AttributeSet<>;
957
958  typedef Attribute CommonArguments<>;
959
960  enum NameOrOidEnum {
961      NAME_OR_OID_NONE, /* 0 */
962      NAME_OR_OID_GLOBAL, /* 1 */
963      NAME_OR_OID_LOCAL, /* 2 */
964  };
965
966  union NameOrOid switch(NameOrOidEnum designator) {
967      case NAME_OR_OID_NONE:
968          void;
969      case NAME_OR_OID_GLOBAL:
970          ObjectIdentifier globalForm;
971      case NAME_OR_OID_LOCAL:
972          Text localForm;
973  };
974
975  /* distinguishedNameString 9.1.5.7 */
976  struct DistinguishedNameString {
977      Text name;
978      NameOrOid *syntaxOptionPtr;
979  };
980
981  enum QualifiedNameEnum {
982      QUALIFIED_NAME_NONE,
983      QUALIFIED_NAME_SIMPLE,
984      QUALIFIED_NAME_OTHER
985  };
986
987  struct OtherName {
988      Text object;
989      Text otherOption;
990  };
991
992  union QualifiedName switch (QualifiedNameEnum designator) {
993      case QUALIFIED_NAME_NONE:
994          void;
995      case QUALIFIED_NAME_SIMPLE:
996          Text simpleName;
997      case QUALIFIED_NAME_OTHER:
998          OtherName otherName;
999  };
1000
1001  typedef QualifiedName QualifiedNameSet<>;
1002

```

```

1003 --- typedef DistinguishedNameString DistinguishedNameStrSeq<>;
1004 ---
1005 --- /*
1006 --- // Note: The value syntax for time attributes is
1007 --- // implemented as Cardinal.
1008 --- */
1009
1010
1011 4.1.2 ASN.1
1012
1013
1014 --- The following constants are used in later ASN.1 data types
1015 ---
1016 --- ub integer = 2147483647 biggest int = 2**31-1
1017 --- ub message string = 4095
1018 --- ub name string = 255
1019 --- ub octet string = 255
1020 ---
1021
1022 SimpleName ::= CHOICE {
1023 --- iso 646 irv [0] VisibleString(SIZE(0..ub name string)),
1024 --- ecitt t 61 [1] T61String(SIZE(0..ub name string)),
1025 --- iso latin1 [2] Latin1String(SIZE(0..ub name string)),
1026 --- iso ucs 2 [3] UCS2Level2String(SIZE(0..ub name string))}
1027
1028 AttributeId ::= OBJECT IDENTIFIER
1029
1030 Attribute ::= SEQUENCE {
1031 --- attribute id [0] AttributeId,
1032 --- attribute values [1] SET OF ANY --- DEFINED BY attribute id
1033 }
1034
1035 CommonArguments ::= SET OF Attribute
1036
1037 JobIdentifier ::= PrintableString (SIZE (1..255))
1038
1039 Message ::= CHOICE {
1040 --- iso 646 irv [0] VisibleString(SIZE(0..ub message string)),
1041 --- ecitt t 61 [1] T61String(SIZE(0..ub message string))
1042 --- iso latin1 [2] Latin1String(SIZE(0..ub message string)),
1043 --- iso ucs 2 [3] UCS2Level2String(SIZE(0..ub message string))}
1044
1045 PositiveInteger ::= INTEGER (1..ub integer)
1046
1047 DeltaTime ::= INTEGER (0..ub integer)
1048
1049 Cardinal ::= INTEGER (0..ub integer)
1050
1051 NameOrOid ::= CHOICE {
1052 --- global form [0] OBJECT IDENTIFIER,
1053 --- local form [1] SimpleName}
1054
1055 DistinguishedNameString ::= SEQUENCE {
1056 --- name [0] Text,
1057 --- name syntax [1] NameOrOid OPTIONAL }
1058
1059 Global Name
1060 FROM ISO STANDARD 9541 FONT RESOURCE
1061 { iso(1) standard(0) 9541 2 1 }
1062

```

```

1063 FontReference ::= CHOICE {
1064 simple font name [0] SimpleName,
1065 iso 9541 font name [1] Global Name }
1066
1067 AttributeValueAssertion ::= SEQUENCE {
1068 attribute id [0] AttributeId,
1069 attribute values [1] SET OF ANY DEFINED BY attribute id
1070
1071
1072 GeneralizedTime ::= from ISO 8824
1073
1074
1075 ErrorMessage ::= SEQUENCE {
1076 data [0] CHOICE {
1077 iso 646 irv [0] VisibleString(SIZE(0..ub message string)),
1078 ecitt t 61 [1] T61String(SIZE(0..ub message string)),
1079 iso latin 1 [2] Latin1String(SIZE(0..ub message string)),
1080 iso ucs 2 [3]
1081 UCS2Level2String(SIZE(0..ub message string)),
1082 other code set [4] OCTET STRING(SIZE(0..ub message string))
1083 },
1084
1085 4.2 Errors
1086
1087 This section identifies each of the individual error that might be
1088 returned in any of the operation results.
1089
1090 4.2.1 ASN.1
1091
1092 AccessProblem ::= CHOICE {
1093 standard problem ENUMERATED {
1094 inappropriate object class (1),
1095 insufficient access rights (2),
1096 cannot interrupt job (3),
1097 inappropriate object state (4) },
1098 extended problem OBJECT IDENTIFIER }
1099
1100 AccessErrorSequence ::= SEQUENCE OF SEQUENCE {
1101 object identification [0] ObjectIdentification,
1102 problem [1] AccessProblem,
1103 error message [2] ErrorMessage }
1104
1105 AttributeProblem ::= CHOICE {
1106 standard problem ENUMERATED {
1107 invalid attribute syntax (2),
1108 undefined attribute type (3),
1109 inappropriate matching (4),
1110 constraint violation (5),
1111 unsupported attribute type (6),
1112 illegal modification (7),
1113 inconsistent with other attributes (8),
1114 undefined attribute value (9),
1115 unsupported attribute value (10),
1116 invalid non compulsory attribute modification (11),
1117 per job attribute inadmissible (12),
1118 not multi valued (13),
1119 mandatory attribute omitted (14),
1120 attribute illegal for object class (15) },
1121 extended problem OBJECT IDENTIFIER }
1122

```

```

1123 AttributeErrorSequence ::= SEQUENCE {
1124 object identification [0] ObjectIdentification OPTIONAL,
1125 problems [1] SEQUENCE OF SEQUENCE {
1126 problem [0] AttributeProblem,
1127 attribute [1] Attribute,
1128 error message [2] ErrorMessage } }
1129 ---
1130 DocumentAccessProblem ::= CHOICE {
1131 standard problem ENUMERATED {
1132 document not available (1),
1133 referent modified (2),
1134 access denied (3),
1135 unknown document (4),
1136 no documents in job (5) },
1137 extended problem OBJECT IDENTIFIER }
1138 ---
1139 DocumentAccessErrorSequence ::= SEQUENCE {
1140 problem [0] DocumentAccessProblem,
1141 object identification [1] ObjectIdentification,
1142 error message [2] ErrorMessage }
1143 PrinterProblem ::= CHOICE {
1144 standard problem ENUMERATED {
1145 printer error (1),
1146 printer needs attention (2),
1147 printer needs key operator (3) },
1148 extended problem OBJECT IDENTIFIER }
1149 PrinterErrorSequence ::= SEQUENCE {
1150 problem [0] PrinterProblem,
1151 object identification [1] ObjectIdentification,
1152 error message [2] ErrorMessage }
1153 SecurityProblem ::= CHOICE {
1154 standard problem ENUMERATED {
1155 inappropriate authentication (1),
1156 invalid credentials (2),
1157 insufficient operation rights (3),
1158 invalid pac (4) },
1159 extended problem OBJECT IDENTIFIER }
1160 SecurityErrorSequence ::= SEQUENCE {
1161 problem [0] SecurityProblem,
1162 error message [1] ErrorMessage }
1163 SelectionProblem ::= CHOICE {
1164 standard problem ENUMERATED {
1165 invalid identification (1),
1166 unknown identification (2),
1167 object already exists (3) }
1168 extended problem OBJECT IDENTIFIER }
1169 SelectionErrorSequence ::= SEQUENCE OF
1170 SEQUENCE {
1171 problem [0] SelectionProblem,
1172 attribute [1] Attribute OPTIONAL,
1173 object identification [2] ObjectIdentification,
1174 ServiceProblem ::= CHOICE {
1175 standard problem ENUMERATED {
1176 server busy (1),
1177 server unavailable (2),
1178 operation too complex (3),
1179 resource limit exceeded (4),
1180 unclassified server error (5),
1181 too many items in list (6),
1182 compulsory resource not available (7),

```

```

1183 cancel document unsupported (8),
1184 modify document unsupported (9),
1185 print multiple documents unsupported (10),
1186 unsupported parameter value (11),
1187 invalid checkpoint (12),
1188 invalid continuation context (13),
1189 pause limit exceeded (14),
1190 unsupported operation (15) },
1191 extended problem OBJECT IDENTIFIER }
1192 ServiceErrorSequence ::= SEQUENCE OF
1193 SEQUENCE {
1194 problem [0] ServiceProblem,
1195 attribute [1] Attribute OPTIONAL,
1196 object identification [2] ObjectIdentification,
1197 error message [3] ErrorMessage }
1198 UpdateProblem ::= CHOICE {
1199 standard problem ENUMERATED {
1200 no modifications allowed (1),
1201 insufficient update rights (2),
1202 previous operation incomplete (4),
1203 cancellation not possible (5) },
1204 extended problem OBJECT IDENTIFIER }
1205 UpdateErrorSequence ::= SEQUENCE {
1206 problem [0] UpdateProblem,
1207 object identification [1] ObjectIdentification,
1208 error message [2] ErrorMessage }

```

~~5. Binding and Unbinding~~

~~There are two special operations that are defined for establishing a "session" between a client and a server. These are the BIND and the UNBIND operations.~~

~~5.1 Bind Operation~~

~~5.1.1 Bind Argument~~

~~The following abstract data types are part of the Bind Argument:~~

Printer Name	The name instance of the Print Service Provider (Printer object) to which the bind is being done.
Credentials	These can simple (name of the client perfoming the Bind) or the actual opaque Credential from some security/authorization service. All LDPA implemenations must support at least the simple option.
Other Security Info	Optional additional opaque security information if needed for a given security/authorization service.

~~5.1.1.1 XDR~~

```

1226 struct Creds {
1227 Text name;
1228 opaque password<>;
1229 };
1230
1231 struct Other1 {

```

```

1232 ----- string ----- serverNamePtr<>;
1233 ----- nuint16 ----- connection;
1234 -----};
1235
1236 ----- struct Othern {
1237 ----- nuint16 ----- othern;
1238 -----};
1239
1240 ----- enum CredentialsEnum {
1241 ----- CREDENTIALS_SIMPLE, ----- /* (0) */
1242 ----- CREDENTIALS_CERTIFIED, ----- /* (1) */
1243 ----- CREDENTIALS_OTHER_1, ----- /* (2) */
1244 ----- CREDENTIALS_OTHER_2, ----- /* (3) */
1245 ----- /* ... */
1246 ----- CREDENTIALS_OTHER_n ----- /* (n) */
1247 -----};
1248
1249 ----- union Credentials switch(CredentialsEnum designator) {
1250 ----- case CREDENTIALS_SIMPLE:
1251 ----- Creds ----- simple;
1252 ----- case CREDENTIALS_CERTIFIED:
1253 ----- opaque ----- certified<>;
1254 ----- case CREDENTIALS_OTHER1:
1255 ----- struct Other1 other1;
1256 -----};
1257
1258 ----- struct BindPrinterArgument {
1259 ----- QualifiedName ----- printerId;
1260 ----- Credentials ----- credentials;
1261 ----- nint32 ----- retrieveRestrictionsOption;
1262 ----- opaque ----- bindSecurityOption<>;
1263 -----};
1264
1265 5.1.1.1 ASN.1
1266
1267 ----- PrivilegeAttributeCertificate ::= EXTERNAL
1268
1269 ----- Creds ::= SEQUENCE {
1270 ----- name ----- [0] DistinguishedNameString,
1271 ----- password ----- [1] OCTET STRING ----- }
1272
1273 ----- Credentials ::= CHOICE {
1274 ----- simple ----- [0] Creds, ----- used for initial
1275 ----- ----- authentication -----
1276 ----- certified ----- [1] PrivilegeAttributeCertificate ----- }
1277 ----- ----- used when initial authentication has already taken place -----
1278 ----- ----- external to the DP Server -----
1279
1280 ----- Restrictions ::= SET {
1281 ----- maximum result length ----- [1] ResultLength OPTIONAL ----- }
1282 ----- ----- default is no restriction -----
1283
1284 ----- ResultLength ::= INTEGER (1..ub integer)
1285
1286 ----- BindSecurity ::= EXTERNAL
1287
1288 ----- DpBindArgument ::= SEQUENCE {
1289 ----- credentials ----- [0] Credentials,
1290 ----- retrieve restrictions ----- [1] Restrictions OPTIONAL,
1291 ----- ----- default is none -----

```

~~bind security [2] BindSecurity OPTIONAL }~~

~~5.1.2 Bind Result~~

~~The following abstract data types are part of the Bind Result:~~

Results	The authentication attributes
Errors	Optional Error information
Session Handle	Session Handle

~~5.1.2.1 XDR~~

