Project of the PWG-IPP Working Group

Internet Printing Protocol (IPP): Production Printing Attributes – Set 2

There are 4 issues in this document which are highlighted like this and are TBDs dealing with registration with IANA. Also need to format the document along the lines of the PWG Media standard which follows the ISTO suggested format.

Draft D0.1

August 21, 2002

ftp://ftp.pwg.org/pub/pwg/ipp/new_PPE/pwg-ipp-prod-print-set2-draft-v0_92-020821.doc, .rtf, .pdf

Abstract

This document specifies an extension to the Internet Printing Protocol/1.0 (IPP) [RFC2565, RFC2566] and IPP/1.1 [RFC2910, RFC2911]. This extension consists primarily of OPTIONAL Job Template for submitting print jobs primarily (but not limited) to production printers). These Job Template attributes permit a user to save jobs for later reprinting, provide a recipient name and a job phone number, provide the feed orientation, provide the font name and font size, hold a job until a specific date and time, specify an interpreter initialization file, and print a proof print(s) of the job prior to printing the full run of the job.

There are also Printer Description attributes to list the Job Creation attributes supported, indicate whether jobs are spooled, and list the set of media collections supported.

In addition, semantics for Attribute Precedence, a Queue Override capability, and the capability to guarantee protocol precedence over the PDL attribute are defined.

This document is a draft of an IEEE-ISTO PWG Proposed Standard and is in full conformance with all provisions of the PWG Process (see ftp://ftp.pwg.org/pub/pwg/general/pwg-process.pdf). PWG Proposed Standards are working documents of the IEEE-ISTO PWG and its working groups. The list of current PWG projects and drafts can be obtained at http://www.pwg.org.

Copyright (C) 2002, IEEE Industry Standards and Technology Organization. All rights reserved.

TABLE OF CONTENTS

1 INTRODUCTION 8

2 TERMINOLOGY 9

2.1 Conformance Terminology 9

2.2 Other terminology 9

3 SEMANTICS COMMON TO MORE THAN ONE ATTRIBUTE 11

3.1 Guaranteed Protocol Precedence Over the PDL and Queue Overrides 11

3.2 Attribute Precedence 11

3.3 Job Save and Reprint Capability 13
3.3.1 Reprinting Saved Jobs 13
3.3.1.1 Reprinting using the Print-URI operation 13

4 OPERATION ATTRIBUTES 14

4.1 job-password (octetString(255)) 14

4.2 job-password-encryption (type3 keyword | name(MAX)) 15

5 JOB TEMPLATE ATTRIBUTES 15

5.1 feed-orientation (type3 keyword) 15
5.1.1 feed-orientation-default (type3 keyword) 16
5.1.2 feed-orientation-supported (1setOf (type3 keyword)) 16

5.2 font-name-requested (name(MAX)) 16
5.2.1 font-name-requested-default (name(MAX)) 16
5.2.2 font-name-requested-supported (1setOf name(MAX)) 16

5.3 font-size-requested (integer (1:MAX)) 17
5.3.1 font-size-requested-default (integer (1:MAX)) 17
5.3.2 font-size-requested-supported (1setOf rangeOfInteger (1:MAX)) 17

5.4 job-hold-until-time (dateTime) 18
5.4.1 Why there is no job-hold-until-time-default 19
5.4.2 job-hold-until-time-supported (rangeOfInteger(0:MAX)) 19

5.5 job-phone-number (text(127)) 20
5.5.1 job-phone-number-default (text(127)) 20
5.5.2 job-phone-number-supported (boolean) 20

5.6 job-recipient-name (name(MAX)) 21
5.6.1 job-recipient-name-default (name(MAX)) 21
5.6.2 job-recipient-name-supported (boolean) 21

5.7 job-save-disposition (collection) 21
5.7.1 Member Attributes for the "job-save-disposition" Job Template Attribute 21
5.7.1.1 save-disposition (type3 keyword) 22
5.7.1.1.1 save-disposition-supported (1setOf type3 keyword) 23
5.7.1.2 save-info (1setOf collection) 23
5.7.1.2.1 There is no save-info-default Printer attribute 24
5.7.1.2.2 save-info-supported (1setOf type2 keyword) 24
5.7.1.2.3 Member Attributes for the "save-info" member attribute 24
5.7.1.2.3.1 save-location (uri) 24
5.7.1.2.3.1.1 save-location-default (uri) 26
5.7.1.2.3.1.2 save-location-supported (1setOf uri) 26
5.7.1.2.3.2 save-name (name(MAX)) 26
5.7.1.2.3.2.1 There is no “save-name-default” Printer attribute 27
5.7.1.2.3.2.2 save-name-supported (boolean) 27
5.7.1.2.3.3 save-document-format (mimeMediaType) 28
5.7.1.2.3.3.1 save-document-format-default (mimeMediaType) 29
5.7.1.2.3.3.2 save-document-format-supported (1setOf mimeMediaType) 29
5.7.1.2.3.4 Printer actions for various combinations of attributes 29
5.7.2 job-save-disposition-default (collection) 30
5.7.3 job-save-disposition-supported (1setOf type2 keyword) 31

5.8 pdl-init-file (1setOf collection) 31
5.8.1 Member Attributes for the "pdl-init-file" Job Template Attribute 32
5.8.1.1 pdl-init-file-location (uri) 32
5.8.1.1.1 pdl-init-file-location-supported (1setOf uri) 32
5.8.1.2 pdl-init-file-name (name(MAX)) 32
5.8.1.2.1 pdl-init-file-name-supported (1setOf name(MAX)) 33
5.8.1.3 pdl-init-file-entry (name(MAX)) 33
5.8.1.3.1 pdl-init-file-entry-supported (1setOf name(MAX)) 33
5.8.2 pdl-init-file-default (1setOf collection) 33
5.8.3 pdl-init-file-supported (1setOf type2 keyword) 33

5.9 proof-print (collection) 33
5.9.1 Member Attributes for the "proof-print" Job Template Attribute 34
5.9.1.1 proof-print-copies (integer (0:MAX)) 34
5.9.1.2 media (type3 keyword | name(MAX)) or media-col (collection) 35
5.9.2 proof-print-default (collection) 35
5.9.3 proof-print-supported (1setOf type2 keyword) 35

6 JOB DESCRIPTION ATTRIBUTES 36

6.1 job-printer-make-and-model (text(127)) 36

7 PRINTER DESCRIPTION ATTRIBUTES 36

7.1 job-creation-attributes-supported (1setOf type2 keyword) 36

7.2 job-password-supported (integer(0:127)) 37

7.3 job-password-encryption-supported (1setOf (type3 keyword | name(MAX))) 37

7.4 job-spooling-supported (type2 keyword) 38

7.5 max-save-info-supported (integer(1:MAX)) 38

7.6 media-col-database (1setOf collection) 39

7.7 printer-detailed-status-messages (1setOf text(MAX)) 39

7.8 which-jobs-supported (1setOf type2 keyword) 39

8 ADDITIONAL VALUES AND SEMANTICS FOR EXISTING IPP ATTRIBUTES 39

8.1 Additional values and semantics for the "pdl-override-supported" Printer Description attribute 40

8.1.1 pdl-override-supported (type2 keyword) 40

8.2 Additional values for the "which-jobs" Operation attribute and the “which-jobs-supported” Printer Description attribute 43

8.2.1 which-jobs (type2 keyword) 44

8.3 Additional values and semantics for the "job-state-reasons" Job Description attribute 45

8.3.1 job-state-reasons (1setOf type2 keyword) 45

8.4 Additional semantics for the IPP "media-col" Job Template Attribute 51

8.4.1 media-tooth (type3 keyword | name(MAX)) 51

8.4.1.1 media-tooth-supported (1setOf (type3 keyword | name(MAX))) 52

8.4.2 media-grain (type3 keyword | name(MAX)) 52

8.4.2.1 media-grain-supported (1setOf (type3 keyword | name(MAX))) 52

8.4.3 media-material (type3 keyword | name(MAX)) 53

8.4.3.1 media-material-supported (1setOf (type3 keyword | name(MAX))) 53

8.4.4 media-thickness (integer(1:MAX)) 53

8.4.4.1 media-thickness-supported (rangeOfInteger(1:MAX)) 53

9 CONFORMANCE REQUIREMENTS 54

9.1 Conformance Requirements for Printer objects 54

9.2 Conformance Requirements for clients 54

9.3 Conformance Requirements for the Job Save and Reprint Capability 54
9.3.1.1 Client Conformance Requirements for Job Save and Reprint Capability 55
9.3.1.2 Printer Conformance Requirements for Job Save and Reprint Capability 55
9.3.1.3 Job Save Format Specification Conformance Requirements 56

10 IANA CONSIDERATIONS 56

10.1 Attribute Registration 56

10.2 Attribute Value Registration 57

11 INTERNATIONALIZATION CONSIDERATIONS 58

12 SECURITY CONSIDERATIONS 58

13 NORMATIVE REFERENCES 58

14 INFORMATIVE REFERENCES 59

15 AUTHOR'S ADDRESSES 59

16 APPENDIX B: SUMMARY OF OTHER IPP DOCUMENTS 60

17 APPENDIX C: DESCRIPTION OF THE IEEE INDUSTRY STANDARDS AND TECHNOLOGY (ISTO) 61

18 APPENDIX D: DESCRIPTION OF THE IEEE-ISTO PWG 62

Table of Tables

Table 1 - Summary of Attributes defined 8
Table 2 - "job-save-disposition" member attributes 21
Table 3 - "save-info" member attributes 24
Table 4 - Printer actions for various Job and Printer attribute values 30
Table 5 - "pdl-init-file" member attributes 32
Table 6 - "proof-print" member attributes 34
Table 7 - Rules for 'guaranteed' value of "pdl-override-supported" attribute 42
Table 8 - Values of "job-state-reasons" attribute for various job conditions 50
Table 9 – Additional "media-col" member attributes 51

1 Introduction

This document specifies an extension to the Internet Printing Protocol/1.0 (IPP) [RFC2565, RFC2566] and IPP/1.1 [RFC2910, RFC2911]. This extension consists primarily of OPTIONAL Job Template attributes defined for submitting print jobs primarily (but not limited to) to production printers and Printer Description attributes as summarized in Table 1. All of these Job Template attributes are OPTIONAL for a Printer to support. However, some of these Job Template attributes do require other Job Template attributes in this document to be supported. See the Conformance section (section 9.1).