```
struct BindResult {
    OctetString authAttributeSet<>;
    ErrorReturn *errorReturnOptionPtr;
    nint32 sessionHandle;
};
```

~~5.1.2.2 ASN.1~~

~~AuthenticationAttribute ::= EXTERNAL~~

~~DpBindResult ::= SET {~~
~~authentication attributes [0] SET OF AuthenticationAttribute }~~

~~DpBindError ::= CHOICE {~~
~~service error [0] ServiceProblem,~~
~~security error [1] SecurityProblem }~~

~~5.2 Unbind Operation~~

~~5.2.1 Unbind Argument~~

~~The following abstract data types are part of the Unbind Argument:~~

Session Handle	Session Handle
---------------------------	---------------------------

~~5.2.1.1 XDR~~

```
struct UnbindArgument {
    nint32 sessionHandle;
};
```

~~5.2.1.2 ASN.1~~

~~DpUnbind ::= ABSTRACT UNBIND~~
~~FROM { dp-user[S], dp-administration[S] }~~

~~5.2.2 Unbind Result~~

~~The following abstract data types are part of the Unbind Argument:~~

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

~~5.2.2.1 XDR~~


```

1342 struct UnbindResult {
1343     ErrorReturn *errorReturnOptionPtr;
1344 };
```

1347 5.2.2.2 ASN.1

1348 ~~No arguments or errors associated with Unbind.~~

1351 6. User Operations

1353 6.1 Print Operation

1354 When an end-user uses GUI to submit a job, the GUI client gets an
 1355 HTML form from the default printer. If the end-user changes the
 1356 selected printer, the GUI client gets the HTML form from that
 1357 printer. The HTML form comes with the values supported by the printer
 1358 and it is initialized by the values from the job template associated
 1359 with the named printer.

1360 [Further work needs to done to define the above concept.]

1364 6.1.1 Print Argument

1365 [We should be trying to create a protocol where the entire job is
 1366 incorporated into a single transmission. This eliminates the need
 1367 for Add Document and Close Job.]

1368 The following abstract data types are part of the Print Argument:

Session Handle	The handle for this session.
Create Job	One of the three modes for the Print Arguments (Create Job, Add Document, Close Job). If it is a Create Job, the Job Id is returned in the Print Results.
Printer	
Name	
Job Submission Complete	
Job and Document Attributes	
AllFirst Document Contents Description Add Document	Transfer method, content, type, and Document Attributes
Job Id	The job to which this document is added.
Job Submission Complete	
First Document Description Close Job	Transfer method, content, type, and Document Attributes

Job-Id	The job to close (no more documents can be added)
Common Arguments	Common to all three forms of Print Argument

6.1.1.1 XDR

```

struct DocumentDescription {
    ObjectIdentifier transferMethod;
    DocumentContent *documentContentOptionPtr;
    ObjectIdentifier documentType;
    AttributeSet documentAttributes;
};

struct CreateJob {
    QualifiedName printerName;
    bool jobSubmissionComplete;
    AttributeSet jobAttributes;
    DocumentDescription *firstDocumentOptionPtr;
    CommonArguments commonArgumentsOption;
};

struct AddDocument {
    PrtContainedObjectID existingJob;
    bool jobSubmissionComplete;
    DocumentDescription *newDocumentPtr;
    CommonArguments commonArgumentsOption;
};

struct CloseJob {
    PrtContainedObjectID existingJob;
    CommonArguments commonArgumentsOption;
};

enum PrintArgEnum {
    PRINT_ARG_CREATE_JOB, /* (0) */
    PRINT_ARG_ADD_DOCUMENT, /* (1) */
    PRINT_ARG_CLOSE_JOB /* (2) */
};

union PrintOperation switch(PrintArgEnum designator) {
    case PRINT_ARG_CREATE_JOB:
        CreateJob createJob;
    case PRINT_ARG_ADD_DOCUMENT:
        AddDocument addDocument;
    case PRINT_ARG_CLOSE_JOB:
        CloseJob closeJob;
};

struct PrintArgument {
    nint32 sessionHandle;
    PrintOperation printOperation;
};

```

6.1.1.2 ASN.1

```

DocumentDescription ::= SEQUENCE {
    transfer method {0} OBJECT IDENTIFIER
    DEFAULT id val transfer method with request,

```

```

document content [1] DocumentContent OPTIONAL,
document type [2] OBJECT IDENTIFIER
DEFAULT id val document type printable,
document attributes [3] SET OF Attribute OPTIONAL
-- Contains any document attributes valid for the document,
-- except any document status attributes.
document type = printable, font, or resource.
If document type is font, a font identifier attribute is
required in the document attributes element.
-- If document type is resource, a resource name attribute
is required in the document attributes element. }

PrintArgument ::= CHOICE {
create job [0] SEQUENCE {
printer name [0] SimpleName,
job submission complete [1] BOOLEAN DEFAULT TRUE,
job attributes [2] SET OF Attribute OPTIONAL,
may include any job attribute, except
id att job identifier,
-- id att printer name requested, and
any job status attribute
first document [3] DocumentDescription OPTIONAL,
common arguments [4] CommonArguments OPTIONAL },
add document [1] SEQUENCE {
existing job [0] JobIdentifier,
job submission complete [1] BOOLEAN DEFAULT TRUE,
new document [3] DocumentDescription,
common arguments [4] CommonArguments OPTIONAL },
close job [2] SEQUENCE {
existing job [0] JobIdentifier,
common arguments [4] CommonArguments OPTIONAL } }

```

6.1.2 Print Result

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

Job Id	Used for all other operations on this Job.
Server State	Optional state information about the Print Service Provider
Message	Optional message
Document Status	Optional document status information
Job Status	Job state information
Errors	Optional Error Information

6.1.2.1 XDR

```

struct PrintResult {
    PrtContainedObjectId jobIdentification;
    ObjectIdentifier serverStateOption;
    NameOrOid *serverMessageOptionPtr;
    AttributeSet documentStatusOption;
    AttributeSet jobStatus;
    ErrorReturn *errorReturnOptionPtr;
};

```

~~6.1.2.2 ASN.1~~

```

PrintResult ::= SEQUENCE {
  job identification [0] JobIdentifier,
    value of id att job identifier
    value of id att server state
  server message [2] Message OPTIONAL,
    value of server's id att message
  document status [3] SET OF Attribute OPTIONAL,
    may include id att document state,
    id att document sequence number,
    id att file reference, and
    id att copies completed.
    See document status attributes subelause.
  job status [4] SET OF Attribute
    may include any job status attributes
    See job status attributes subelause.
}

```

6.2 Cancel Job Operation

6.2.1 Cancel Job Argument

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

Session Handle	The handle for this session.
Job Id	The identifier of the job to be canceled.
Document Number	Optional document number of the document to cancel within a given job. <u>[probably not supported]</u>
Message	Optional message to the operator.
Retention Period	Optional period for retaining the cancelled job.
Common Arguments	

~~6.2.1.1 XDR~~

```

struct CancelJobArgument {
  nint32 sessionHandle;
  PrtContainedObjectId jobIdentifier;
  nuint32 documentNumberOption;
  NameOrOid *cancelMessageOptionPtr;
  IntegerOption retentionPeriodOption;
  CommonArguments commonArgumentsOption;
};

```

~~6.2.1.2 ASN.1~~

```

CancelJobArgument ::= SEQUENCE {
  job identification [0] JobIdentifier,
  document number [1] PositiveInteger OPTIONAL,
    required for addressing individual
    documents in a multiple document print job
  cancel message [2] Message OPTIONAL,

```

~~sets value of id att job message from administrator~~
~~retention period [3] DeltaTime OPTIONAL,~~
~~common arguments [4] CommonArguments OPTIONAL }~~

~~6.2.2 Cancel Job Result~~

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

Job Status	Optional Job status information
Errors	Optional Error Information

~~6.2.2.1 XDR~~

```
struct CancelJobResult {
    AttributeSet jobStatusOption;
    ErrorReturn *errorReturnOptionPtr;
};
```

~~6.2.2.2 ASN.1~~

```
CancelJobResult ::= SEQUENCE {
    status [0] SET OF Attribute OPTIONAL
    -- any job status or document status attributes }
```

~~6.3 Get List Object Attributes Operation~~

~~6.3.1 Get List Object Attributes Argument~~

The following abstract data types are part of the Get Attributes List Object Attributes Argument:

Session Handle	Handle for this session.
Operation	CONTINUE or SPECIFICATION
SPECIFICATION	
Class	The class type for which this operation is being performed (Printer, Job, Document, etc.)
Scope	Levels of object containment to report
Selector	A set of job or printer name URL (the class is implicit in the object named) object identifiers (possibly wild carded), optional filter information, time limits, and count limits.
Requested Attributes	A set of attributes in which the requestor is interested
Operation	ATTRIBUTES or ORDERED_JOBS if requesting Jobs contained by a given Printer.
CONTINUATION	[I would like to get rid of this if possible]
Context	Context for continuing
Abort	Should the operation be aborted? (boolean)

Common Arguments

6.3.1 Get Jobs Argument

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

<u>Selector</u>	<u>A printer name</u>
<u>Filtering</u>	<u>A lightweight filtering mechanism, such as all jobs versus a particular user's jobs.</u>
<u>Requested Attributes</u>	<u>A set of job attributes in which the requestor is interested</u>
<u>Common Arguments</u>	

6.3.1.1 XDR

```

struct Selector {
    ObjectIdentificationSeq objectIdentificationSeqOption;
    Filter                  *objectFilterOptionPtr;
    nuint32                 timeLimitOption;
    nuint32                 countLimitOption;
};

enum ListOperatorEnum {
    LIST_OP_ATTRIBUTES,                /* (0) */
    LIST_OP_ORDERED_JOBS = 2          /* (1) */
};

struct ListSpecification {
    ObjectIdentifier         objectClass;
    nuint32                  scope;                /* default 0; */
    Selector                 *selectorOptionPtr;
    ObjectIdentifierSet      *requestedAttrsOptionPtr;
    ListOperatorEnum         listOperator;
    /* default DpaReturnAttributes */
    CommonArguments         commonArgumentsOption;
};

struct ListContinuation {
    OctetString              context;
    bool                     abort;
    CommonArguments         commonArgumentsOption;
};

enum ListAttrsArgEnum {
    LIST_ATTRIBUTES_ARG_CONTINUE,      /* (0) */
    LIST_ATTRIBUTES_ARG_SPEC          /* (1) */
};

union ListAttrsOperation switch(ListAttrsArgEnum designator) {
    case LIST_ATTRIBUTES_ARG_CONTINUE:
        ListContinuation continuation;
    case LIST_ATTRIBUTES_ARG_SPEC:
        ListSpecification specification;
};

```