In addition, semantics for Attribute Precedence, a Queue Override capability, and the capability to guarantee protocol precedence over the PDL attribute are defined.

Table 1 - Summary of Attributes defined

Attribute Name (syntax)	Description
Operation attributes:	
job-password (octetString(255))	password for secure printing
job-password-encryption (type3 keyword | name(MAX))	password encryption method for secure printing
Job Template attributes:	
feed-orientation (type3 keyword)	media edge to be fed into the print engine from the paper tray
font-name-requested (name(MAX))	font name when missing from the document data (e.g. text documents)
font-size-requested (integer (1:MAX))	font size when missing from the document data (e.g. text documents)
job-hold-until-time (dateTime)	hold the job until the supplied date and time
job-phone-number (text(127))	contact telephone number for the job
job-recipient-name (name(MAX))	name of the person that is to receive the output of the job
job-save-disposition (collection)	save the Document Data of a job, such that the job can be re-printed
pdl-init-file (1setOf collection)	controls initialization of the Printer's PDL interpreter(s)
proof-print (collection)	control a proof print of the job before printing a full run of the job
Job Description attribute:	
job-printer-make-and-model (text(127))	make and model of the output device which saved this job
Printer Description	
job-creation-attributes-supported (1setOf type2 keyword)	set of Job Creation attributes supported
job-password-supported (integer(0:127))	maximum unencrypted password length supported
job-password-encryption-supported (1setOf (type3 keyword | name(MAX)))	encryption methods supports for Secure Print
job-spooling-supported (type2 keyword)	indicates whether or not jobs are spooled before printing
max-save-info-supported (integer(1:MAX))	maximum number of "save-info" member attribute collections that a Printer can accept
media-col-database (1setOf collection)	set of media collections available in the printer's media database
printer-detailed-status-messages (1setOf text(MAX))	additional detailed and technical information about the printer
which-jobs-supported (1setOf type2 keyword)	supported values for the "which-jobs" operation attribute of the Get-Jobs operation

Many of these functions MAY be specified in a document format (PDL). In such cases, the user MAY request that the application include these instructions as part of the document data when the document is generated, rather than in the IPP protocol at print time. However, some applications are unable to support some of the functions. Also some of these functions are not supported in some PDLs. Finally, in a production environment, the document may be generated separately from being printed, in which case the end user or the production printer operator supplies the instructions at print time, long after the document had been created.

2 Terminology

This section defines the following additional terms that are used throughout this document.

2.1 Conformance Terminology

Capitalized terms, such as **MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED NOT,** and **OPTIONAL**, have special meaning relating to conformance to this document. These terms are defined in [RFC2911 section 13.1 on conformance terminology, most of which is taken from RFC 2119 [RFC2119]. Since support of the attributes defined in this specification is OPTIONAL for conformance to IPP/1.0 ([RFC2566], [RFC2565]) or IPP/1.1 ([RFC2911], [RFC2910]), the terms MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED NOT, and OPTIONAL apply ***if and only if the attribute is implemented.*** Thus a feature labeled as REQUIRED in this document is not REQUIRED if implementing the basic IPP/1.1 protocol defined by [RFC2911] and [RFC2910], but is REQUIRED if implementing the attribute in which the term occurs.

2.2 Other terminology

Actual File Name	The absolute URI path to one of the job components of a saved job (See Virtual File Name.)
Document Creation Operations	The operations that create documents: Print-Job, Print-URI, Send-Document and Sent-URI.
Job Creation operation	One of the operations that creates a Job object: Print-Job, Print-URI and Create-Job. The Restart-Job operation [RFC2911] is not considered a Job Creation operation, since the Printer re-uses the existing Job object. The Validate-Job operation is not considered a Job Creation operation because no Job object is created. Therefore, when a statement also applies to either the Restart-Job and/or the Validate-Job operation, they are mentioned explicitly.
Job Instructions	Information that affects how the job and its associated documents are to be processed. This includes the Job Template Attributes, some Operation Request Attributes, and other attributes (such as defaults) that are applied to a job.
Job Save and Reprint	The Job Save and Reprint Capability is additional OPTIONAL

Capability	functionality to allow a user to save a print job as part of job processing and print it at another time. When saving a job, a user MAY specify saving parameters such as the location (remote or local) to save the Job Instructions and/or the Document Data, and/or specify the format in which the print data and job instructions are saved.
Job-Submission Operations	The Job-Submission Operations are the IPP operations that create jobs and send document content, namely Print-Job, Print-URI, Create-Job, Send-Document and Send-URI. See [RFC2911] for further information.
Precedence	The specification of the order or ranking of a series of instructions or attributes from multiple sources referring to the same functionality. See Section 3.2 of this specification for a description of the Attribute Precedence model.
Print-stream pages	The sequence of pages according to the definition of pages in the language used to express the document data defined relative to the Input Document. See Section 2.5 of [prod-print] for more information.
Production Printer	A Printer that produces large quantities of high quality output, that often requires operator participation to make decisions as to the choice of job and its parameters.
Raster image	A binary representation of an image.
RIP	Raster Image Processor - a page description language interpreter.
Virtual File Name	The absolute URI path supplied by the client as a handle when saving a job that the Printer will associate with the saved job and all its job components. Whether or not the Virtual File Name is the same as the Actual File Name of one of the job components depends on the save file format. When referencing a saved job, a client MUST be able to use the Virtual File Name, and the Printer MUST resolve the Virtual File Name to identify the saved job components. For example, the difference between a Virtual File Name and an Actual File Name might be the addition of a file name extension to the Virtual File Name to reference an Actual File Name of a component (see the description of the "save-location" member attribute for the "job-save-disposition" Job Template attribute, described in this specification.) The general mechanism should be that the same value for “save-location”, which is the Virtual File Name, is also used for referencing the saved job. The fact that the Actual File name of a component can be reference directly is an implementation coincidence, and should not be encouraged. However, a Printer MUST support both means of referencing a saved job in the Print-URI operation.

1

3 Semantics common to more than one attribute

3.1 Guaranteed Protocol Precedence Over the PDL and Queue Overrides

The IPP "pdl-override-supported" attribute has a new 'guaranteed' value which a Printer implementation uses to indicate that it can guarantee that Job Template attribute supplied in the protocol will override corresponding instructions in the PDL document data. The existing values for this attribute are 'not-attempted' and 'attempted' which are weaker than 'guaranteed'. See the description of the "pdl-override-supported" attribute in [RFC2911].

IPP does not preclude multiple Printer objects representing a single output device, i.e., so-called "device fan-in". [RFC2911] describes device fan-in in detail and also introduces the concept of "Printer fan-in" in which multiple Printer objects represent a single Printer object, each with the ability to accept IPP requests. “Printer fan-in” is described more fully in [admin-ops]. A system with multiple queues can be represented as separate Printer objects for each queue, using either device fan-in or Printer fan-in depending on implementation. The administrator can configure each such Printer object with different policies, including supported and default attributes. Each Printer can also have a different access control list, as well.

Sometimes, the system administrator needs to set up a Printer object that will override one or more attributes with a single fixed value for each attribute. These are called "queue overrides". The administrator can configure a Printer that supports the 'guaranteed' value of "pdl-override-supported" to achieve a queue override for the "xxx" attribute by configuring the "xxx-supported" Printer attribute with only a single value. That value will not only override the PDL, but it will also force the job to have that same value when queried or processed. See the description of the "pdl-override-supported" attribute in Section 8.1 of this volume.

3.2 Attribute Precedence

This section defines the precedence rules for Queue Override attributes, Page Override attributes, Document Override attributes, attributes supplied in the protocol, PDL instructions, and Printer object defaults. Since each attribute can be specified for all of these levels, it is important to define which level takes precedence when the same attribute occurs at different levels with differing values.

The attribute precedence model has the following features:

1) In principle, each defined attribute can occur at each of the precedence levels.

2) Attributes can be introduced and modified at different points in the job workflow, e.g., in the application, print driver, submission client, protocol, and Printer:

   - Attributes are supplied by different agents in the system: application, user, print driver, administrator, operator, Printer object, and the output device hardware.

- Attributes are supplied at different points in time: PDL generation, job submission, job acceptance, job pending, job processing.

3) Attributes can be introduced and modified at different points in the job workflow, e.g., in the application, print driver, submission client, protocol, and Printer:

The following levels of precedence are defined in order of decreasing priority:

1. **queue override** - The value that the Printer enforces for any job submitted to it (see section 3.1). Set by the administrator when configuring the Printer by setting an "xxx-supported" Printer attribute to contain a single value, i.e., the override value. Example: the administrator sets the Printer's "sides" = 'two-sided-long' so that all jobs submitted to that (logical) Printer are printed two-sided flipped along the long edge.

   Note: The **queue override** level is only available for Printers that support the 'guaranteed' value for its "pdl-override-supported" attribute (see section 3.1 of this specification). All other levels are available for all Printers independent of the "pdl-override-supported" value.

2. **page override programming** - A special "page-overrides" collection Job Template attribute that contains the attributes **that** are to have the page override status (see [override]). Set by the client when submitting a job either in the protocol, or set by the user or operator after the job has been accepted either using the Set-Job-Attributes operation, or by the operator using means local to the Printer. Example: "page-overrides" = {"page-number" = '1'; "media" = 'letterhead'}

3. **document override programming** - A special "document-overrides" collection Job Template attribute that contains the attributes that are to have the document override status (see [override]). Set by the client when submitting a job either in the protocol, or set by the user or operator after the job has been accepted either using the Set-Job-Attributes operation, or by the operator using means local to the Printer. Example: "document-overrides" = {"document-number" = '5; "media" = 'transparency'}

4. **protocol job object attribute** - The Job Template attribute submitted in the protocol (IPP, LPR, etc). Set by the client when submitting a job in the protocol, or set by the user or operator after the job has been accepted either using the Set-Job-Attributes operation, or by the operator using means local to the Printer. Examples: "media" = 'na-letter' and "copies" = '2'.