```

1602 struct ListObjectAttrsArgument {
1603     nint32 sessionHandle;
1604     ListAttrsOperation listAttrsOperation;
1605 };
1606
1607 6.3.1.2 ASN.1
1608
1609 SubstringMatchCriteria ::= ENUMERATED {
1610     exact (0),
1611     case insensitive (1),
1612     same letter (2), ignoring accents, case, etc.
1613     approximate (3) implementation defined }
1614
1615 FilterItem ::= CHOICE {
1616     equality [0] AttributeValueAssertion,
1617     substrings [1] SEQUENCE {
1618         attribute id [0] AttributeId,
1619         match criteria [1] SubstringMatchCriteria,
1620         initial string [2] ANY OPTIONAL,
1621         -- DEFINED BY attribute id
1622         any string [3] SEQUENCE OF ANY OPTIONAL,
1623         -- DEFINED BY attribute id
1624         final string [4] ANY OPTIONAL },
1625         -- DEFINED BY attribute id
1626     greater or equal [2] AttributeValueAssertion,
1627         -- asserted value is greater than or equal to
1628         -- the attribute value
1629     less or equal [3] AttributeValueAssertion,
1630         -- asserted value is less than or equal to
1631         -- the attribute value
1632     present [4] AttributeId,
1633         -- asserted attribute is present (with any value)
1634     subset of [5] AttributeValueAssertion,
1635         -- asserted value is a subset of attribute value
1636     superset of [6] AttributeValueAssertion,
1637         -- asserted value is a superset of attribute value
1638     non null set intersection [7] Attribute
1639         -- at least one of the members of the asserted
1640         -- value is present in the attribute value }
1641 Filter ::= CHOICE {
1642     item [0] FilterItem,
1643     and [1] SET OF Filter,
1644     or [2] SET OF Filter,
1645     not [3] Filter }
1646
1647 SubstringMatchCriteria ::= ENUMERATED {
1648     exact (0),
1649     case insensitive (1),
1650     same letter (2), ignoring accents, case, etc.
1651     approximate (3) implementation defined }
1652
1653 ContinuationContext ::= OCTET STRING
1654     implementation specific information
1655 Selector ::= SET {
1656     object identification [0] SEQUENCE OF ObjectIdentification
1657         OPTIONAL,
1658     -- should not be omitted if class is id oc document
1659     object filter [1] Filter OPTIONAL,
1660     time limit [2] DeltaTime OPTIONAL,
1661     count limit [3] PositiveInteger OPTIONAL }

```

```

1662 ObjectIdentification ::= CHOICE {
1663   job identifier [0] JobIdentifier,
1664   document identifier [1] DocumentIdentifier,
1665   object identifier [2] OBJECT IDENTIFIER,
1666   object name [3] DistinguishedNameString,
1667   font reference [4] FontReference,
1668   name or oid [6] NameOrOid,
1669   simple name [7] SimpleName }
1670
1671 DocumentIdentifier ::= SEQUENCE {
1672   job identifier [0] JobIdentifier,
1673   document number [1] PositiveInteger OPTIONAL
1674   document sequence number
1675
1676 ListOperator ::= ENUMERATED {
1677   get attributes (0),
1678   get ordered jobs (2) }
1679
1680 ListObjectAttributesArgument ::= SEQUENCE {
1681   CHOICE {
1682     continuation [0] SEQUENCE {
1683       context [0] ContinuationContext,
1684       abort [1] BOOLEAN DEFAULT FALSE,
1685       common arguments [2] CommonArguments OPTIONAL },
1686     specification [1] SEQUENCE {
1687       class [0] OBJECT IDENTIFIER, id-oc-xxx
1688       scope [1] Cardinal DEFAULT 0,
1689       scope is contained objects in levels 0 through n
1690       where 0 means the base object specified
1691       by the object identification
1692       selector [2] Selector OPTIONAL,
1693       should not be omitted if class is id-oc-document
1694       requested attributes [3] SET OF AttributeId OPTIONAL,
1695       list operator [4] ListOperator
1696       DEFAULT get attributes,
1697       common arguments [5] CommonArguments OPTIONAL } } }

```

6.3.2 ~~GetList Object~~ Attributes and Get Jobs Result

The following abstract data types are part of the List Object Attributes Result:

Time	The operation can take an indeterminate amount of time to process. The results to a single Argument can be returned in multiple phases. This Result of for one of those phases. This processing time element is the time required for this phase of the operation.
Continuation Context	Optional opaque context information for performing another argument request on the next phase of the same operation.
Limit Encountered	Information on the type of limit that was encountered which forces the end of the operation even if there is a potential for more results. Values include TIME, COUNT, ERRORS.
Result Attributes	Attribute set containing the returned results.
Errors	Optional Error Information

~~6.3.2.1 XDR~~

```

enum LimitEncounteredEnum {
    LIMIT_ENCOUNTERED_TIME, /* (0) */
    LIMIT_ENCOUNTERED_COUNT, /* (1) */
    LIMIT_ENCOUNTERED_ERROR /* (2) */
};

struct LimitEncounteredOption {
    nint32 length; /* 0 or 1 */
    LimitEncounteredEnum value;
};

struct ObjectResult {
    ObjectIdentification objectIdentification;
    AttributeSet attributes;
    ObjectIdentifier objectClass;
};

typedef ObjectResult ObjectResultSet<>;

struct ListObjectAttrsResult {
    nuint32 answerTime;
    OctetString continuationOption;
    LimitEncounteredOption limitEncounteredOption;
    ObjectResultSet resultSet;
    ErrorReturn *errorReturnOptionPtr;
};

```

~~6.3.2.2 ASN.1~~

```

ContinuationContext ::= OCTET STRING
-- implementation specific information

LimitEncountered ::= ENUMERATED {
    time limit (0),
    count limit (1),
    error limit (2) }

ObjectResult ::= SEQUENCE {
    object identification [0] ObjectIdentification,
    attributes [1] SET OF Attribute
    object class [2] OBJECT IDENTIFIER },
-- id-oc-xxx

ListObjectAttributesResult ::= SEQUENCE {
    answer time [1] GeneralizedTime,
    continuation [2] ContinuationContext OPTIONAL,
    limit encountered [3] LimitEncountered OPTIONAL,
    result set [4] SEQUENCE OF ObjectResult }

```

6.4 Modify Job Operation

6.4.1 Modify Job Argument

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

Session Handle	Handle for this session.
Job Id	Which job to modify.
Document Number	Optionally the document to modify if not modifying a job attribute. [There are no document attributes to modify.]
Job Attributes	Attribute set for Job attributes. Values can be modified in any of the following ways: ADD_ATTRIBUTE, REPLACE, ADD_VALUES, REMOVE_VALUES, SET_TO_DEFAULT, or REMOVE_ATTRIBUTE Only replacement is possible; the GUI fetches the value and then sets a new one.
Document Attributes	Attribute set for Document attributes.
Message	Optional Message.
Common Arguments	

6.4.1.1 XDR

```

enum ModifyOperatorEnum {
    MODIFY_OP_NULL, /* (0) */
    MODIFY_OP_REPLACE, /* (1) */
    MODIFY_OP_ADD_VALUES, /* (2) */
    MODIFY_OP_REMOVE_VALUES, /* (3) */
    MODIFY_OP_SET_TO_DEFAULT, /* (4) */
    MODIFY_OP_REMOVE_ATTRIBUTE /* (5) */
};

struct ModifyJobArgument {
    nint32 sessionHandle;
    PrtContainedObjectId jobIdentification;
    nuint32 documentNumberOption;
    AttributeSet jobAttrModificationSet;
    AttributeSet docAttrModificationSet;
    NameOrOid *modifyMessageOptionPtr;
    CommonArguments commonArgumentsOption;
};

```

6.4.1.2 ASN.1

```

JobAttrModification ::= SEQUENCE {
    attribute_id [0] AttributeId,
    Any job attributes, except:
    id att job identifier,
    id att job owner, id att job originator,
    id att printer name requested,
    id att initial value job,
    any access and accounting attributes,
    any job security attributes, and
    any job status attributes.
    Any document attributes, except:
    id att transfer method, id att document content,
    id att initial value document, and
    any document status attributes
    attribute values [1] SET OF ANY

```

```

----- DEFINED BY attribute id ----- OPTIONAL,
----- omitted for set to default
modify operator [2] ModifyOperator DEFAULT replace }
ModifyOperator ::= ENUMERATED {
    replace (0),
    add values (1),
    remove values (2),
    set to default (3) }
ModifyJobArgument ::= SEQUENCE {
    job identification [0] JobIdentifier,
    document number [1] PositiveInteger OPTIONAL,
    ----- required for addressing individual
    ----- documents in a multiple document print job
    job attr modification [2] SEQUENCE OF JobAttrModification,
    modify message [3] Message OPTIONAL,
    sets value of id att job message from administrator
    common arguments [4] CommonArguments OPTIONAL }

```

6.4.2 Modify Job Result

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

Modify Status	Modify result attributes.
Errors	Optional Error Information

~~6.4.2.1 XDR~~

```

struct ModifyJobResult {
    AttributeSet statusOption;
    ErrorReturn *errorReturnOptionPtr;
};

```

~~6.4.2.2 ASN.1~~

```

ModifyJobResult ::= SEQUENCE {
    status [0] SET OF Attribute OPTIONAL
    ----- any job status or document status attributes
}

```

6.5 Resubmit Job Operation

6.5.1 Resubmit Job Argument

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

Session Handle	Handle for this session.
Destination Printer Name	Optional name of the destination printer.
Destination Printer Address	The address of the destination printer (can be used instead of the name).
Operation	MOVE or COPY

Job Set	A set of jobs to move or copy. Each entry in the set has: Job Id, Document Number, Job attributes, and Document attributes.
Message	Optional Message
Common Arguments	

6.5.1.1 XDR

```

enum ResubmitOpEnum {
    RESUBMIT_OP_COPY, /* (0) */
    RESUBMIT_OP_MOVE /* (1) */
};

/*
// If documentNumber is 0, docAttrSet is applied to all documents
*/

struct ResubmitJob {
    PrtContainedObjectId jobId;
    nuint32 documentNumber;
    AttributeSet jobAttrSet;
    AttributeSet docAttrSet;
};

typedef ResubmitJob ResubmitJobSet<>;

struct ResubmitJobsArgument {
    nint32 sessionHandle;
    QualifiedName destPrinterNameOption;
    NetAddress *destPrinterNetAddressPtr;
    ResubmitOpEnum operation;
    ResubmitJobSet resubmitJobSet;
    NameOrOid *resubmitMessageOptionPtr;
    CommonArguments commonArgumentsOption;
};

```

6.5.1.2 ASN.1

```

ResubmitJobArgument ::= SEQUENCE {
    object-class [0] OBJECT IDENTIFIER,
    id oc job, id oc printer,
    id oc server
    object-identification [1] ObjectIdentification,
    printer [2] DistinguishedNameString,
    message [3] Message OPTIONAL,
    common arguments [4] CommonArguments OPTIONAL
}

```

6.5.2 Resubmit Job Result

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

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

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

~~6.5.2.1 XDR~~

```

struct ResubmitJobResult {
    PrtContainedObjectId oldJobIdentifier;
    PrtContainedObjectId newJobIdentifier;
    AttributeSet jobStatusOption;
};

typedef ResubmitJobResult ResubmitJobResultSet<>;

struct ResubmitJobsResult {
    ResubmitJobResultSet resubmitJobResultSet;
    ErrorReturn *errorReturnOptionPtr;
};

```

~~6.5.2.2 ASN.1~~

```

ObjectStatus ::= SEQUENCE {
    object-status [0] SET OF Attribute OPTIONAL }
    job identifier and new job identifier shall be
    returned at least. For any jobs that could not
    be resubmitted, the new job identifier attribute
    shall be omitted as the only error indication.
}

ResubmitJobResult ::= SEQUENCE {
    result-set [0] SEQUENCE OF ObjectStatus }
    one result set for each job resubmitted
    (or for each job attempted to be resubmitted)
}

```

7. Object Attributes

This section describes the attributes and their associated values that are part of the LDPA protocol. The list below shows the objects and their attributes that are included within the scope of this protocol:

Job Attributes