5. **PDL document data** - an instruction in the PDL document data. Set by the application or print driver that created the PDL. Example: PostScript setPageDevice specifying na-letter media.

6. **PDL Init File data** - a file that the Printer uses to initialize the PDL Interpreter before it starts interpreting each document in a Job. See description of “pdl-init-file” Job Template attribute in Section 5.8 of this specification. It is immaterial how this file is

specified, whether it be by a queue override, job object attribute or printer default – the precedence of the contents of this file is at this level.

7. **printer default** - a Printer "xxx-default" attribute that is applied by the Printer provided that none of the higher levels have supplied a value. The Printer implementation MUST supply values for all of the Printer's "xxx-default" attributes that reflect the out-of-the-box action by the output device. For example, if the output device will use media from the large-capacity input tray if not directed otherwise, the implementation MUST supply the Printer's "media-default" attribute with the 'large-capacity' keyword value without requiring the administrator to configure that value. The implementation MUST allow the administrator to configure the Printer's "xxx-default" attributes to other values, if other value are supported by the implementation. Example: The administrator changes the Printer's "media-default" from 'large-capacity' to 'na-letter'.

Note: Because of the requirement that the implementation supply pre-configured values for the Printer's "xxx-default" attributes, there is no need for a lower "hardware default" precedence level.

3.3 Job Save and Reprint Capability

The Job Save and Reprint Capability is additional OPTIONAL functionality to allow a user to save a print job as part of job processing and print it at another time. When saving a job, a user MAY specify saving the Document Data, and specify the format in which the print data is saved. These semantics necessitate the definition of additional values for the "job-state-reasons" Job Description attribute, the "job-printer-make-and-model" Job Description attribute, and the use of the Print-URI operation to print these saved jobs.

This section, in conjunction with the description of the "job-save-disposition" Job Template attribute (see section 5.7) fully describe the Job Save and Reprint Capability.

3.3.1 Reprinting Saved Jobs

This section defines the method by which to reprint the saved job using the Print-URI operation.

3.3.1.1 Reprinting using the Print-URI operation

Job Reprint is accomplished by using the Print-URI operation. First a client requests the Printer to create a saved job by supplying the “job-save-disposition” Job Template attribute in a Job Creation operation (see description in section 5.7) with a URI value for the “save-location” member attribute and a name value for the “save-name” member attribute. To reprint the saved job, a client concatenates the original “save-location” and “save-name” values into a URI reference value to the saved job using the "document-uri" (uri) operation attribute (in Group 1) of the Print-URI operation (see [RFC2911]). The "document-uri" attribute value MUST be a URI reference to an object in saved document format (see description of "save-document-format" member attribute in the "job-

save-disposition" description in Section 5.7). The Printer MAY perform implementation dependent actions in order to resolve the URI, since the URI MAY be a Virtual Name (see section 2.2 and the "save-location" member attribute of "job-save-disposition" described in Section 5.7.1.2.3.1), or an Actual File Name (see section 2.2), that the Printer uses to determine the locations of the saved job's component files and directories.

If the "document-uri" operation attribute value in the Print-URI request is a valid reference to a PDL document format, and the Printer object supports that format as specified by the "document-format-supported" Printer Description attribute, the Printer processes the job as defined by the Print-URI operation defined in [RFC2911].

4 Operation attributes

4.1 job-password (octetString(255))

This Operation attribute may OPTIONALLY be included in the request of the following Job Creation operations: Print-Job, Print-URI, and Create-Job.

The OPTIONAL "job-password" operation attribute allows a user to perform Secure Print. The user enters a password in the job submitting application which is encrypted by the client using one of the methods specified by the "job-password-encryption-supported" attribute. The encrypted password is sent to the Printer as the value of the "job-password" attribute. If a "job-password" value is provided, the printer MUST hold the job in the 'pending-held' state, and the 'job-password-wait' value is added to the "job-state-reason" attribute.

The user enters the same password at the device to release the job for printing. The Printer uses the same encryption method specified in the "job-password-encryption" attribute on this password. The printer MUST resume the print job when the locally-supplied encrypted password matches the value of the "job-password" attribute. The method in which the password is entered and validated at the Printer is implementation dependent.

The "job-password" attribute value MUST NOT be returned in a Get-Job-Attributes response.

If a "job-password" attribute value other than a zero-length string is provided in a job creation operation, the job is placed into the 'pending-held' state, and the 'job-password-wait' value is added to the "job-state-reason" attribute.

Other jobs may be printed before the release of the Secure Print Job from the 'pending-held' state. If the Secure Print Job is released by the user while another job is printing, the Secure Job MUST NOT resume printing until the current job is done printing. The Secure Job SHOULD be the next job printed after the current job, unless there is another job in the Printer which has a higher priority than the Secure Job as determined by the "job-priority" attribute.

4.2 job-password-encryption (type3 keyword | name(MAX))

The "job-password-encryption" attribute specifies the type of encryption used for the value of the "job-password" attribute in the request of the following Job Creation operations: Print-Job, Print-URI, and Create-Job. The client MUST supply this operation attribute whenever the "job-password" attribute is supplied.

The valid keyword values are the same as the valid values for the "job-password-encryption-supported" attribute (See description below in section 7.3).

5 Job Template Attributes

Job Template attributes describe job processing behavior and conform to a set of rules. See [RFC2911] for the complete text of the rules that apply to each Job Template attribute called "xxx".

5.1 feed-orientation (type3 keyword)

The OPTIONAL "feed-orientation" Job Template attribute specifies the media edge which is fed into the print engine from the paper tray. Long-edge-first is the preferred method of feeding media to attain the fastest printing speed. Certain media will have adverse effects when fed from a certain direction; for instance, some labels will peel off in the paper path when fed long-edge-first, but not when fed short-edge-first. The feed orientation also may allow the job to be finished in more acceptable ways; for instance, a stationary finishing stapler may provide a preferred stapling location when the media is fed short-edge-first.

When this attribute is specified, the printer selects media that has already been loaded in the requested orientation. If the media is not currently loaded in the requested orientation, the job may enter the 'processing-stopped' state with a "job-state-reason" of 'resources-are-not-ready'.

Standard keyword values are:

Keyword	Description
'long-edge-first'	The specified media is fed using the long edge first.
'short-edge-first'	The specified media is fed using the short edge first.

This attribute allows a PDL interpreter to determine which way to send the image data to the frame buffer for imposition on the media. Although Postscript supports this attribute as a PDL construct, support in other PDLs such as PCL and TIFF are dependent on the implementation. Therefore, this attribute is defined to be specified as a Job Template attribute such that printers may use this feature with PDLs other than Postscript. Some printer implementations may be able to internally determine the best feed orientation to use so that if this attribute was sent over the wire to the printer, the attribute would be ignored.

This attribute has no special interaction with the "orientation-requested" attribute. It is possible to print a portrait document on either SEF (short edge feed) or LEF (long edge feed) paper, and likewise for landscape documents. However, this may affect staple placement and other finishing, depending on the device's capabilities.

This attribute applies as to how the media is loaded into the tray rather than the media itself, and is therefore a job-level attribute rather than a media collection member attribute. This attribute may be used with "page-overrides" and "document-overrides" just as any other Job Template attribute (see [override]).

5.1.1 feed-orientation-default (type3 keyword)

The "feed-orientation-default" (type3 keyword) Printer attribute specifies the default value of "feed-orientation" when not supplied in a request. This default depends upon the media-size being requested and is printer implementation dependent upon how the default value is calculated.

5.1.2 feed-orientation-supported (1setOf (type3 keyword))

The "feed-orientation-supported" (1setOf (type3 keyword)) Printer attribute specifies which values of "feed-orientation" that the Printer supports.

5.2 font-name-requested (name(MAX))

5.2.1 font-name-requested-default (name(MAX))

5.2.2 font-name-requested-supported (1setOf name(MAX))

This extension enables an IPP client to specify what default font name the printer MUST use to print a job if the document data is in a format that does not have inherent font information (e.g., ‘text/plain’). For document formats which have inherent font information (such as PostScript), this attribute will be ignored and will NOT override that information.

For some document formats (such as 'application/postscript'), the desired default font name of the print-stream pages is specified within the document data. This information is generated by a device driver prior to the submission of the print job. Other document formats (such as 'text/plain') do not include the notion of desired font name within the document data. In the latter case it is possible for the Printer object to bind the desired font name to the document data after it has been submitted. It is expected that a Printer object would only support "font-name-requested" for some document formats (e.g., 'text/plain' or 'text/html') but not others (e.g., 'application/postscript'). This PDL-dependent behavior is no different than any other Job Template attribute since a Printer object may support or not support any Job Template attribute based on the document format supplied by the client. However, a special mention is made here since it is very likely that a Printer object will support "font-name-requested" for only a subset of the supported document formats.

This attribute can be specified as a Document Override that affects the Input-Document. The use of this attribute on a Page override basis is not supported since changing the font characteristics can affect the pagination.

NOTE: The use of the “xxx-requested” pattern for attribute names indicates that the value of the attribute is to be used ONLY in the case when a value for the attribute is not contained within the source document. This value will override the printer’s default value but will not override the source document’s value. See the description of the “orientation-requested” Job Template attribute later in [RFC2911].

5.3 font-size-requested (integer (1:MAX))

5.3.1 font-size-requested-default (integer (1:MAX))

5.3.2 font-size-requested-supported (1setOf rangeOfInteger (1:MAX))

This extension enables an IPP client to specify what default font size the printer MUST use to print a job if the document data is in a format that does not have inherent font information (e.g., ‘text/plain’). For document formats which have inherent font information (such as PostScript), this attribute will be ignored and will NOT override that information.

For some document formats (such as 'application/postscript'), the desired default font size of the print-stream pages is specified within the document data. This information is generated by a device driver prior to the submission of the print job. Other document formats (such as 'text/plain') do not include the notion of desired font size within the document data. In the latter case it is possible for the Printer object to bind the desired font size to the document data after it has been submitted. It is expected that a Printer object would only support "font-size-requested" for some document formats (e.g., 'text/plain' or 'text/html') but not others (e.g., 'application/postscript'). This PDL-dependent behavior is no different than any other Job Template attribute since a Printer object may support or not support any Job Template attribute based on the document format supplied by the client. However, a special mention is made here since it is very likely that a Printer object will support "font-size-requested" for only a subset of the supported document formats.