```

Job Informational Attributes (set by client)
    job-identifier
    job-owner originator (an authenticated value)
    job-name
    job-originating-host
Job Informational Attributes (set by Printer)
    job-identifier
    job-identifier-on-output-device (used by operator)
Printer Selection Attributes (set by client)
    printer-name-requested
    output-device-requested
Job Status Attributes (set by Printer)
    current-job-state
    printers-assigned [ let's keep it simple ]
    submission-time
    print-checkpoint
    job-message-from-administrator
    completion-time

```

1957 job-state-reasons
 1958 impressions-completed
 1959 media-sheets-completed
 1960 number of documents
 1961 ~~job-submission-complete~~
 1962 Job sheet Job Results Handling Attributes (set by client)
 1963 job-sheets
 1964 ~~document-sheets~~
 1965 Job Event Handling Attributes (set by client)
 1966 notification-profile (two classes of events, delivery
 1967 methods other than email are a problem with internet)
 1968 Job Scheduling Instructions Attributes (set by client)
 1969 job-hold
 1970 job-priority
 1971 ~~job-print-after~~
 1972 Job-print-off-peak
 1973 job-retention-period
 1974 ~~Document Attributes~~
 1975 Document Description Attribute
 1976 document-format
 1977 ~~document-content~~
 1978 ~~transfer-method~~
 1979 ~~Document Production Instruction Attributes (set by client)~~
 1980 document-format
 1981 ~~default-font~~
 1982 ~~default-medium-select~~
 1983 number-up
 1984 finishing
 1985 sides
 1986 ~~copies-copy-count~~
 1987 ~~reset-printer~~
 1988 printer-resolution-select
 1989 print-quality
 1990 page-select
 1991 Attributes for Conversion of Text Files (set by client)
 1992 width
 1993 length
 1994 left-margin
 1995 right-margin
 1996 top-margin
 1997 bottom-margin
 1998 repeated-tab-stops
 1999 header-text
 2000 footer-text
 2001 number-pages
 2002 default-font
 2003 default-character-set
 2004 content-orientation
 2005 Job Resource Document Characteristics Attributes (set by
 2006 process which produces PDL file; for use in scheduling)
 2007 document-format-used
 2008 fonts-used
 2009 character-sets-used
 2010 media-used
 2011 sides-used
 2012 print-quality-used
 2013 finishing-used
 2014 printer-resolution-used
 2015 total-job-octets
 2016 job-impression-count

2017 job-media-sheet-count
 2018 Document Status Attributes
 2019 document sequence number
 2020 Document Contents (one per document)
 2021 number-of-documents
 2022 document-content (actual contents or a path reference)
 2023 Operation Attributes
 2024 operation-locale
 2025 default-delivery-addresses
 2026 Printer Attributes (Print Servers and Output Devices)
 2027 printer-name
 2028 printer-location
 2029 printer-model
 2030 printer-types
 2031 printer-state
 2032 printer-state-message
 2033 message
 2034 notification-profile
 2035 access-control-list
 2036 printer-initial-value-job
 2037 printer-initial-value-document
 2038 fonts-supported
 2039 font-substitutions
 2040 fonts-ready
 2041 media-supported
 2042 media-ready
 2043 printer-associated-printers
 2044 document-formats-supported
 2045 numbers-up-supported
 2046 finishings-supported
 2047 sides-supported
 2048 print-qualities-supported
 2049 maximum-printer-speed
 2050 printer-resolutions-supported
 2051 delivery-methods-supported
 2052 character-sets-supported
 2053 job-sheets-supported
 2054 document-sheets-supported
 2055 maximum-copies-supported
 2056 maximum-job-octets
 2057 maximum-job-retention-period
 2058 maximum-job-priority
 2059 maximum-impressions
 2060 maximum-media-sheets
 2061 off-peak-times
 2062 notification-delivery-methods-supported
 2063 server-name
 2064 server-state
 2065 downstream-printers
 2066 physical-printers-supported
 2067 logical-printers-supported
 2068 events-supported
 2069 transfer-methods-supported
 2070 locales-supported
 2071 locale
 2072 multiple-documents-supported
 2073 cancel-individual-document-supported
 2074 modify-individual-document-supported
 2075 sheet-count
 2076 printer-timeout-period

~~Initial Value Job Attributes~~
~~Initial Value Document Attributes~~
Job Template (attributes from the following sections of Job)
Job sheet Attributes
Job Event Handling Attributes
Job Scheduling Instructions Attributes
Document Production Instruction Attributes
Attributes for Conversion of Text Files

In the following sections, most of the text has been taken word for word from ISO/IEC 10175 DPA (Final, June 1996).

7.1 Job Attributes

A job object contains a set of job attributes and one or more document ~~objects~~. The server shall create a printable job object in response to a client that invokes one or more Print abstract-operations. A client shall use a job template associated with the selected printer in order to initialize the job. In addition, initial value job objects are created in a server by means outside the scope of this part of ISO/IEC 10175 in order to represent complete sets of default values for job attributes (see the initial value job object class).

~~In addition to the attributes specifically defined for the job and initial value job objects, certain of the generic attributes may also be associated with these objects. For example, when requesting a list of attribute values for an object of these classes, the client may identify one or more of the generic attributes in the following table, for which the server shall return values if the attributes are implemented.~~

~~There are no notification profiles included in this LDPA specification.~~

There is a table for each attribute that shows its: name, syntax, multi or single valuedness (S or M), and any relevant notes.

7.1.1 Job Informational Attributes

These attributes provide information to identify a print-job.

The client may specify job-information attributes in:

- a) Print: all, except id-att-job-identifier
- b) ModifyJob: all, except id-att-job-identifier, id-att-job-owner
- c) ListObjectAttributes: all

7.1.1.1 job-identifier

job-identifier	jobIdentifierSyntax	S	
----------------	---------------------	---	--

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

The value of the job-identifier attribute shall be returned by the server as part of the PrintResult in the first Print

abstract-operation for the job. The client shall pass its value as part of the argument in subsequent abstract-operations for the same job.

7.1.1.1 ~~job-identifier-on-output-device~~~~printer~~

This attribute holds the job-identifier assigned by an Output Device. It provides a way for a Print Server to relates its copy of a job with a job on the Output Device.

7.1.1.2 ~~job-originator~~~~owner~~

job-<u>originator</u> owner	Text distinguishedNameStrin gSyntax	S	
--	---	---	--

This attribute specifies the name of the person submitting the print job. The Print Server or Output Device shall set this attribute to the most authentic name that it can obtain from the client.

[We may choose to call this job-owner, but it should still be the most authenticated name of the user]

~~Attribute value types that specify the name of an object, file, or person as a string that can be either (1) a simple name by itself or (2) a simple name qualified with a path name employ this generic data type and syntax. If the path name is included, an optional name syntax element may be used to specify the syntax of the path name, i.e., to identify the name syntax of the service being used. If the name syntax element is omitted, the server shall assume the name syntax is identified by some other means.~~

~~The following standard values are defined for use in the name syntax element to identify the syntax of names:~~

Descriptive Name	Object Identifier	Descriptor Text
		automatic id val dn sy
		X-500 id val dn sy
		XFN id val dn sy
DCE	id val dn syntax-dec	Distributed Computing Environment includes X.500 and CDS
		CDS id val dn sy
		NIS id val dn synta
		*-nis
		Network Information Service
		DNS id val dn synta
		*-dns
		Domain Name Service

2191	DEC NS	
2192	id val dn synta	
2193	* dec ns	
2194	Digital Name	
2195	Service	
2196	Internet mail id val dn syntax internet mail	Internet Mail
2197	XNS	
2198	id val dn synta	
2199	* xns	
2200	Xerox Network	
2201	System	
2202	Bindery	
2203	id val dn synta	
2204	* bindery	
2205	NDS	
2206	id val dn synta	
2207	* nds	
2208	Novell	
2209	Directory Service	
2210	URL	
2211	id val dn synta	
2212	* url	
2213	HTTP Universal	
2214	Resource Locator	
2215	POSIX	
2216	id val dn synta	
2217	* posix	
2218	POSIX file name	
2219	(ISO/IEC 9945-1)	
2220	UNIX	
2221	id val dn synta	
2222	* unix	
2223	UNIX(TM) file	
2224	name	
2225	OS/2	
2226	id val dn synta	
2227	* os2	
2228	OS/2 file name	
2229	PC DOS	
2230	id val dn synta	
2231	* pc dos	PC
2232	DOS file name	
2233	NT	
2234	id val dn synta	
2235	* nt	NT
2236	file name	
2237	MVS	
2238	id val dn synta	
2239	* mvs	MVS
2240	file name	
2241	VM	
2242	id val dn synta	
2243	* vm	VM
2244	file name	
2245	OS/400	
2246	id val dn synta	
2247	* os400	
2248	OS/400 file	
2249	name	

2250 VMS _____
 2251 _____ id val dn synta
 2252 * vms _____ VMS
 2253 file name
 2254 UNC _____
 2255 _____ id val dn synta
 2256 * unc _____
 2257 _____ Microsoft
 2258 Universal Name
 2259 Convention
 2260 NetWare _____
 2261 _____ id val dn synta
 2262 * netware _____
 2263 _____ NetWare file
 2264 path name

2265
 2266 ~~As with any NameOrOid, implementors may use their own object~~
 2267 ~~identifiers or simple names (if they have not assigned an OID) for~~
 2268 ~~implementation defined name syntaxes.~~

2269
 2270 ~~This attribute supplies the name of the human owner of the~~
 2271 ~~print job.~~

2272
 2273 ~~The value of job owner will often be the same as job originator.~~
 2274 ~~The job owner will be different from job originator when the job~~
 2275 ~~has been submitted by the originator on behalf of the owner.~~

2276
 2277 ~~If this attribute is not specified, the value of user name or~~
 2278 ~~job originator should be used for any circumstances which require~~
 2279 ~~a value for job owner.~~

2280
 2281
 2282 7.1.1.3 TBD

2283
 2284
 2285 7.1.1.4 job-name

2286 job-name	simpleNameSyntax	S	
---------------	------------------	---	--

2287
 2288 This attribute supplies a human readable string for the print-job.
 2289 This string is used for naming the print-job in human-readable
 2290 "free-form" fashion.

2291
 2292 This attribute is intended for enabling a user or the user's
 2293 application to convey a job name that may be printed on a start
 2294 sheet, returned in a Get-Attributes ~~ListObjectAttributes~~ result,
 2295 or used in notification or logging messages.

2296
 2297 ~~If the client does not specify this attribute is not specified, A~~
 2298 ~~Print Server or Output Device shall set it to the name of the file~~
 2299 ~~of the first document in the job. no job name is assumed, but~~
 2300 ~~implementation specific defaults are allowed, such as the value of~~
 2301 ~~the document name attribute of the first document in the job.~~

2302
 2303 7.1.1.4 job-originating-host

2304
 2305 The client sets this attribute to contain the host name from which
 2306 the job is submitted.

2307

7.1.1.4 number-of-documents

The client sets this attribute to be the number of document being submitted.

7.1.2 Printer Selection Attributes

These attributes provide information to help select a particular printer. If more than one printer-selection attribute is specified, the server shall select a printer that meets all of the criteria.

The client may specify printer-selection attributes in:

- a) Print: all, except the value of printer-name-requested (which shall be passed as an explicit parameter of the first PrintArgument, rather than as an attribute)
- b) ModifyJob: all, except printer-name-requested
- c) ListObjectAttributes: all

7.1.2.1 printer-name-requested

printer-name-requested	simpleNameSyntax	S	
------------------------	------------------	---	--

This attribute identifies the printer to be used for printing the job. ~~The client shall specify the value of this attribute with the first invocation of the Print abstract operation for the print job as the explicit printer name component of the PrintArgument, rather than as an attribute.~~

NOTES

- ~~1 To cause a server to select a printer according to other attributes, the system administrator should define a logical printer that supports the desired set of physical printers.~~
- ~~2 Initial value job objects should have the value of their printer name requested attribute specified as an empty value in order to indicate that no printer name is defaulted.~~

7.1.2.1 output-device-requested

<u>output-device-requested</u>	simpleNameSyntax	<u>S</u>	
--------------------------------	------------------	----------	--

This attribute identifies the Output Device to be used for printing the job. This attribute has significance when the printer-name-requested references a Print Server and that Print Server has two or more downstream Output Devices.

7.1.3 Job Status Attributes

These attributes specify the job status before, during and after the processing of the print-job by the server. The server shall create the job object with these attributes (if implemented) and shall assign appropriate values to each such job-status attribute.

The client may specify job-status attributes in:

- a) Print: none

- b) ModifyJob: none
 c) GetAttributes and GetJobsListObjectAttributes: all

7.1.3.1 current-job-state

current-job-state	objectIdentifierSyntax	S	
-------------------	------------------------	---	--

This attribute identifies the current state of the job (pending, printing, held, etc.).

The following job state standard values are defined:

id-val-job-state-unknown, id-val-job-state-pre-processing,
 id-val-job-state-held, id-val-job-state-pending,
 id-val-job-state-processing, ~~id-val-job-state-paused,~~
~~id-val-job-state-interrupted,~~ id-val-job-state-terminating,
 id-val-job-state-retained, id-val-job-state-completed

The IPPLDPA protocol supports all values for job states, but printers are not required to generate all job states, only those which are appropriate for the particular implementation.

~~If a printer implementation or policy is to start processing documents before the last print request (with a TRUE value for the job submission complete parameter), the printer may change the job's current job state from pre-processing directly to the processing state when the printer begins processing any of the job's documents.~~

7.1.3.2 printers-assigned

printers-assigned	simpleNameSeqSyntax	S	
-------------------	---------------------	---	--

This attribute identifies the Output Device ~~physical printer or printers~~ to which this job has been assigned, if any.

When the job is first submitted and the printer has not yet assigned any printers to the job, the valueSEQUENCE shall be empty.

If the Print Server ~~printer intends to use a single printer for the job, and the printer~~ has assigned a printer to the job, the value SEQUENCE shall contain just the assigned Output Device that printer.

~~If a printer has split the job into multiple pieces and assigned each piece to a different printer, the SEQUENCE shall contain n elements, one for each assigned printer. A job with multiple job result sets is an example of a job that would be easy to split into multiple pieces.~~

~~An empty value SEQUENCE with no elements~~ shall be returned if this attribute is supported, but this job has not yet been assigned to any Output Devices ~~physical printers~~.

~~The number of elements in the SEQUENCE for this attribute shall be the same as the number of elements in the SEQUENCE for the associated job attribute printer state of printers assigned.~~

~~In addition, the ith element of the value of printer state of printers assigned shall be the state of the printer named by the ith element of printers assigned.~~

The printers-assigned value shall not be the same as the printer requested by the user. ~~if the job's printer name requested attribute specified a logical printer that supports one or more different physical printers. The printers assigned value might differ also if the job has been re-assigned by an operator to ensure successful completion of the job, allowing the user to find out where a job has been re-assigned (when necessary).~~

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

7.1.3.3 submission-time

submission-time	generalized TimeSyntax	S	
-----------------	-----------------------------------	---	--

This attribute indicates the time at which the ~~latest print request or~~ this job was accepted by the printer. If the printer does not support the notion of time, the attribute is not stored as part of the job object.

~~7.1.3.4 print checkpoint~~

print-checkpoint	print CheckpointSyntax	S	
-----------------------------	-----------------------------------	--------------	--

~~This attribute indicates the job copies, document copies, pages, and octets completed for the document or documents on the specified printer(s) and the local context information at which the last checkpoint was taken.~~

~~This attribute allows a print service to provide information about a checkpoint of a job that is printing. This would indicate where a print server could resume printing of this job at a page and copy number of a document close to the point at which the job was paused due to malfunction or operator request.~~

~~The context info element shall contain information that would be needed by the server and printer to enable them to resume printing at the last checkpoint. The format of this element identified by the checkpoint format element. The content of this element is implementation specific; the intent is that the client would return this element, without alteration, in a ResumeJobArgument in order to resume the job.~~

~~NOTE A server should encode the value of context info in such a way as to protect against clients submitting ResumeJob requests with altered context info.~~

~~The checkpoint format element shall identify the encoding format used for the context info element. Standard values are defined in the printer attribute checkpoint format supported.~~

~~Some systems support concurrent printing of a job on multiple printers. In such cases, the server shall return a PrintCheckpoint sequence for each printer currently assigned to the job.~~

~~The ability to generate the previous internal state of the job and the printer is dependent on the page independence supported by the document format. If a document format is not page independent, it may be possible to emulate the resumption of the job at the checkpoint by processing through the entire document to the checkpoint page without printing any additional pages, then continue printing pages from that point. Some document formats may not support any form of checkpointing.~~

~~If a PauseJob operation causes a job to pause in the middle of a document encoded in a document format that does not support checkpointing, the server shall set the checkpoint to a value that will force the system to resume back at the beginning of the current copy. Obviously the ability to checkpoint a job is very implementation dependent.~~

7.1.3.5 job-message-from-administrator

job-message-from-administrator	simpleNameSyntax	S	
--------------------------------	------------------	---	--

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

7.1.3.6 completion-time

completion-time	generalized TimeSyntax	S	
-----------------	-----------------------------------	---	--

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

7.1.3.7 job-state-reasons

job-state-reasons	enumobjectIdentifierSyntax	M	
-------------------	---------------------------------------	---	--

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

The following standard values are defined:
 id-val-reasons-documents-needed, id-val-reasons-job-hold-set,
 id-val-reasons-job-print-after-specified,
 id-val-reasons-required-resources-not-ready,
 id-val-reasons-successful completion,

id-val-reasons-completed-with-warnings,
 id-val-reasons-completed-with-errors,
 id-val-reasons-cancelled-by-user,
 id-val-reasons-cancelled-by-operator,
 id-val-reasons-aborted-by-system, id-val-reasons-logfile-pending,
 id-val-reasons-logfile-transferring

7.1.3.8 number-of-documents

number-of-documents	cardinalSyntax	S	
---------------------	----------------	---	--

This attribute indicates the number of documents in the job. ~~The number indicates how many Print abstract operations that specified a document (of any document type) have been submitted for printing until job submission has been completed; at that point this attribute shall then indicate the total number of printable documents, fonts, and resource objects submitted by the client in the job. If the first Print abstract operation does not contain a first document component, the value of this attribute shall be 0.~~

~~The server shall count fonts and resource objects passed to the server by means of Print abstract operation invocations, as documents for the purposes of this attribute.~~

~~NOTE the value of the number of documents attribute represents the total number of documents that the client has submitted to the server during the course of job submission, regardless of whether or not the client has cancelled any of the documents. See CancelJob abstract operation.~~

~~7.1.3.9 job submission complete~~

job submission complete	booleanSyntax	S	
------------------------------------	--------------------------	--------------	--

~~This attribute indicates whether all documents of the print job have been submitted (i.e., all Print abstract operations have been invoked for the job). The value FALSE indicates that more documents are expected to be submitted for the job, by means of additional print invocations.~~

7.1.3.9 impressions-completed

<u>impressions-completed</u>	<u>cardinalSyntax</u>	<u>S</u>	
------------------------------	-----------------------	----------	--

This attribute contains the number of impressions completed by the Output Device.

7.1.3.9 media-sheets-completed

<u>media-sheets-completed</u>	<u>cardinalSyntax</u>	<u>S</u>	
-------------------------------	-----------------------	----------	--

This attribute contains the number of media-sheets completed by the Output Device

7.1.4 ~~Job sheet~~ ~~Job Results Handling~~ Attributes

These attributes specify the actions to be undertaken ~~for printing of job sheets. after printing of a job has been completed. This includes assembly of documents into job sets, finishing operations applied to job sets, and delivery of the completed job sets.~~

The client may specify job-results-handling attributes in:

- a) Print: all
- b) ModifyJob: all
- c) ListObjectAttributes: all

7.1.4.1 job-sheets

job-sheets	enumnameOrOidSyntax	S	
------------	--------------------------------	---	--

~~This attribute determines what type of job-sheets the Output Device should print with the job. The possible enum values are: none, default and custom. 'None' means that no job sheets should be printed. 'Default' means that the job sheets defined by an administrator should be used. 'Custom' means that additional attributes, not defined in this document, determined whether start sheets, end sheets or slip sheets should be in the document~~

~~Attribute value types that encode identifiers that may have either a global form or a local form employ this generic syntax or datatype in their definitions.~~

~~The global form is of the object identifier type, and is expected to be used wherever such a value has been defined for the object in question. The local form is intended for local implementation convenience, for use when a global form is not available or has not been defined for the object to be identified.~~

~~NOTE — It must be stressed that the local form is not guaranteed to be unique, since there are no procedures in place to control the creation and usage of simple name types. It is possible for two different sites to create and use the same simple name to identify two different entities. If those two sites are interconnected subsequently, unexpected results can occur because of this duplication of simple names. For this reason, the local form is to be used only for purely local or temporary purposes; the global form must be used in all other cases.~~

~~This attribute specifies the auxiliary sheets that the server shall insert into the job as separators, covers, and trailers.~~

~~7.1.4.2 document sheets~~

document sheets	nameOrOidSyntax	S	
----------------------------	----------------------------	--------------	--

~~This attribute is similar to job sheets. The difference is that it applies to documents within the job rather than the job itself.~~

7.1.5 Job Event Handling Attributes

7.1.5.1 notification-profile

notification-profile	eventHandlingProfileSyntax	M	
----------------------	----------------------------	---	--

This attribute is a specification of events about which the user and/or designate are to be notified. In addition, this attribute specifies how the event notifications are to be delivered.

This attribute will support three events classes: job-completion only, job-problems-only, and job-completion-and-problems.

This attribute will support only one delivery method, namely email. This method of notification is quite deficient for timely notification to an end-user who receives a lot of email, but there are no other choices. The internet community needs to solve this problem, perhaps with an extremely-urgent email.

~~Printers may produce the same information for notification and logging or they may produce different information, depending on implementation.~~

7.1.6 Job Scheduling Instructions Attributes

These attributes provide additional hints for the scheduling of a print-job. How a print-service uses this information in scheduling jobs is implementation-specific.

The client may specify job-scheduling-instruction attributes in:

- a) Print: all
- b) ModifyJob: all
- c) ListObjectAttributes: all

7.1.6.1 job-hold

job-hold	booleanSyntax	S	
----------	---------------	---	--

This attribute specifies whether the print-job is a candidate for scheduling for printing or not, when the server would otherwise place the job in the pending or processing states. ~~The PauseJob and ResumeJob operations may be used independently of the value of this attribute.~~

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

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

job-state-reasons attribute and, if no other reasons remain, shall change the job's current-job-state to pending so that the job becomes a candidate for being scheduled on printer(s).

7.1.6.2 job-priority

priority	prioritySyntax	S	
----------	----------------	---	--

This attribute specifies a priority for scheduling the print-job. It is used by servers that employ a priority-based scheduling algorithm.

A higher value specifies a higher priority. The value 1 is defined to indicate the lowest possible priority (a job which a priority-based scheduling algorithm shall pass over in favour of higher priority jobs). The value 100 is defined to indicate the highest possible priority. Priority is expected to be evenly or 'normally' distributed across this range. The mapping of vendor-defined priority over this range is implementation-specific. The omission of this attribute implies that the user places no constraints concerning priority on the scheduling of the print-job.

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

7.1.6.3 job-print-off-peakafter

job-print-after	enumbooleangeneralizedTime Syntax	S	
-----------------	--	---	--

This attribute specifies whether or not a job should print during off-peak hours. If this attribute is present, it contains a value with which an administrator has associated allowable print times.

~~This attribute specifies the calendar date and time of day after which the print job shall become a candidate to be scheduled for printing.~~

~~If the value of this attribute is in the future, the server shall set the value of the job's current job state to held and add the job print after specified value to the job's job state reasons attribute and shall not schedule the print job for printing until the specified date and time has passed. When the specified date and time arrives, the server shall remove the job print after specified value from the job's job state reason attribute and, if no other reasons remain, shall change the job's current job state to pending so that the job becomes a candidate for being scheduled on printer(s).~~

~~The printer shall assign an empty value to the job print after attribute when no print after time has been assigned or when it does not support the notion of time within the printer, so that the job shall be a candidate for scheduling immediately.~~

7.1.6.4 job-retention-period

job-retention-period	deltaTimeSyntax	S	
----------------------	-----------------	---	--

Delta time provides an integer value for a period of elapsed time, measured in seconds.

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

Job-retention-period specifies a lower bound on how long job attributes, document attributes and document data shall be retained by a server after printing has completed, whilst job-discard-time sets an upper bound on retention of the job and document attributes independent of whether the job is ever scheduled for, starts or completes printing.

In addition to providing status information to a user after a job has completed printing, the job-retention-period also provides the mechanism for retaining job's document data after it has been printed, so that the job may be printed again, possibly with modified attributes, such as the job-copies component of the job-results attribute.

NOTE - The mechanism to reprint the job is outside the scope of this part of ISO/IEC 10175; part 3 of this International Standard (in preparation) includes a Resubmit abstract-operation to enable this function.

7.2 Document Attributes

~~This section discusses attribute that pertain to documents, A document object contains a set of document attributes, including the document-content attribute which specifies the document data. A document object may be of type printable, font, or resource as specified by the document's document type attribute. The printer shall create document objects as contained members of job objects in response to a client that performs one or more print requests (see 8.2.1). In addition, initial value document objects are created in a server by means outside the scope of this part of ISO/IEC 10175 in order to represent complete sets of default values for document attributes (see the initial value document object class). This subelause of ISO/IEC 10175 specifies the document attributes for both document and initial value document objects.~~

~~In addition to the attributes specifically defined for the document and initial value document objects, certain of the generic attributes may also be associated with these objects.~~

~~NOTE There are no attributes that apply to both the job and document objects. Thus the server may return both job and document attributes mixed together without ambiguity in the ModifyJob and CancelJob requests.~~

7.2.1 Document Description Attributes

These attributes identify the intended document format, ~~and its characteristics and specify the method by which the document is acquired by the print server.~~