The “font-size-requested” units are points, equivalent to $1/72^{nd}$ of an inch.

This attribute can be specified as a Document Override that affects the Input-Document. The use of this attribute on a Page override basis is not supported since changing the font characteristics can affect the pagination.

NOTE: The use of the “xxx-requested” pattern for attribute names indicates that the value of the attribute is to be used ONLY in the case when a value for the attribute is not contained within the source document. This value will override the printer’s default value but will not override the source document’s value. See the description of the “orientation-requested” Job Template attribute in [RFC2911].

5.4 job-hold-until-time (dateTime)

This attribute permits the client to specify the date and time after which the Job MUST become a candidate for processing.

Note: The client application has the choice to present the "job-hold-until-time" to the end user as either a delta time (the amount of time until the job is no longer held), or a fixed time when the job will no longer be held. The fixed time is represented using the appropriate time zone(s) (usually the client's time zone, or the Printer's time zone, or both if the client is capable). See the description of "printer-current-time" in [RFC2911].

If the Printer supports the "job-hold-until-time" attribute, the "job-hold-until" attribute MUST also be supported. However, if the "job-hold-until" attribute is supported, the "job-hold-until-time" attribute NEED NOT be supported. If the Printer supports the "job-hold-until-time" attribute, the "printer-current-time" (dateTime) Printer Description attribute MUST also be supported.

The client MUST NOT supply both the "job-hold-until" and "job-hold-until-time" Job Template attributes in a Job Creation request. If the client supplies such a mal-formed request by supplying both, the Printer MUST (depending on implementation) either:

1. reject the request and return the 'client-error-bad-request' status code (see [RFC2911]) or
2. use either the "job-hold-until" or the "job-hold-until-time" attribute, independent of the value of the "ipp-attribute-fidelity" attribute supplied by the client.

An Activity Diagram for "job-hold-until-time" is shown in Figure 1 below. A Hold Job request is sent by the Client with a "job-hold-until-time" value specified as a dateTime. The Printer calculates the number of seconds between the "job-hold-until-time" value and the "printer-current-time" value. If this number of seconds is NOT in the range specified by the Printer's "job-hold-until-time-supported" attribute value, then the Printer either:

1. rejects the request with the 'client-error-attributes-or-values-not-supported' status code (see [RFC2911]) if the "ipp-attribute-fidelity" is 'true' or
2. accepts the request with the 'successful-ok-ignored-or-substituted-attributes', if the client-supplied "ipp-attribute-fidelity" is 'false'. Also, the Printer MUST return the "job-hold-until-time" attribute and the client-supplied value in the Unsupported Attributes group of the operation response. If the "job-hold-until-time" value is earlier than the "printer-current-time" value plus the minimum value of "job-hold-until-time-supported", the Printer MUST set the value of the Job's "job-hold-until-time" to the current dateTime plus the minimum time. Also, if the "job-hold-until-time" value is later than the "printer-current-time" value plus the maximum value of "job-hold-until-time-supported", the Printer MUST set the value of the Job's "job-hold-until-time" to the current dateTime plus the maximum time.

If the job is accepted, the Printer then determines the state in which to place this job. If the dateTime value is equal to or sooner than the Printer's "printer-current-time" dateTime value, then no hold is placed on the job and the job moves into the pending state (assuming there are no other reasons to hold the job).

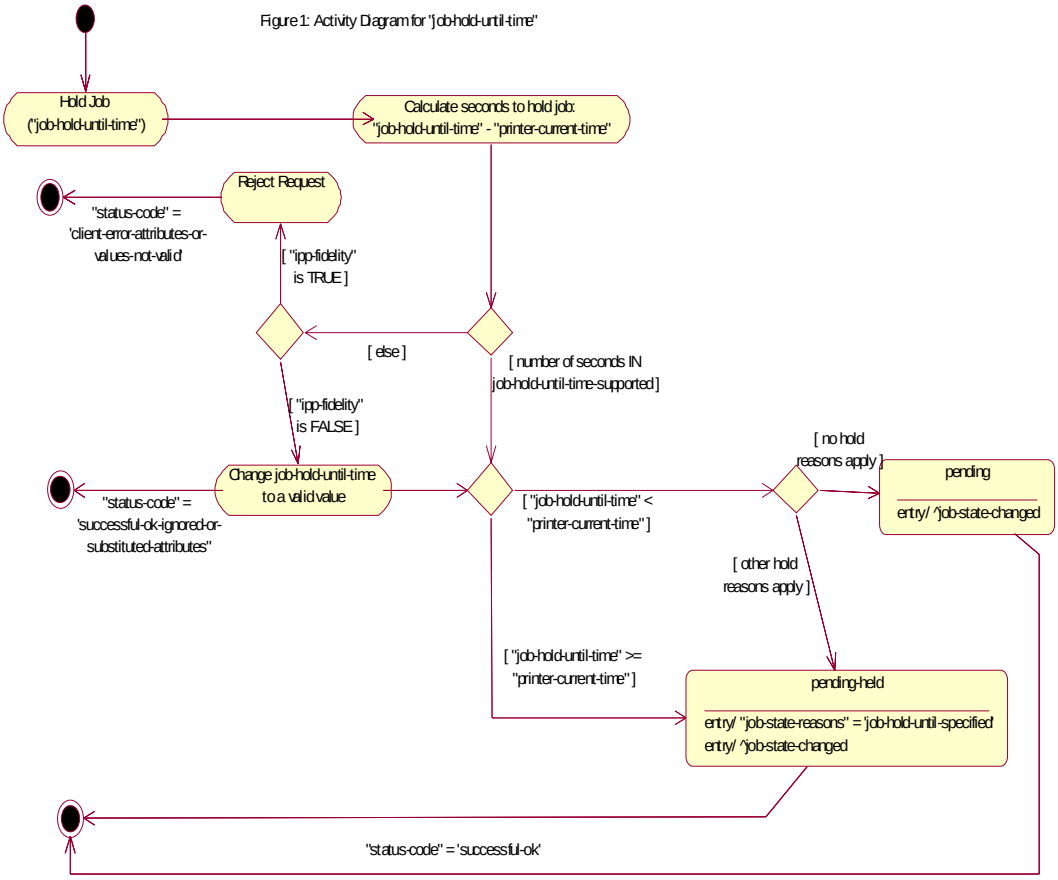
If the value of this attribute specifies a time period that is in the future, the Printer 1) SHOULD add the 'job-hold-until-specified' value to the job's "job-state-reasons" attribute, 2) MUST move the job to the 'pending-held' state, and 3) MUST NOT schedule the job for printing until the specified time arrives. When the specified time arrives, the Printer MUST remove the 'job-hold-until-specified' value from the job's "job-state-reason" attribute and, and then if there are no other job state reasons that keep the job in the 'pending-held' state, the Printer MUST consider the job as a candidate for processing by moving the job to the 'pending' state.

5.4.1 Why there is no job-hold-until-time-default

There is no attribute "job-hold-until-time-default" because it would either be a fixed time in the future or would have to be updated periodically. Neither seem useful enough to be worth the complexity.

5.4.2 job-hold-until-time-supported (rangeOfInteger(0:MAX))

A range of integer values of the "job-hold-until-supported" attribute specifies the span of relative amount of time in seconds into the future that the printer supports keeping the submitted job in the 'pending-held' state. The Printer MUST use the time range to validate a job that is submitted with a "job-hold-until-time" attribute.



1

2 5.5 job-phone-number (text(127))

3 5.5.1 job-phone-number-default (text(127))

4 5.5.2 job-phone-number-supported (boolean)

5 The “job-phone-number” attribute contains the contact telephone number for the job.

5.6 job-recipient-name (name(MAX))

5.6.1 job-recipient-name-default (name(MAX))

5.6.2 job-recipient-name-supported (boolean)

This attribute contains the name of the person that is to receive the output of the job. The value of the "job-recipient-name" attribute is commonly printed on job sheets printed with the job. An example of another use of the "job-recipient-name" attribute is if the printer accesses a database to get job delivery instructions for the recipient of a job. A zero-length value indicates that there is no job recipient name.

If the client omits this attribute in a create request, the printer MAY use the “job-recipient-name-default” attribute value, unless it has not been configured by the administrator, or MAY use the “authenticated user” name (see [MOD1.1] section 8.3), depending on implementation.

5.7 job-save-disposition (collection)

The “job-save-disposition” Job Template attribute is used to archive/save the Document Data of a job, such that the job can be re-printed on demand at some undefined time in the future. See also section 3.3 of this specification for a detailed description of the Job Save and Reprint Capability.

If a Printer object is able to save a Job in a particular document format, the Printer SHOULD be able to ingest that document format for reprint using the Print-URI operation.

Interaction with "job-hold-until": The "job-hold-until" attribute causes the job to be moved to the ‘pending-held’ state, which prevents processing of the job. The "job-save-disposition" attribute on the other hand does not become relevant until the job is moved to the ‘processing’ state. These two attributes can be used in tandem with no adverse interactions.

5.7.1 Member Attributes for the "job-save-disposition" Job Template Attribute

The member attributes of the "job-save-disposition" attribute are:

Table 2 - "job-save-disposition" member attributes

Member Attribute Name	Attribute Syntax	Request	Printer Support
save-disposition	type3 keyword	MUST	MUST
save-info	1setOf collection	MAY	MUST

5.7.1.1 save-disposition (type3 keyword)

The "save-disposition" member attribute specifies whether or not the job MUST be printed and/or saved. When a job moves into the ‘processing’ state, the value of the member attribute "save-disposition" is checked to determine what is to happen during the ‘processing’ state.

The "save-disposition" member attribute specifies processing steps that either MUST occur or that are prohibited. However, other processing steps MAY occur during the ‘processing’ state, provided they are not prohibited by the value of the "save-disposition" member attribute.