The client may specify document-description attributes in:

- a) Print: ~~all, except document type, transfer method, and document content shall be passed as explicit parameters of the Print abstract operation and shall not be passed as attributes.~~
- b) ModifyJob: ~~noneall, except id att transfer method, and id att document content~~
- c) ListObjectAttributes: ~~all, except id att document content~~

7.2.1.1 document-format

document-format	docFormatSyntax	S	
-----------------	-----------------	---	--

This client specifies this attribute if the included documents are not yet in a format acceptable to an Output Device. This attribute specifies the format that the Printer shall translate the documents to in order to print.

Note: this attribute is rarely needed.

~~This attribute identifies the overall print document format used for the document. It consists of three elements, a document format, a document format variants and a document format version. The latter two elements are optional.~~

~~The document format element identifies a particular family of document formats, of which there may exist several versions or variants. The document format variants and document format version elements identify a specific instance of a document format. The variant refers to a particular functional subset of a format. For example, the format PostScript has variants of level 1 and level 2, and the format PCL has several variants, including PCL4 and PCL5.~~

~~The version distinguishes among successive releases of the same basic format and variant. For example, successive versions of Xerox Interpress include versions 2.0, 2.1, 3.0, 3.1, etc.~~

~~The document format variants element consists of a single text string. If it is necessary to identify more than one variant, the respective variant values shall all be contained in the document format variants element, separated from one another by commas.~~

~~If the client omits the document format variants or document format version elements, the server may supply a format specific default.~~

~~Proprietary values for the document format, document format variants, and document format version elements are assigned by the owners of those formats.~~

~~7.2.1.2 document content~~

document-content	documentContentSyntax	S	
-----------------------------	----------------------------------	--------------	--

~~This attribute specifies a transfer method specific reference for the document to be transferred. It indicates whether the content is included or referenced. If it is referenced, the reference is of syntax DOR. The DOR datatype (Distinguished Object Reference) is imported from ISO/IEC 10031-2. The DOR datatype may be used for other transfer methods, e.g., ftam by server.~~

~~7.2.1.3 Reserved~~

~~7.2.1.4 transfer method~~

transfer method	objectIdentifierSyntax	S	
----------------------------	-----------------------------------	--------------	--

~~This attribute identifies the method by which the document is transferred to or acquired by the print server.~~

~~Standard values are defined as: TBS.~~

~~Conforming client and server implementations shall support at least id-val transfer method with request, which is the default transfer method.~~

7.2.2 Document Production Instruction Attributes

These attributes provide information that affect the rendering and finishing of the document and are referred to as document production instructions (DPI). DPI may also be contained in the document to be printed.

After the information from these attributes has been folded into the document data, they are no longer relevant and can be discarded from a job. The resource attributes indicate from printer features a document needs in order to print correctly.

If there is a conflict between the value of one of these attributes, and a corresponding parameter found in the document (either implicit or explicit), the value of the attribute shall take precedence over the document parameter, ~~unless specifically mandated otherwise in the standard defining that document format.~~

~~All the default xxx attributes (e.g. default medium) specifically allow for the document contents to override the default xxx attribute under all conditions.~~

The client may specify document production-instruction attributes in:

- a) Print: all
- b) ModifyJob: all
- c) ListObjectAttributes: all

~~7.2.2.1 default font~~

2912

default-font	nameOrOidSyntax	S	
-------------------------	----------------------------	--------------	--

2913

2914

2915

2916

2917

2918

2919

2920

2921

2922

2923

2924

2925

2926

2927

2928

2929

2930

7.2.2.2 ~~default-medium-select~~

default-medium-select	nameOrOidSyntax	S	
----------------------------------	----------------------------	--------------	--

2931

2932

2933

2934

2935

2936

2937

2938

2939

2940

2941

2942

2943

2944

2945

2946

2947

2948

2949

2950

2951

2952

2953

2954

2955

2956

2957

2958

2959

2960

2961

2962

2963

2964

This attribute identifies the a-medium that the Printer server shall use as the medium default for all the pages of the document that require a specification, regardless of what media are specified within the document.

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

Standard values are defined: TBD

~~If the document data, itself, specifies media, such specification shall override the default medium attribute on a page by page basis. If the document data specifies media which are not also values of media used, then a printer may receive a document which requires media that are not ready. In such a case, an implementation may either abort the document or try printing the document on some alternative medium, such as the default medium.~~

~~A client has numerous ways to specify the media to be used when printing a document and different document pages can be specified in different ways. The client can specify the media in the document contents or with attributes. Some attributes override the document contents, and other attributes may be overridden by the document contents. In addition, the client can specify the media by name or by the input tray containing it.~~

~~Before printing each page of a document, the server determines the medium or input tray for that page by finding the first condition in the list of numbered steps below that is satisfied. For this discussion, either the medium or the input tray is sufficient information:~~

- ~~a) If page media select has a medium value for the current page, use that medium, regardless of document contents and other attributes.~~
- ~~b) If input tray select has a value, use that tray.~~
- ~~c) If the document contents specify a medium, and that medium is the same as the value of one of the original medium elements in the media substitution attribute, then use the corresponding substitution medium in the media substitution attribute.~~
- ~~d) If the document contents specify a medium, use that medium.~~
- ~~e) If the document contents specify an input tray, use that input tray.~~
- ~~f) If the default medium has a value, and the document format interpreter allows its use, and that medium is the same as the value of one of the original medium elements in media substitution attribute, then use the corresponding substitution medium in the media substitution attribute.~~
- ~~g) If the default medium has a value and the document format interpreter allows its use, use the default medium.~~
- ~~h) If the default input tray has a value and the document format interpreter allows its use, use the default input tray.~~
- ~~i) Use the medium or input tray selected by the document format processor in the printer. This selection is implementation dependent.~~

7.2.2.3 number-up

number-up	cardinalOrNameOrOidSyntax	S	
-----------	---------------------------	---	--

A CardinalSyntax allows attribute values that can specify either a Cardinal or an OID (that normally names a Cardinal).

This attribute specifies the number of source page-images to impose upon a single instance of a selected medium. The attribute can be specified either by a number directly or by naming an imposition object which specifies some particular number-up imposition.

In general, only certain numeric values are valid for this attribute, depending upon the server and printer implementations to which the print-request is directed. A value of 0 or none shall suppress any server default number up, if any.

This attribute primarily controls the translation, scaling and rotation of page images, but a site may choose to add embellishments, such as borders to each logical page. A site may even choose to add an attribute to control the presence or characteristics of such embellishments.

The following standard values are defined: id-val-generic-none, id-val-imposition-simple-1-up, id-val-imposition-simple-2-up, id-val-imposition-simple-4-up.

NOTE - The value 0 or none specifies that no convenience imposition functions shall be performed; 0 or none is needed to suppress any special number-up operation because a value of 1 for some sites may cause the server to alter the placement, or size of the page image, or to add embellishments, such as borders or to rotate the page depending on content-orientation.

The server may support three values for number-up besides 0 (and id-val-generic-none), namely 1 (and id-val-imposition-1-up), 2 (and id-val-imposition-simple-2-up) and 4 (and id-val-imposition-simple-4-up), which this document will reference by the respective names of 0-up, 1-up, 2-up and 4-up, henceforth. These 1-up, 2-up and 4-up values provide a simple means for users to request the printing of compact documents of a temporary or informal nature.

7.2.2.4 finishing

finishing	<u>enum</u> finishingSyntax	<u>SM</u>	
-----------	-----------------------------	-----------	--

This attribute identifies a sequence of one or more finishing-processes to be applied to each copy of the printed document.

Finishing encompasses the operations that may be applied to the media output of a print-job. Examples include stapling, saddle-stitching, hole-drilling, binding with tape, etc.

This attribute allows the requester to specify one or more individual finishing processes may be specified in the finishing attribute. Each of the individual processes is specified by including the required parameters for each of the individual finishing processes in the finishing attribute. Standard values for this attribute are defined: TBD.

7.2.2.5 sides

sides	<u>enum</u> sidesSyntax	S	
-------	-------------------------	---	--

This attribute specifies ~~the whether the document should be printed in one of three ways: 1-sided (simplex), 2-sided-flip-on-long-edge (duplex), 2-sided-flip-on-short-edge (tumble), number of printable surfaces of the medium to be imaged. SidesSyntax is an integer restricted to the range {1..2}.~~

7.2.2.6 copies-count

<u>copies</u> - <u>count</u>	cardinalSyntax	S	
------------------------------	----------------	---	--

This attribute specifies the number of copies of the job~~documents, or of the selected pages of the document,~~ to be printed.

A value of 1 for copies-count shall generate a single human perceptible copy of the electronic document. ~~If a value of 0 is supplied, then:~~

- ~~a) if the server supports specification of the value 0, the job shall be processed normally, but no print output shall be produced; or~~
- ~~b) if the server does not support specification of the value 0, the server shall return an unsupported attribute value AttributeError.~~

~~7.2.2.7 reset printer~~

3077

<u>reset-printer</u>	<u>booleanSyntax</u>	<u>S</u>	
----------------------	----------------------	----------	--

3078

3079

3080

3081

3082

3083

3084

3085

3086

3087

3088

3089

3090

3091

3092

3093

3094

3095

7.2.2.1 printer-resolution-select

<u>printer-resolution-select</u>	<u>cardinalSyntax</u>	<u>S</u>	
----------------------------------	-----------------------	----------	--

3096

3097

3098

3099

3100

3101

7.2.2.1 print-quality

<u>print-quality</u>	<u>enum</u>	<u>S</u>	
----------------------	-------------	----------	--

3102

3103

3104

3105

3106

3107

7.2.2.1 page-select

<u>page-select</u>	<u>integerRange</u>	<u>S</u>	
--------------------	---------------------	----------	--

3108

3109

3110

3111

3112

3113

3114

7.2.2.1 files-are-one-document (for the future 2 or more documents)

<u>files-are-one-document</u>	<u>booleanSyntax</u>	<u>S</u>	
-------------------------------	----------------------	----------	--

3115

3116

3117

3118

3119

3120

3121

3122

3123

3124

This attribute is relevant only if a job consists of two or more files. It controls finishing operations, and job-sheet placement.

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

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

7.2.2.1 files-are-interleaved (for the future 2 or more documents)

<u>files-are-interleaved</u>	<u>booleanSyntax</u>	<u>S</u>	
------------------------------	----------------------	----------	--

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

7.2.2 Attributes for Conversion of Text Files (set by client)

The attributes in this section specify formatting for text documents. If any of these attributes is not specified, a Printer shall use its own defaults.

7.2.2.1 width

<u>width</u>	<u>cardinalSyntax</u>	<u>S</u>	
--------------	-----------------------	----------	--

This attribute specifies the media width for the document.

7.2.2.1 length

<u>length</u>	<u>cardinalSyntax</u>	<u>S</u>	
---------------	-----------------------	----------	--

This attribute specifies the media length for the document.

7.2.2.1 left-margin

<u>left-margin</u>	<u>cardinalSyntax</u>	<u>S</u>	
--------------------	-----------------------	----------	--

This attribute specifies the left-margin for the document.

7.2.2.1 right-margin

<u>right-margin</u>	<u>cardinalSyntax</u>	<u>S</u>	
---------------------	-----------------------	----------	--

This attribute specifies the right-margin for the document.

7.2.2.1 top-margin

<u>top-margin</u>	<u>cardinalSyntax</u>	<u>S</u>	
-------------------	-----------------------	----------	--

This attribute specifies the top-margin for the document.

7.2.2.1 bottom-margin

3172

<u>bottom-margin</u>	<u>cardinalSyntax</u>	<u>S</u>	
----------------------	-----------------------	----------	--

3173

3174

This attribute specifies the bottom-margin for the document.

3175

3176

7.2.2.1 repeated-tab-stops

3177

<u>repeated-tab-stops</u>	<u>cardinalSyntax</u>	<u>S</u>	
---------------------------	-----------------------	----------	--

3178

3179

This attribute specifies the tab stops for the document.

3180

3181

7.2.2.1 header-text

3182

<u>header-text</u>	<u>Text</u>	<u>S</u>	
--------------------	-------------	----------	--

3183

3184

This attribute specifies the header text for the document.

3185

3186

7.2.2.1 footer-text

3187

<u>footer-text</u>	<u>Text</u>	<u>S</u>	
--------------------	-------------	----------	--

3188

3189

This attribute specifies the footer text for the document.

3190

3191

7.2.2.1 number-pages

3192

<u>number-pages</u>	<u>booleanSyntax</u>	<u>S</u>	
---------------------	----------------------	----------	--

3193

3194

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

3195

3196

3197

7.2.2.1 default-font

3198

<u>default-font</u>	<u>nameSyntax</u>	<u>S</u>	
---------------------	-------------------	----------	--

3199

3200

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

3201

3202

3203

7.2.2.1 default-character-set

3204

<u>default-character-set</u>	<u>enum</u>	<u>S</u>	
------------------------------	-------------	----------	--

3205

3206

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

3207

3208

3209

7.2.2.1 content-orientation

3210

<u>content-orientation</u>	<u>enum</u>	<u>S</u>	
----------------------------	-------------	----------	--

3211

3212

This attribute specifies the orientation of the document: landscape or portrait.

3213

3214

7.2.3 ~~Job Resource Document Characteristics~~ Attributes

This group of attributes describes the resources needed to print the job. ~~characteristics of the document to be printed.~~

The values provided by these attributes are intended to assist the print-server in validating and scheduling the print-job. Providing these attributes independent of the document allows the server to schedule a job or to validate the resources required to print the document without interpreting the contents of the document. This provides the opportunity for a server to support a broad set of document formats yet still support fast efficient scheduling and validation of each job. ~~The values provided by these attributes are also intended to provide parameters to print server services, such as a text formatter or imposition's number up procedure.~~

The values of these attributes are hints to the server about production instructions and resources needed to print a document, but the printer does not use these attributes during the actual printing of a document. It is the duty of the process that translates the document to the printer's PDL to provide these values. ~~If such values are lacking, the Printer shall assume that the document doesn't ask for any resources that are unavailable. Such value may be missing if the translation process fails to provides such values, or if no translation occurs (e.g. the document is a PostScript document. The values of these attributes are intended to come from the document content, but some may come from intentions of the client. The values of these attributes are assigned as follows:~~

- ~~a) First the client may, at its option, either omit these attributes, or assign values to any of them based on the document content or client intent.~~
- ~~b) The server may then, at its option, either leave any of these attributes unchanged, or assign values to any of them based on its own analysis of the document contents. If the document contains incomplete information or no information about the attribute, or the server cannot ascertain the information, the server may choose to assign some default value. For xxx used attributes which have a corresponding default xxx, the server shall use the value of the default xxx as the default. When a default is used with a MULTI-VALUE attribute, it may be one of several values in the attribute, e.g. some pages may have an explicit medium, others may use a default.~~
- ~~c) Finally, the server may choose to assign a value to these attributes only when the client does not supply a value, or the server may choose to override whatever the client supplies, or the server may also choose to do nothing, regardless of what the client supplies.~~

~~If the client performs the ModifyJob operation on any of these attributes, the server shall follow rules b and c, above, for the modified attributes. Thus, in effect, the server shall have control over whether to honor a client's requested change.~~

~~For validation and scheduling, the server shall use these attributes and shall not examine the document contents. However, according to the rules above, the server may have examined the document contents earlier to assign values to these attributes.~~

~~Processes, such as text formatting and number up may use some of these attributes as parameters, or they may do their own independent analysis during the procedure.~~

The client may specify document-characteristic attributes in:

- a) Print: all (translation process, not the end-user)
- b) ModifyJob: none all
- c) ListObjectAttributes: all

7.2.3.1 document-format-used

<u>document-format-used</u>	<u>docFormatSyntax</u>	<u>S</u>	
-----------------------------	------------------------	----------	--

This attribute identifies the document format needed to print this job.

This attribute identifies the overall print document format used for the document. It consists of three elements, a document-format, a document-format-variants and a document-format-version. The latter two elements are optional.

[The format needs to be simplified, but I'll leave in the text below for now.]

The document-format element identifies a particular family of document formats, of which there may exist several versions or variants. The document-format-variants and document-format-version elements identify a specific instance of a document format. The variant refers to a particular functional subset of a format. For example, the format PostScript has variants of level 1 and level 2, and the format PCL has several variants, including PCL4 and PCL5.

The version distinguishes among successive releases of the same basic format and variant. For example, successive versions of Xerox Interpress include versions 2.0, 2.1, 3.0, 3.1, etc.

The document-format-variants element consists of a single text string. If it is necessary to identify more than one variant, the respective variant values shall all be contained in the document-format-variants element, separated from one another by commas.

If the client omits the document-format-variants or document-format-version elements, the server may supply a format-specific default.

Proprietary values for the document-format, document-format-variants, and document-format-version elements are assigned by the owners of those formats.

7.2.3.1 fonts-used

<u>fonts-used</u>	<u>fontReferenceSyntax</u>	<u>M</u>	
-------------------	----------------------------	----------	--

This attribute identifies the font resources used ~~specified~~ in the job~~document~~.

7.2.3.1 character-sets-used

<u>character-set-used</u>	<u>enums</u>	<u>S</u>	
---------------------------	--------------	----------	--

This attribute identifies the character-set used in the document. This attribute is relevant only for files that are not in ASCII, such as text files and possibly PCL files. PostScript files are always ASCII.

7.2.3.2 media-used

media-used	name OrOidSequence Syntax	<u>MS</u>	
------------	--------------------------------------	-----------	--

This attribute identifies the media, input-trays or electronic forms needed to print the job.~~specified in the document.~~

~~The values in this attribute should contain the actual media required for printing the document, taking into account the results of interpreting the document contents, and applying the attributes: page media select, input tray select, media substitution, default medium and default input tray.~~

Standard values for this attribute are defined: TBD

~~This attribute contains a SEQUENCE of values rather than a SET because the ith element of this attribute corresponds to the ith attribute of the assured-reproduction-areas-used attribute.~~

This attribute is intended for scheduling and validation. The server uses this attribute with the printer attributes media-supported for validation and media-ready for scheduling.

7.2.3.2 sides-used

<u>sides-used</u>	<u>enum</u>	<u>S</u>	
-------------------	-------------	----------	--

This attribute specifies whether a job needs simplex, duplex or tumble printing.

7.2.3.2 output-bin-used

<u>output-bin-used</u>	<u>enum</u>	<u>S</u>	
------------------------	-------------	----------	--

This attribute specifies what output bins the job needs.

7.2.3.2 print-quality-used

<u>print-quality-used</u>	<u>enum</u>	<u>S</u>	
---------------------------	-------------	----------	--

This attribute specifies what print quality the job needs.

7.2.3.2 finishing-used

<u>finishing-used</u>	<u>enum</u>	<u>S</u>	
-----------------------	-------------	----------	--

This attribute specifies what finishing the job needs.

7.2.3.2 printer-resolution-used

<u>printer-resolution-used</u>	<u>cardinalSyntax</u>	<u>S</u>	
--------------------------------	-----------------------	----------	--

This attribute specifies what resolution the job needs. This attribute is the first of three that a client can use to specify the size of a job.

7.2.3.2 total-job-octets

<u>total-job-octets</u>	<u>nameSyntax</u>	<u>S</u>	
-------------------------	-------------------	----------	--

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

7.2.3.2 job-impression-count

<u>job-impression-count</u>	<u>nameSyntax</u>	<u>S</u>	
-----------------------------	-------------------	----------	--

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

7.2.3.2 job-media-sheet-count

<u>job-media-sheet-count</u>	<u>nameSyntax</u>	<u>S</u>	
------------------------------	-------------------	----------	--

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

7.2.4 Document Contents (one per document)

7.2.1 Document Description Attributes

These attributes name and reference the individual documents in a job.

The client may specify document-description attributes in:

- a) Print: all, except document-type, transfer-method, and document-content shall be passed as explicit parameters of the Print abstract-operation and shall not be passed as attributes.
- b) ModifyJob: none
- c) ListObjectAttributes: all, except id-att-document-content

7.2.1.1 number-of-documents

<u>number-of- documents</u>	<u>cardinalSyntax</u>	<u>S</u>	
---------------------------------	-----------------------	----------	--

This attribute specifies the number of documents in the job.

7.2.1.2 document-content

<u>document- content</u>	<u>documentContentSyntax</u>	<u>S</u>	
------------------------------	------------------------------	----------	--

This attribute is a sequence with one element for each document in the job. Each element contains the following fields:

- document-name: the name of the document specified by the end-user.
- file-reference: the path to the current location of the file
- URL: the URL of the file.
- contents: actual contents of the file.

Of the last three fields, only one should be present at any time.

~~7.2.4 Document Status Attributes~~

~~These attributes specify the document status, before, during, and after processing of the document by the server. The server shall create the document object with these attributes (if implemented) and shall assign appropriate values to each such document status attribute.~~

~~The client may specify document status attributes in:~~

- ~~—— a) Print: none~~
- ~~—— b) ModifyJob: none~~
- ~~—— c) ListObjectAttributes: all~~

~~7.2.4.1 document-sequence-number~~

document- sequence-number	cardinalSyntax	S	
--	---------------------------	--------------	--

~~This attribute specifies the number of this document in relation to the set of documents in this job. The first document in the job is numbered 1.~~

~~The document-sequence-number is not passed as an input attribute in the Print abstract operation. Documents are assumed to be submitted in order (i.e., document number 1 followed by document number 2, etc.).~~

~~A server shall return a value of 0 for this attribute if the first Print abstract operation has not submitted a document (e.g., the first document element is omitted in the create job element of the Print abstract operation).~~

7.3 Operation Attributes

TBD

7.4 Printer Attributes

A printer object may represent either a Print Server or Output Device physical printer or a logical printer, or both.

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

~~A Printer Object in a Print Server logical printer is a printer object~~ containing a set of printer object attributes that are the union of the Printer objects in the downstream Output Devices. ~~have been grouped under one name in order to represent some class of printer or printing effect. This object that extends the capabilities of an Output Device.~~ For example, an administrator might define a single Print Server logical printer to represent all of the Output Devices physical printers of the same type and capability in a single location, associated with a particular server. A user/client would normally send a print-job to a Print Server logical printer, and allow the Print Server server to assign the job to a particular Output Device physical printer based on the relative load and availability of the printers under its control, thus providing a load balancing service. However, ~~nothing ISO/IEC 10175 does not~~ precludes a user/client from sending a print-job to an Output Device physical printer. Such a restriction is up to the policy of the system administrator and the access control that the administrator specifies.

~~Logical and physical printer objects may be defined to specify that a particular set of default values for job and document attributes are to be assumed when a client identifies that printer. Such things as default media, fonts, finishing operations, etc., may be specified for a job simply by sending the job to a particular logical or physical printer. When the client identifies a logical printer, the server shall assign the job to a particular physical printer that the administrator has explicitly associated with the logical printer. Depending on implementation, the server may assign the job when the job is received, or the server may delay the assignment, until a physical printer is free, thereby achieving more dynamic load balancing between several physical printers.~~

~~A printer object shall have one of three realizations, as specified by the value of its printer realization attribute logical, physical, or logical and physical. The logical and physical value is used in the simple and frequent case when the system administrator creates a single printer object to represent both a logical printer and a physical printer. This would be the case when a single physical printer is associated with a single print server or when the administrator has decided not to offer additional sets of defaults for the physical printer. In order to create more than one set of defaults for a physical printer, the system manager shall create an associated logical printer and sets its printer realization to logical.~~

~~If the printer realization attribute is not implemented, the server shall treat all printer objects as if the printer realization attribute had the value logical and physical.~~

~~Throughout ISO/IEC 10175, the term printer shall refer to both logical and physical printers, and shall be used when no distinction is being made between logical and physical printers. The term logical printer shall be used for a printer object whose printer realization attribute has the value logical or logical and physical. The term physical printer shall be used for a printer object whose printer realization attribute has the value physical or logical and physical.~~

The attributes defined in this subclause provide information about a particular Print Server or Output Device logical or physical printer; all of the attributes apply to Print Servers and Output Devices logical and physical printers.

~~In addition to the attributes specifically defined for this object, certain of the generic attributes may also be associated with this object. For example, when requesting a list of attribute values for an object of this class, the client may identify one or more of the generic attributes in the following table, for which the server shall return values if the attributes are implemented.~~

7.4.1 printer-name

printer-name	simpleNameSyntax	S	
--------------	------------------	---	--

This attribute uniquely identifies the printer.

7.4.2 printer-state

printer-state	objectIdentifierSyntax	S	
---------------	------------------------	---	--