Standard keyword values are:

Keyword	Description	Printer Support
'none'	The Printer MUST print the job. The Printer MUST NOT save any portion of the job except for processing purposes. The Printer MUST NOT apply the "job-save-disposition-default" attribute. If the print was successful, the Printer MUST add the ‘job-completed-successfully’ value to the job’s “job-state-reasons” attribute. If the printing had (1) warnings or (2) errors (possibly with warnings), the Printer MUST add the ‘job-completed-with-warnings’ or ‘job-completed-with-errors’ value, respectively.	MUST
'save-only'	The following constraints apply to the ‘processing’ of the job: • The job MUST NOT be printed. • The Document Data MUST be saved to the location specified by the "save-location" member attribute. The Printer’s job scheduling algorithm MAY depend on whether or not the job is ‘save-only’. However, the Printer MUST save the job while the job is in the ‘processing’ state (the job’s “job-state” attribute is set to ‘processing’). Thus the Printer MUST move the job (typically from the ‘pending’ state) to ‘processing’ state and add the ‘job-saving’ value (see description of new "job-state-reasons" later in section 8.3) to the job’s “job-state-reasons” attribute. When the Printer completes saving the job, the Printer MUST move the job to the ‘completed’ state by setting the job’s “job-state” attribute to the ‘completed’ value and removing the ‘job-saving’ value (see description of new "job-state-reasons" later in section 8.3) from the job’s “job-state-reasons” attribute. If the save was not successful, the Printer MUST add the ‘job-save-error’ value to the job’s “job-state-reasons” attribute (see description of new "job-state-reasons" later in section 8.3). If the Printer encounters an error during saving, such that a reprint of that job using the Print-URI operation (see section 3.3.1.1) will not produce complete results, the Printer MUST ensure that the saved job is not accessible for such a reprint in some IMPLEMENTATION DEFINED way.	MUST
'print-save'	The following constraints apply to the ‘processing’ of the job: • The job MUST be printed. • The Document Data MUST be saved to the location specified by the "save-location" member attribute.	MUST

The order of the processing steps required for the ‘print-save’ disposition is implementation dependent. However, the Printer MUST save the job during the same period that it prints the job, namely, while the job is in the ‘processing’ state (the job’s “job-state” attribute is set to ‘processing’). Thus the Printer MUST move the job (typically from the ‘pending’ state) to ‘processing’ state and add the ‘job-printing’ and ‘job-saving’ value (see description of new "job-state-reasons" later in section 8.3) to the job’s “job-state-reasons” attribute at the appropriate times.

When the Printer completes saving the job, then the Printer MUST move the job to the ‘completed’ state by setting the job’s “job-state” attribute to the ‘completed’ value and removing the ‘job-printing’ and ‘job-saving’ (see description of new "job-state-reasons" later in section 8.3) values from the job’s “job-state-reasons” attribute.

If both the save and the print were successful, the Printer MUST add the ‘job-completed-successfully’ value to the job’s “job-state-reasons” attribute.

If the printing had (1) warnings or (2) errors (possibly with warnings), the Printer MUST add the ‘job-completed-with-warnings’ or ‘job-completed-with-errors’ value, respectively.

As with the ‘save-only’ value, if the save was not successful, the Printer MUST add the ‘job-save-error’ value to the job’s “job-state-reasons” attribute (see description of new "job-state-reasons" later in section 8.3).

5.7.1.1.1 save-disposition-supported (1setOf type3 keyword)

The “save-disposition-supported (1setOf type3 keyword)” Printer attribute defines the save dispositions supported by the Printer for the “save-disposition” member attribute (see standard keyword values in above table).

5.7.1.2 save-info (1setOf collection)

The "save-info" member attribute is a collection that contains the attributes that tell the printer how to save the job. This includes the format in which the Document Data MUST be saved, and the location to which these are saved. Multiple save locations or document formats MAY be saved by specifying multiple collections within this attribute.

If the client supplies the “job-save-disposition” Job Template attribute, but omits the “save-info” member attribute, the Printer supplies a single collection value for the “save-info” member attribute from the values of its “save-location-default” attribute, the job’s “job-name”, and its “save-document-format-default” attribute (see descriptions immediately below).

A client MUST supply in a request a number of collections not to exceed the maximum number supported specified in the "max-save-info-supported" Printer Description attribute (see "max-save-info-supported" description in section 7.5). As with any Job Template attribute, if the client does supply more values than the Printer supports and the values of the “ipp-attribute-fidelity” is ‘false’ (or omitted), the Printer MUST accept the job, return the ‘successful-ok-ignored-or-substituted-attributes’ status code, return the ignored values in the Ignored Attributes group, use the first n values, and ignore the remaining values. If the client does supply more values than the Printer

supports and the values of the “ipp-attribute-fidelity” is ‘true’, the Printer MUST reject the request and return the ‘client-error-request-entity-too-large’ status code.

5.7.1.2.1 There is no save-info-default Printer attribute

There is no “save-info-default” Printer attribute. If the client supplies the “job-save-disposition” Job Template attribute, but omits the “save-info” member attribute, the Printer creates a single collection value for the “save-info” member attribute using from the Printer’s “save-location-default” and “save-document-format-default” Printer attributes and the job’s “job-name” attribute (see "save-info" description above).

5.7.1.2.2 save-info-supported (1setOf type2 keyword)

The "save-info-supported (1setOf type2 keyword)" attribute identifies the names of the member attributes supported in the "save-info" collection attribute, i.e., the names of the member attributes in Table 3 that the Printer supports.

5.7.1.2.3 Member Attributes for the "save-info" member attribute

The "save-info" collection member attributes are:

Table 3 - "save-info" member attributes

Member Attribute Name	Attribute Syntax	Request	Printer Support
save-location	uri	MAY	MUST
save-name	name (MAX)	MAY	MAY
save-document-format	mimeMediaType	MAY	MUST

5.7.1.2.3.1 save-location (uri)

The "save-location" member attribute specifies the path to the directory as a URI where the Printer MUST save the Document Data and other information.

The "save-location" and (“save-location-default” and “save-location-supported”) attribute value MUST be an Absolute URI [RFC2396]. Absolute URIs are specified with a URI scheme, an optional authority component, and an absolute path (e.g., ‘ftp://printhost.printco.com/var/spool/jobdir/’ or ‘file:///job-repository/jobdir/’).

Note: As any other member attribute of a Job Template attribute, the Printer validates the “save-location” member attribute as follows: If the client supplies the “save-location” member attribute, the value MUST match one of the values of the Printer’s “save-location-supported” (1setOf uri) attribute. If the value does not match, the Printer’s action depends on the value of the job’s “ipp-attribute-fidelity” (boolean)

attribute: If “ipp-attribute-fidelity” is ‘false’ or omitted, the Printer MUST accept the job, return the ‘successful-ok-ignored-or-substituted-attributes’ status code, return the ignored attributes in the Unsupported Attributes Group, and perform one of the following actions DEPENDING ON IMPLEMENTATION: (1) ignore the attribute and not save the job, (2) put the job in the ‘held’ state and let the operator fix the problem, either by changing the job’s “save-location” attribute value or adding the value to the Printer’s “save-location-supported” attribute, or (3) substitute the job’s “save-location” value with one of the values of the Printer’s “save-location-supported” values. If “ipp-attribute-fidelity” is ‘true’, the Printer MUST reject the request, return the ‘client-error-attributes-or-values-not-supported’ status code, along with the “job-save-disposition” attribute in the Unsupported Attributes Group.

If the client wants to put the saved job in a sub-directory (whether it exists or not) of one of the directories specified by the “save-location-supported” Printer attribute, the client supplies that sub-directory name as part of the “save-name” attribute (see "save-name" description below), not as part of the “save-location” member attribute.

Note: As with any other ‘name’, ‘integer’, ‘uri’, or ‘collection’ Job Template or Job Template member attribute, the administrator can suspend validation by configuring the Printer’s “user-defined-values-supported” attribute with the ‘save-location’ keyword (see “user-defined-values-supported” description in [prod-print]), so that the user can specify an arbitrary path and the Printer will accept the job. If the Printer cannot process the job using the specified value for "save-location", then the Printer MUST hold the job so that an operator can attempt to create the necessary sub-directories to populate the path, if they don’t already exist, or change the job’s “save-location” attribute value. See Table 4 below entitled "Printer actions for various Job and Printer attribute values", at the end of the "job-save-disposition" description.

If the client supplies the “save-info” member attribute, but omits the “save-location” member attribute, the Printer supplies the “save-location” member attribute value from its “save-location-default” attribute (see “save-location-default” description below).

Defined Job Save URI Schemes for use in the “save-location” member attribute include:

URI Scheme	Description	Printer Support
ftp	Use IETF FTP protocol [RFC959], [RFC2228], [RFC2640]	SHOULD
file	Use the Printer’s configured file system. Since the ‘file:’ URI scheme cannot be separated from the Printer object’s native file system, the semantic of the ‘file:’ URI scheme is IMPLEMENTATION DEPENDENT. Also the ‘file:’ URI scheme on some Printer objects MAY be limited to the local file system, while on other Printer objects it MAY be configured to use a network file system. The Printer implementation MAY allow the system administrator to configure where the ‘file:’ scheme is based in the file system. However, the base for the ‘file:’ scheme MUST be the same as for all other uses of the ‘file’: scheme, such as the value of the “document-uri” operation attribute in a Printer-URI operation (for print by reference of a saved job - see section 3.3.1.1).	SHOULD

It is RECOMMENDED that for each URI scheme supported for saving jobs, that all of these values have the first token in the file path be ‘job-repository’. Then an administrator that has to manage the saved jobs for several Printers will have a consistent naming schemes for locating all the saved jobs by whatever means used to access the saved jobs. For example, for the ‘file:’ scheme, all of the values SHOULD start out with: file:///job-repository/ and for the ‘ftp:’ scheme, all of the values SHOULD start out with: ftp://xxx/job-repository/. (Note: in URIs, the final "/" is redundant, and the Printer MUST behave the same whether or not the final “/” is present).

5.7.1.2.3.1.1 save-location-default (uri)

The “save-location-default (uri)” Printer attribute indicates the value that the Printer supplies, if the client omits the “save-location” member attribute from the “save-info” member attribute.

5.7.1.2.3.1.2 save-location-supported (1setOf uri)

The “save-location-supported (1setOf uri)” Printer attribute defines a list of URI values supported by the Printer object for the "save-location" member attribute (see above description). The client MUST supply a value of the "save-location" member attribute that completely matches one of these values. Note: The administrator can configure the “save-location-supported” to have separate directories for individuals, groups, projects, or may just have one directory for all saved jobs. Whether or not access control is applied to these directories is IMPLEMENTATION DEPENDENT.

5.7.1.2.3.2 save-name (name(MAX))

The “save-name” member attribute specifies the name of the saved job in the directory specified by the “save-location” member attribute. The client can supply a “save-name” attribute value that is either a simple file name or a relative path where each component of the path is separated by a “/” character. The "save-name" member attribute value concatenated with the “save-location” attribute value (supplying a “/” between them, if necessary) specifies a Virtual File Name (see definition in

Section 2.2) that will be associated with the saved job. For example, if “save-location” is ‘a/b’ or ‘a/b/’ and ‘save-name’ is ‘c/d’, the resulting Virtual File name is: ‘a/b/c/d’.

The actual job *components* saved by the Printer object are a function of the “save-document-format” attribute value. Some values of the “save-document-format” attribute MAY cause the Printer to append a file type extension to the Virtual File Name supplied by the client in the “save-name” attribute.

If the “save-name” attribute value specifies a handle that would require a sub-directory to be created, the Printer MUST attempt to create that sub-directory. Access control MAY prevent the Printer from creating such a sub-directory, DEPENDING ON IMPLEMENTATION, including configuration.

If the "save-name" attribute value in combination with the "save-location" attribute value specifies a handle that already exists, the Printer MUST accept the job, return the ‘successful-ok’ status code, hold the job, and prompt an administrator in an IMPLEMENTATION-DEFINED manner to change the job’s "save-location" or “save-name” attribute values before processing the job. The Printer MUST hold the job independent of the value of the “ipp-attribute-fidelity” (since with operator assistance, the Printer is able to save the job) and whether or not the ‘save-name’ keyword has been configured as a value of the Printer’s “user-defined-values-supported” attribute. See Table 4 below entitled "Printer actions for various Job and Printer attribute values", at the end of the "job-save-disposition" description. Note: A “job-save-disposition” member attribute to control the Printer’s action (reject job, over write file, hold job and prompt administrator, or auto-generate a non-conflicting name) to resolve any URI conflict may be a desirable future extension.

If the “save-name” member attribute is not specified by the client in the “job-save-disposition” Job Template attribute, the Printer object MUST generate a “save-name” attribute value for the job using the job’s “job-name” attribute value. Note: Printers MUST generate a “job-name” value, when the client does not supply a “job-name” operation attribute (see Print-Job operation), so there is always a “job-name” value. After generating the “save-name” value, the Printer MUST handle any conflict between the generated value and the existing values as above.

5.7.1.2.3.2.1 There is no “save-name-default” Printer attribute

There is no “save-name-default”. If the client omits the “save-name” member attribute when supplying the “save-info” member attribute, the Printer uses the job’s “job-name” attribute as the value of the “save-name” attribute (described above).

5.7.1.2.3.2.2 save-name-supported (boolean)

The “save-name-supported (boolean)” attribute indicates support for the OPTIONAL "save-name" member attribute of the "save-info" member attribute. Note: The means to query the Printer to see what saved jobs it has is a potential future extension.

5.7.1.2.3.3 save-document-format (mimeMediaType)

The "save-document-format" member attribute indicates the document format in which the job content (Document Data) MUST be saved. If the "job-save-disposition" attribute is supported, the printer object MUST support the "save-document-format" member attribute.

If the client supplies the “save-info” member attribute, but omits the “save-document-format” member attribute, the Printer MUST use the value of its “save-document-format-default” attribute (see above "save-info" description).

Although there can be a many-to-one relationship of documents to a job, all documents in a job specified with a disposition of 'save' or 'print-save' MUST be saved in the same format as specified by the "save-document-format" attribute value. A job MAY be saved to multiple formats by specifying multiple collection values for the "save-info" member attribute.

The “job-save-disposition” member attribute (and the “save-document-format” member attribute) is NOT defined to be used with the “document-overrides” and “page-overrides” attributes (see [override]). Thus all documents MUST be saved in the same document format in each saved job.

The ‘mimeMediaType’ values defined for the “save-document-format” attribute are listed in the Table below. All values MUST include saving PDL data. A Printer MUST support at least one value. However, at this time, this specification does NOT REQUIRE a Printer to support any particular formats for saving jobs.

If the saved job is submitted to a different make-and-model printer that supports the same document-format (i.e., the same value of the “document-format” operation attribute in the Document Creation request), then the Printer MUST be able to print the job, but the output cannot be guaranteed to appear exactly the same as the original output.

MIME Media Type	Description
any PDL document format	If any PDL document-format value is supplied from the Printer’s “document-format-supported” attribute, such as ‘application/postscript’, then the Printer MUST save the PDL data in that document format. The ‘application/octet-stream’ value MUST NOT be supplied for this attribute since it doesn’t make any sense for saving jobs.

The referenced specifications that define the saved format MUST meet the conformance requirements listed in section 9.3.1.3.

A "hard-proof" is when the user want to print one copy of a job to review before printing the remaining copies. This can be accomplished using the "job-save-disposition" attribute by setting the "save-disposition" attribute to ‘print-save’ with the "copies" attribute set to 1. The job is printed and saved. If the proof is satisfactory, the user submits a Print-URI request (see section 3.3.1.1) to print the saved job with no "job-save-disposition" specified and the "copies" attribute set to the desired value.

5.7.1.2.3.3.1 save-document-format-default (mimeMediaType)

If the client supplies the “save-info” member attribute, but omits the “save-document-format” member attribute (see above description), the Printer MUST use the value of its “save-document-format-default” (mimeMediaType) attribute.

5.7.1.2.3.3.2 save-document-format-supported (1setOf mimeMediaType)

The "save-document-format-supported" (1setOf mimeMediaType) attribute specifies the document formats which a Printer supports for saving jobs.

If a Printer supports this Job Save & Reprint Capability, the Printer SHOULD be able to accept all of the formats for Reprint using the Print-URI operation (see Section 3.3.1.1) as the formats in which it is capable of saving jobs, i.e., the mimeMediaType values of the Printer’s “document-format-supported” Printer Description attribute SHOULD be a superset of the values of the Printer’s “save-document-format-supported” attribute. For example, if a Printer supports the ‘application/X-single-page-tiff’ format for saving jobs, it SHOULD also accept and support ‘application/X-single-page-tiff’ format in a Printer-URI operation as indicated by the ‘application/X-single-page-tiff’ also being a value of the Printer’s “document-format-supported” Printer Description attribute (along with PDL mimeMediaType values, such as ‘application/postscript’).

5.7.1.2.3.4 Printer actions for various combinations of attributes

This section lists the Printer actions for various combinations of "save-location" (uri), "save-name" (name(MAX)), "save-location-supported" (1setOf uri), existing saved jobs, ipp-attribute-fidelity (boolean), "user-defined-values-supported" (1setOf keyword) and Printer actions.

The possible Printer actions are (see above descriptions of "save-location" and "save-name" for details):

1. reject: The Printer rejects the Job Creation request, returns the ‘client-error-attributes-or-values-not-supported’ status code and the “job-save-disposition” attribute and value in the Unsupported Attributes group.

2. accept-save: The Printer accepts the Job Creation request, returns the ‘successful-ok’ status code, and attempts to create all necessary files and sub-directories.

3. accept-substitute: The Printer accepts the Job Creation request, returns the ‘successful-ok-ignored-or-substituted-attributes’ status code and the “job-save-disposition” attribute and value in the Unsupported Attributes group, and, DEPENDING ON IMPLEMENTATION: (1) ignores the attribute and does not perform the save, (2) puts the job in the ‘held’ state and let the operator fix the problem, either by changing the job’s “save-location” attribute value or adding the value to the Printer’s “save-location-supported” attribute, or (3) substitutes the job’s “save-location” value with one of the values of the Printer’s “save-location-supported” values.

4. accept-hold: The Printer accepts the Job Creation request, returns the ‘successful-ok’ status code, and holds the job for the operator to resolve the conflict so that the job can be saved.

Table 4 shows the Printer actions for the various possible value combinations of the other Job and Printer attributes. In the example, there is already one saved job: ‘a/bar’. The Printer supports three save-location directories: a, b, c.

Table 4 - Printer actions for various Job and Printer attribute values

MIME Media Type	save-name	save-location-supported	existing saved jobs	ipp-attribute-fidelity	user-defined-values-supported	Printer action
a	foo	a, b, c	a/bar	T | F	none	accept-save
a	d/foo	a, b, c	a/bar	T | F	none	accept-save
a	bar	a, b, c	a/bar	T | F	none	accept-hold
d	foo	a, b, c	a/bar	F	none	accept-substitute
d	foo	a, b, c	a/bar	T	none	reject
a/d	foo	a, b, c	a/bar	F	none	accept-substitute
a/d	foo	a, b, c	a/bar	T	none	reject
a	foo	a, b, c	a/bar	T | F	save-location	accept-save
a	d/foo	a, b, c	a/bar	T | F	save-location	accept-save
a	bar	a, b, c	a/bar	T | F	save-location	accept-hold
d	foo	a, b, c	a/bar	F	save-location	accept-hold
d	foo	a, b, c	a/bar	T	save-location	accept-hold
a/d	foo	a, b, c	a/bar	F	save-location	accept-hold
a/d	foo	a, b, c	a/bar	T	save-location	accept-hold

5.7.2 job-save-disposition-default (collection)

The "job-save-disposition-default (collection)" attribute specifies the “job-save-disposition” member attributes and values that the Printer will provide, if any, if the client omits the "job-save-disposition" collection attribute. A Printer MUST support the same member attributes for these default attributes as it supports for the corresponding Job Template attributes (see Table 2).