This attribute identifies the current state of the printer. The LDPA protocol support all values for printer states, however printers are not required to generate all the printer states, only those which are appropriate for the particular implementation.

The following standard values are defined:

id-val-printer-state-unknown, id-val-printer-state-idle,
id-val-printer-state-printing,
id-val-printer-state-needs-attention, id-val-printer-state-paused,
id-val-printer-state-shutdown,
id-val-printer-state-job-start-wait,
id-val-printer-state-job-end-wait,
id-val-printer-state-job-password-wait,
id-val-printer-state-needs-key-operator,
id-val-printer-state-connecting-to-printer,
id-val-printer-state-timed-out

7.4.2 printer-state-reasons

3597

<u>printer-state-reasons</u>	<u>enum</u>	<u>S</u>	
------------------------------	-------------	----------	--

3598

3599

This attributes specifies reasons for being in a printer-state.

3600

3601 7.4.3 message

3602

message	messageSyntax	S	
---------	---------------	---	--

3603

3604

3605

3606

3607

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

3608

3609

~~7.4.4 printer initial value job~~

3610

printer-initial-value-job	nameOrOidSyntax	S	
--------------------------------------	----------------------------	--------------	--

3611

3612

3613

3614

3615

3616

3617

3618

3619

3620

3621

~~This attribute identifies an initial value job object in the server for this printer. An initial value job object contains those attributes that the server shall default when a print job is submitted, if the client does not specify an initial value job attribute with the print request, the server shall use the initial value job object specified by the printer's printer initial value job attribute to initialize the job object when the job is submitted and set the job's initial value job attribute to the value of the printer initial value job attribute.~~

3622

3623

3624

3625

3626

3627

3628

~~In an initial value job object, SINGLE VALUE attributes (1) shall contain one attribute value or (2) may specify no attribute values, i.e., an empty attribute value (see DPA 9.1.2). MULTI VALUE attributes shall contain zero or more attribute values. Attributes containing no values either (1) are not supported by the printer, or (2) are expected to be defaulted by the printer hardware itself.~~

3629

3630

3631

3632

3633

3634

~~LDPA requires that a printer shall implement the printer initial value job attribute. This requirement is important, so that the server defaulting mechanism shall permit a client to submit a print job with many attributes omitted, and the server supplies default values.~~

3635

3636

~~7.4.5 printer initial value document~~

3637

3638

printer-initial-value-document	nameOrOidSyntax	S	
---	----------------------------	--------------	--

3639

3640

3641

3642

3643

3644

3645

~~This attribute identifies an initial value document object in the server for this printer. If the client does not specify an initial value document attribute with the print request, the server shall use the initial value document object specified by the printer's printer initial value document attribute to initialize the document object when the document is submitted and~~

~~set the document's initial value document attribute to the value of the printer initial value document attribute.~~

~~A printer may specify only one initial value document object, which will be used to initialize all document object instances targeted at this printer unless overridden by the initial value document attribute as described above. Each document in a job may therefore use a different initial value document object even though the printer may specify only one.~~

~~In an initial value document object, SINGLE VALUE attributes (1) shall contain one attribute value or (2) may specify no attribute values, i.e. an empty attribute value (DPA see 9.1.2). MULTI VALUE attributes shall contain zero or more attribute values. Attributes containing no values either (1) are not supported by the printer, or (2) are expected to be defaulted by the printer hardware itself.~~

~~LDPA requires that a printer shall implement the printer initial value document attribute. This requirement is important so that the server defaulting mechanism shall permit a client to submit a document print request with many attributes omitted, and the server supplies default values.~~

7.4.6 fonts-supported

fonts-supported	fontReferenceSyntax	M	
-----------------	---------------------	---	--

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

~~7.4.7 fonts-ready~~

fonts-ready	fontReferenceSyntax	M	
------------------------	--------------------------------	--------------	--

~~This attribute identifies the font resources currently ready to be used on this printer.~~

7.4.8 media-supported

media-supported	nameOrOidSyntax	M	
-----------------	-----------------	---	--

This attribute identifies the media, media-sizes, input trays, and electronic forms supported by this printer, and indicates the state of readiness for each resource. There may be just two states: ready and needs-installing, or there may be a third state: needs-purchasing.

~~7.4.9 media-ready~~

media-ready	nameOrOidSyntax	M	
------------------------	----------------------------	--------------	--

~~This attribute identifies the media currently ready to be used on this printer.~~

~~7.4.10 printer-associated-printers~~

printer-associated-printers	distinguishedNameStringSyntax	M	
--	--	--------------	--

~~This attribute identifies the logical/physical printers associated with this physical/logical printer.~~

7.4.11 document-formats-supported

document-formats-supported	docFormatSyntax	M	
----------------------------	-----------------	---	--

This attribute identifies the document-formats, including the document-format-variants and document-format-versions, supported by the Output Device and the server software collectively. This set includes both the formats that are native to the Output Device and those formats that the server software can translate to one that is native to the Output Device. From the client's point of view, this set contains all formats in which documents can be submitted to this printer.

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

7.4.12 numbers-up-supported

numbers-up-supported	numbersUpSupportedSyntax	S	
----------------------	--------------------------	---	--

This attribute identifies the number-up values and imposition objects supported by this printer. The cardinal-range is an alternative (shorthand) way of specifying consecutive cardinal-values.

There are no standard values defined.

7.4.13 finishings-supported

finishings-supported	nameOrOidSyntax	S	
----------------------	-----------------	---	--

This attribute identifies the per-document finishing objects supported by this printer, that is the server-installed finishing objects that may be used as values of the finishing document attribute.

NOTE: What are the values of this attribute since we have no Finishing objects.

7.4.14 sides-supported

sides-supported	sidesSyntax	M	
-----------------	-------------	---	--

This attribute indicates the values of the sides attribute supported by this printer, i.e., the different numbers of surfaces of a medium that can be imaged by this printer.

7.4.15 job-sheets-supported

job-sheets-supported	nameOrOidSyntax	M	
----------------------	-----------------	---	--

This attribute identifies the auxiliary-sheet-s values supported by this printer.

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

If the job-sheets attribute is not specified or contains a value which the printer does not support, and the job-sheets value is non-compulsory (so that the server accepts the job), then the server may select from among the values of this attribute. The server shall not select the value id-val-generic-none unless it is the only value specified for the job-sheets-supported attribute.

NOTE - It is preferable for the server to produce some job auxiliary-sheet, even if not the desired one, rather than produce none at all.

7.4.16 document-sheets-supported

document-sheets-supported	nameOrOidSyntax	M	
---------------------------	-----------------	---	--

This attribute identifies the auxiliary-sheets values supported by this printer.

To allow no document sheets, the system administrator shall include the value id-val-generic-none as a value for this attribute. The client specifies that there are no document sheets by using the value id-val-generic-none as the value of the document-sheets attribute.

If the document-sheets attribute is not specified or contains a value which the printer does not support, and the document-sheets value is non-compulsory (so that the server accepts the job), then the server may select from among the values of this attribute. The server shall not select the value id-val-generic-none unless it is the only value specified for the document-sheets-supported attribute.

NOTE - It is preferable for the server to produce some job auxiliary-sheet, even if not the desired one, rather than produce none at all.

7.4.17 maximum-copies-supported

maximum-copies-supported	cardinalSyntax	S	
--------------------------	----------------	---	--

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

A server shall ensure that neither a document's copy-count attribute nor any single job-copies element of a ResultsProfile exceeds the value specified in this attribute. A server may ensure that for each document the product of the document's copy-count and the sum of all job-copies in all result-sets does not exceed this value.

A value of 0 shall indicate there is no limit on the maximum number of document copies for this printer.

7.4.18 notification-delivery-methods-supported

notification-delivery-methods-supported	TBD	S	
---	-----	---	--

7.4.19 physical-printers-supported

physical-printers-supported	distinguishedNameStringSyntax	M	
-----------------------------	-------------------------------	---	--

This attribute identifies the physical printers (printer's realization attribute is either physical or logical-and-physical) supported by this server.

7.4.20 Logical-printers-supported

logical-printers-supported	distinguishedNameStringSyntax	M	
----------------------------	-------------------------------	---	--

This attribute identifies the logical printers (printer's realization attribute is either logical or logical-and-physical) supported by this server.

7.4.21 events-supported

events-supported	objectIdentifierSyntax	S	
------------------	------------------------	---	--

This attribute identifies the event types and event classes supported by this printer.

7.4.22 transfer-methods-supported

transfer-methods-supported	objectIdentifierSyntax	M	
----------------------------	------------------------	---	--

This attribute identifies the transfer-methods supported by this server.

7.4.23 locales-supported

TBD

7.4.24 multiple-documents-supported

multiple-documents-supported	booleanSyntax	S	
------------------------------	---------------	---	--

This attribute indicates whether this object (printer or server) is capable of processing and printing multiple documents per job.

This printers shall not support any operation involving multiple documents unless this attribute has the value TRUE. In spite of this requirement, it is still a printer driver implementation option of whether to support modifying and/or cancelling individual documents within a multi-document job or not.

7.4.28 cancel-individual-document-supported

cancel-individual-document-supported	booleanSyntax	S	
--------------------------------------	---------------	---	--

This attribute indicates whether this object (printer or server) is capable of cancelling the printing of individual documents within a multiple document job.

7.4.29 modify-individual-document-supported

modify-individual-document-supported	booleanSyntax	S	
--------------------------------------	---------------	---	--

This attribute indicates whether the server is capable of modifying the print-request parameters for individual documents within a multiple document job.

7.5 Initial Value Job Attributes

The attributes for an Initial Value Job object can be any of the Job object attributes defined in section 7.1.

7.6 Initial Value Document Attributes

The attributes for an Initial Value Document object can be any of the Document object attributes defined in section 7.2.

7.7 Relationship to ISO/IEC 10175 Conformance Levels

In ISO/IEC 10175 DPA Appendix E, three Conformance Levels are defined. For levels 1 and 2, an additional set of attributes for multiple-document job support are defined. These additional levels are indicated by the letter M. Thus, level 2M indicates support for a basic set of operations and attributes with

additional support for multiple-document jobs. The scope of LDPA is essentially the same as level 2M as defined by DPA.

LDPA is explicitly designed to be extensible. This means that in addition to the attributes defined in this specification, specific implementation instances may support not only the basic protocol as defined in this specification, but might add vendor specific extensions.

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

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

8. Security Considerations

This protocol does not identify any new security mechanisms. The authentication mechanisms (as well as extensions) built into the RPC infrastructure are recommended. Also, the Bind operation described in section 5 supports the notion of authentication via simple or credential based arguments.

9. References

- [1] Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759, March 1995.
- [2] Srinivasan, R., "RPC: Remote Procedure Call Protocol Specification Version 2", RFC 1831, August 1995.
- [3] Srinivasan, R., "XDR: External Data Representation Standard", RFC 1832, August 1995.
- [4] Postel, J., "Instructions to RFC Authors", RFC 1543, October 1993.
- [5] ISO/IEC 10175 Document Printing Application (DPA)
- [6] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August 1995.
- [7] Kirk, M. (editor), POSIX System Administration -- Part 4: Printing Interfaces, POSIX 1387.4 D8, 1994.
- [8] ISO 8824 "Abstract Syntax Notation One ASN.1"
- [9] ISO 8825 "Basic Encoding Rules BER"
- [10] McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.

10. Author's Address

3952
3953 Scott A. Isaacson
3954 Novell, Inc.
3955 122 E 1700 S
3956 Provo, UT 84606
3957
3958 Phone: 801-861-7366
3959 Fax: 801-861-4025
3960 EMail: scott_isaacson@novell.com
3961
3962 F. Devon Taylor
3963 Novell, Inc.
3964 122 E 1700 S
3965 Provo, UT 84606
3966
3967 Phone: 801-861-7179
3968 Fax: 801-861-4025
3969 EMail: devon_taylor@novell.com
3970
3971 Mike MacKay
3972 Novell, Inc.
3973 2180 Fortune Dr.
3974 San Jose, CA 95131
3975
3976 Phone: 408-577-6368
3977 Fax: 408-577-5151
3978 EMail: mmackay@novell.com
3979
3980 Peter Zehler
3981 Xerox Corporation
3982 800 Phillips Rd. M/S 111-01X
3983 Webster, NY 14580
3984
3985 Phone: 716-265-8755
3986 Fax: 716-265-8792
3987 EMail: peter_zehler@wb.xerox.com
3988
3989 Carl-Uno Manros
3990 Xerox Corporation
3991 701 S. Aviation Blvd.
3992 El Segundo, CA 90245
3993
3994 Phone: 310-333-8273
3995 Fax: 310-333-5514
3996 EMail: cmanros@cp10.es.xerox.com
3997
3998 Tom Hastings
3999 Xerox Corporation
4000 701 S. Aviation Blvd.
4001 El Segundo, CA 90245
4002
4003 Phone: 310-333-6413
4004 Fax: 310-333-5514
4005 EMail: hastings@cp10.es.xerox.com
4006
4007 Robert Herriot
4008 Sun Microsystems Inc.
4009 2550 Garcia Ave., MPK-17
4010 Mountain View, CA 94043
4011

4012 Phone: 415-786-8995
4013 Fax: 415-786-
4014 Email: robert.herriot@eng.sun.com
4015