The "save-info" (1setOf collection) member attribute of the “job-save-disposition-default” Printer attribute specifies the save information member attributes and values that the Printer will provide, if any, if the client omits the “job-save-disposition” Job Template attribute entirely. A Printer MUST support the same member attributes for these default attributes as it supports for the corresponding Job Template attributes (see Table 2 and Table 3), with the single exception, that the “save-name” member attribute MUST NOT be supported (since the Printer MUST use the “job-name” attribute to generate the default “save-name”).

If the client supplies the “job-save-disposition” Job Template attribute, but omits the “save-info” member attribute, the Printer uses its “save-location-default” and “save-disposition-default” Printer attributes and the job’s “job-name” attribute to create the job’s “save-info” member attribute, rather

than using the “save-info” member attribute of the Printer’s “job-save-disposition-default” attribute (see "save-info" description above).

5.7.3 job-save-disposition-supported (1setOf type2 keyword)

The "job-save-disposition-supported (1setOf type2 keyword)" attribute identifies the names of the member attributes supported in the "job-save-disposition" collection attribute, i.e., the names of the member attributes in Table 2 that the Printer supports. The RECOMMENDED way for a client to determine whether or not a Printer support saving jobs, is to query this Printer attribute.

5.8 pdl-init-file (1setOf collection)

This optional Job Template attribute controls initialization of the Printer’s Page Description Language (PDL) interpreter. This attribute specifies the files that the Printer uses to initialize the PDL interpreter before it starts interpreting each Input Document (i.e. file) in a Job with one exception. When the value of "multiple-document-handling" is 'single-document' or ‘single-document-new-sheet’, a PDL interpreter processes all of the Input-Documents of a Job as if the Input Documents were a single file. In this case, the PDL interpreter is initialized just before processing the first Input Document only.

Some PDLs (i.e. PostScript and HP PCL) have the ability to define reusable macros, code fragments, fonts, and other resources. These components may reside either within the body of the document file to be printed or within files that can be sent to the Printer’s PDL interpreter separately. Having the ability to send these “pre-initialization” (prep) files separately from the main body of the job to be printed, enables an application to “pre-condition” the PDL interpreter with these definitions ahead of time. This in turn makes it possible to reduce the size of a given document data file to be printed and to reuse the components on other jobs.

The exact mechanism by which an initialization file (or list of initialization files) is installed on the Printer is implementation dependent and outside of the current scope of IPP.

Upon receipt of the "pdl-init-file" collection attribute(s) the Printer will send the initialization file (or list of files) referenced to the Printer’s PDL interpreter for processing BEFORE starting to process the document(s) attached to the job (in the Print-Job request) or referenced by the job (in the Print-URI request).

If the Printer receives more than one "pdl-init-file" collection in a given print request, then the list of initialization files will be sent to the Printer’s PDL interpreter in the order that they are encountered in the Job Template attributes.

5.8.1 Member Attributes for the "pdl-init-file" Job Template Attribute

Table 5 lists the member attributes of the "pdl-init-file" Job Template attribute and specifies whether Clients MUST supply them in collection values and whether Printers MUST support them if supporting this collection attribute. The following sub-sections define these member attributes.

Table 5 - "pdl-init-file" member attributes

Member Attribute Name	Attribute Syntax	Request	Printer Support
pdl-init-file-location	uri	MAY	MUST
pdl-init-file-name	name(MAX)	MUST	MUST
pdl-init-file-entry	name(MAX)	MAY	MAY

5.8.1.1 pdl-init-file-location (uri)

This member attribute must be an Absolute URI [RFC 2396] that specifies the path to the directory where the initialization file to be sent to the Printer's PDL interpreter will be found. According to RFC 2396, an Absolute URI MUST have a URI scheme, MAY have an authority (host name) component, and MUST have an absolute path (e.g., 'ftp://printhost.printco.com/var/spool/jobinitfiledir/initfile1' or 'file:///jobinitfiledir/initfile1').

5.8.1.1.1 pdl-init-file-location-supported (1setOf uri)

The "pdl-init-file-location-supported (1setOf uri)" Printer attribute specifies the path(s) to the directory (directories) of the supported initialization file(s) that the client MAY ask the Printer to send to its PDL interpreter prior to processing the document.

5.8.1.2 pdl-init-file-name (name(MAX))

This member attribute specifies the name of the initialization file within the directory specified by the "pdl-init-file-location" member attribute that the Printer MUST send to its PDL interpreter prior to processing the document..

A well-formed request MUST include at least the name of the initialization file to be used (i.e. the "pdl-init-file-location" or directory containing the initialization file need not be supplied by the client and in fact, might not be known to the client). The client can supply a "pdl-init-file-name" attribute value that is either a simple file name or a relative path where each component of the path is separated by a "/" character. The "pdl-init-file-name" member attribute value concatenated with the "pdl-init-file-location" attribute value (supplying a "/" between them, if necessary) specifies a Virtual File Name (see definition in Section 2.2) that will be associated with the saved job. For example, if pdl-init-file-location is 'a/b' or 'a/b/' and "pdl-init-file-name" is 'c/d', the resulting Virtual File name is: 'a/b/c/d'.

5.8.1.2.1 pdl-init-file-name-supported (1setOf name(MAX))

The "pdl-init-file-name-supported (1setOf name(MAX))" Printer attribute specifies the name(s) of the supported initialization file(s) that the client MAY ask the Printer to send to its PDL interpreter prior to processing the document.

5.8.1.3 pdl-init-file-entry (name(MAX))

This member attribute is an optional member of the collection that if present specifies an entry point within the init file that the PDL interpreter starts at.

5.8.1.3.1 pdl-init-file-entry-supported (1setOf name(MAX))

The "pdl-init-file-entry-supported (1setOf name(MAX))" Printer attribute specifies the name(s) of the supported initialization file(s) that the client MAY ask the Printer to send to its PDL interpreter prior to processing the document.

5.8.2 pdl-init-file-default (1setOf collection)

pdl-init-file-default (1setOf collection): The "pdl-init-file-default" Printer attribute collection specifies the default initialization file (or list of files) and related information that the Printer will use if the client omits the "pdl-init-file" Job Template attribute in a Job Creation operation. The member attributes are defined in Table 5. A Printer MUST support the same member attributes and values for this default collection attribute as it supports for the corresponding "pdl-init-file" Job Template attribute.

A "pdl-init-file-name" within "pdl-init-file-default" collection with a “0 length value” will be used to indicate that Printer has no default initialization file (see description of “none” values in Section 2.7 in [prod-print]).

5.8.3 pdl-init-file-supported (1setOf type2 keyword)

pdl-init-file-supported (1setOf type2 keyword): The "pdl-init-file-supported" Printer attribute identifies the keyword names of the member attributes supported in the "pdl-init-file" collection Job Template attribute, i.e., the keyword names of the member attributes in Table 5 that the Printer supports.

5.9 proof-print (collection)

The OPTIONAL "proof-print" attribute allows a user to specify the attributes of the proof print(s) of the job that are to be printed prior to the printing the full run of the job. After the proof prints have been produced, the Printer changes the "proof-print-copies" member attribute to zero, and the job placed in the 'pending-held' state until released by the user or operator. The job is released from the

held state by using the Release-Job operation either through a client application or directly at the Printer device.

If a "proof-print-copies" member attribute value other than 0 is provided in a job creation operation, proof print copies of the job are printed and the job is then placed into the 'pending-held' state, and the 'proof-print-wait' value is added to the "job-state-reason" attribute.

Other jobs may be printed between the proof print and the release of the Proof Job from the 'pending-held' state. If the proof print is accepted by the user while another job is printing, the Proof Job MUST NOT resume printing until the current job is done printing. The proofed job SHOULD be the next job printed after the current job, unless there is another job in the Printer which has a higher priority than the proofed job as determined by the "job-priority" attribute.

Note: The “job-save-disposition” Job Template attribute (see above) details a method for doing a "hard-proof" by saving a job and resubmitting the saved job. If a Printer does not implement “job-save-disposition”, or would rather that proof jobs stay active (in the not-complete state) in the Printer, this “proof-print” attribute provides a simpler method of specifying a "hard-proof".

5.9.1 Member Attributes for the "proof-print" Job Template Attribute

Table 6 lists the member attributes of the "job-sheets-col" collection attribute:

Table 6 - "proof-print" member attributes

Member Attribute Name	Attribute Syntax	Request	Printer Support
proof-print-copies	integer (0:MAX)	MUST	MUST
media	type3 keyword | name(MAX)	MUST be one or the other, but NOT both	MUST
media-col	collection		MAY

5.9.1.1 proof-print-copies (integer (0:MAX))

The "proof-print-copies" member attribute specifies the number of copies the Printer MUST produce in the proof job. The proof print(s) are produced using the Job Template attributes specified with the Job, except any overridden by member attributes in this collection.

If the "proof-print-copies" value is 0, then no proof prints are produced.

After the requested number of proof prints have been produced by the Printer, then the Printer sets the value of "proof-print-copies" to 0, adds the 'proof-print-wait' value to the "job-state-reasons" attribute, and change the "job-state" to be 'pending-held'. A client may request additional proofs of this job by performing a Set-Job-Attributes operation on the "proof-print-copies" attribute. The user

may also select to Set other Job Template attribute before releasing the Job. The user then performs a Release-Job operation, and the Printer changes the "job-state" to 'pending' and removes the 'proof-print-wait' job state reason.

5.9.1.2 media (type3 keyword | name(MAX)) or media-col (collection)

Either the "media" (see [RFC2911]) or the "media-col" member attribute is used to indicate the media that the Printer MUST use for the specified "proof-print-copies" of the Proof Job. The member attributes are the same as those for the "media-col" attribute defined in [prod-print] and in section 8.4 of this specification.

The client MUST supply either the "media" or the "media-col" member attribute, but NOT both. If the client supplies such a mal-formed request by supplying neither or both, the Printer MUST (depending on implementation) either (1) reject the request and return the 'client-error-bad-request' status code (see [RFC2911]) or (2) use either the "media" or the "media-col" member attribute, independent of the value of the "ipp-attribute-fidelity" attribute supplied by the client.

Since this "media" member attribute has the same name as the "media" Job Template attribute (defined [RFC2911]), the "media-supported" (1setOf (type3 keyword | name(MAX))) Printer attribute identifies the values of this "media" member attribute (as well as the values of the "media" Job Template attribute) that the Printer supports, i.e., the names of the supported media. A value that is provided for the "media" member attribute in the collection would have the same effect as if the job were submitted with that value as the value of the "media" Job Template attribute.

Since this "media-col" member attribute has the same name as the "media-col" Job Template attribute (defined in [prod-print] and in section 8.4 of this specification), the "media-col-supported" Printer attribute identifies the keyword names of the member attributes supported in this "media-col" member attribute (as well as the keyword names of the "media-col" Job Template attribute), i.e., the names of the member attributes that the Printer supports.

5.9.2 proof-print-default (collection)

The "proof-print-default" (collection) Printer attribute specifies the default value of "proof-print" MUST use when not supplied in a request. A Printer MUST support the same member attributes for this default collection as it supports for the corresponding "proof-print" Job Template attribute.

5.9.3 proof-print-supported (1setOf type2 keyword)

The "proof-print-supported" (1setOf (type2 keyword)) Printer attribute specifies which member attributes of "proof-print" that the Printer supports.

6 Job Description Attributes

6.1 job-printer-make-and-model (text(127))

This attribute identifies the make and model of the output device which saved this job. The values are the same as the corresponding "printer-make-and-model" Printer Description attribute (see "printer-make-and-model" description in [RFC2911]). If this attribute is present in the saved Job, then only Printers that have the same make and model will be able to print the saved job with identical appearance. Other make and models MAY be able to print the job, but not necessarily with identical appearance, provided that the Printer supports the “save-document-format” value of the saved job. If this attribute is not present in the saved Job Instructions and the Job object after saving, then the saved job was saved in a format that can be printed on any output device and will generate the same appearance provided that the Printer supports the “save-document-format” value of the saved job.

If the printer implementation supports the Job Save capability (see "job-save-disposition" attribute in section 5.7), then it is REQUIRED that the printer support the "printer-make-and-model" Printer Description attribute.

7 Printer Description Attributes

7.1 job-creation-attributes-supported (1setOf type2 keyword)

This OPTIONAL extension enables an IPP client to query the printer for the set of job attributes that can be set by the client during a Create-Job, Print-Job, Validate-Job, or Print-URI operation.

This extension allows the IPP client to dynamically determine all the job attributes that it can specify at the time of job creation.

The list of attribute names in “job-creation-attributes-supported” MUST include:

- all Job Template attributes that may be supplied by the client at the job level
- all operation attributes that are written to the job object as job description attributes (e.g., "job-name") at the job level

The list of attribute names in “job-creation-attributes-supported” MUST NOT include:

- collection member attribute names

  Note: The client can determine which member attributes of “xxx” collection attributes are supported by querying the “xxx-supported” Printer attribute (see [coll]).

- operation attributes that are not job attributes

NOTE: The only mechanism previously available for a IPP client to query the Printer for supported attributes is to specify the ‘job-template’ group value on a Get-Printer-Attributes operation. This has been problematic because: 1) it returns the entire list of “xxx-default” and “xxx-supported” attributes and values which will be excessively burdensome for production printing systems with extensive functionality, and 2) it does not include operation attributes that are written to the job object as job description attributes (i.e., "job-name"). [NOTE: See also the description of the Get-Printer-Attributes operation in [RFC2911] and the distinction between "xxx-default", "xxx-ready", and "xxx-supported" attributes when "xxx" is any Job Template attribute that a client can supply as a top-level attribute vs. (see [coll]) when "xxx" is a member attribute for a Job Template 'collection' attribute (e.g. The "media-color-supported" Printer attribute lists the values of the “media-color” member attribute of the “media-col” collection attribute, but does not belong to the 'job-template' attribute group, or to the ‘printer-description’ attribute group).]

7.2 job-password-supported (integer(0:127))

The "job-password-supported" attribute indicates the maximum length that the Printer will accept for the unencrypted password which the client will encrypt as the value of the "job-password" Operation Attribute. A conforming Printer MUST be able to accept 127 octets without truncation. However, a Printer MAY be implemented as a gateway to another print system that cannot accept the full 127-octet range, in which case the client MUST NOT allow an unencrypted password greater than the length specified by this attribute.

7.3 job-password-encryption-supported (1setOf (type3 keyword | name(MAX)))

The "job-password-encryption-supported" Printer Description attribute specifies which encryption methods the Printer supports for Secure Print.

If the operation attribute "job-password" is supported, then this attribute MUST be supported .

Standard keyword values are:

Keyword	Description
'none'	The "job-password" attribute value is passed in the clear. No encryption has been applied. This value might also be used when the entire Operation is sent over a secure connection.
'md2'	The encryption method uses the MD2 hash algorithm defined in RFC 1319.
'md4'	The encryption method uses the MD4 hash algorithm defined in RFC 1320.
'md5'	The encryption method uses the MD5 hash algorithm defined in RFC 1321.
'sha'	The encryption method uses the Secure Hash Algorithm defined by the National Institute of Standards and Technology.

7.4 job-spooling-supported (type2 keyword)

This attribute indicates whether or not jobs are spooled before the document data is interpreted (RIPped). In other words, this attribute indicates when jobs are processed by the Printer with respect to when the Printer receives and returns responses to Job Creation requests (i.e., Print-Job, Print-URI), receives and returns responses to Document Creation requests (i.e., Send-Document and Send-Uri requests) and "receives" or "fetches" such document data.

This attribute does not describe when resources or OPI images are fetched. Printers MUST pre-scan and fetch resources and OPI images according to the "resource-pre-scan", "opi-image-pre-scan", and "opi-image-insertion" Job Template attributes (see descriptions in [pwg-color-imaging]) which is independent from the "job-spooling-supported" Printer Description attribute. This attribute does not describe when the Printer fetches images that are references from within a document.

If the Printer supports this attribute, the value supported depend on implementation. If the Printer does not support this attribute, the behavior is implementation dependent.

The Get-Printer-Supported-Values operation (see description in [set-ops]) returns a '1setOf type2 keyword' so that all possible values that the implementation is capable of supporting are indicated.

The standard keyword values are:

Keyword	Description
'spool'	The Printer starts processing a job until the Printer has (1) accepted and responded to the Job Creation request and all Document Creation requests (for a multi-document job) and (2) has "received" or "fetched" all document data for the job, i.e., spool rather than stream.
'stream'	The Printer starts processing a job (1) before the Printer has accepted all Document Creation requests and (2) before the Printer has "received" or "fetched" all document data, i.e., stream rather than spool
'automatic'	The Printer chooses whether to process a job before or after the Printer has accepted all Document Creation requests and has "received" or "fetched" all document data, i.e., the Printer MAY spool and/or stream depending on policy and other factors, such as the size of the job, including a combination of spooling and streaming.

7.5 max-save-info-supported (integer(1:MAX))

This attribute specifies the maximum number of “save-info” member attribute collections that a Printer can accept in a job request. If the number of “save-info” member attribute collections supplied by a client in a Job Creation operation exceeds the value of this attribute, the Printer MUST accept or reject the request as described in the "save-info" description in the "job-save-disposition" description (see description in section 5.7). If the Printer only supports one “save-info” collection, then the Printer MAY either (1) support this attribute with a value of ‘1’ or (2) omit support of this attribute. The RECOMMENDED way for a client to determine whether or not a Printer supports saving jobs, is to query the “job-save-disposition-supported” Printer attribute (see description in section 5.7.3).

7.6 media-col-database (1setOf collection)

This OPTIONAL extension enables a IPP client to query the set of pre-defined media collections available in the printer's media database, if the "media-col" attribute is supported. This attribute is identical in format and syntax to "media-col-ready", but returns the entire set of pre-defined media collections known by the printer instead of just the media collections currently in the printer trays. If this attribute is supported, the members of the collections correspond to the supported members of the "media-col" attribute.

The Printer SHOULD NOT return this attribute in the response to the Get-Printer-Attributes operation when the client requested the 'all' or 'printer-description' group names. Therefore the client must request this attribute explicitly in order to get the media collections. The reason for this recommendation is that the amount of data returned in the response would be very large when combined with all of the other attributes.

7.7 printer-detailed-status-messages (1setOf text(MAX))

This OPTIONAL attribute specifies additional detailed and technical information about the printer, in the same way "job-detailed-status-messages" provides additional information about a job. Printer-specific information such as fault and warning messages can be captured, and the 1setOf syntax supports multiple messages. The Printer NEED NOT localize the message(s), since they are intended for use by the system administrator or other experienced technical persons. Clients MUST NOT attempt to parse the value(s) of this attribute.

7.8 which-jobs-supported (1setOf type2 keyword)

This OPTIONAL attribute enables a client to query the printer for the set of values that can be used for the "which-jobs" operation attribute of the Get-Jobs operation. Current allowed values for "which-jobs" are 'completed' and 'not-completed', and the "which-jobs" extensions described above extend the values to include keywords corresponding to the symbolic names of all IPP "job-state" enum values. (See section 8.2)

If this attribute is supported, then the Printer MUST support both 'completed' and 'not-completed' as valid values.

8 Additional Values and Semantics for Existing IPP Attributes

This section defines additional values for existing attributes. Many of the existing attributes are specified by including the entire original description from [RFC2911] with the additions indicates with revision marks. In some case, additional value from other approved documents are included with the source indicated as a reference.

8.1 Additional values and semantics for the "pdl-override-supported" Printer Description attribute

The additional values are represented by revisions to the original description for “pdl-override-supported” from [RFC2911].

8.1.1 pdl-override-supported (type2 keyword)

This REQUIRED Printer attribute expresses the ability for a particular Printer implementation to either attempt to override document data instructions with IPP Job attributes or not.

This attribute takes on the following keyword values:

- 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values take precedence over embedded instructions in the document data, however there is no guarantee.
- 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP attribute values take precedence over embedded instructions in the document data.
- 'guaranteed': This value indicates that the Printer object:
  1) guarantees that the IPP Job Template attribute values take precedence over instructions of any form embedded anywhere in the document data.
  2) and guarantees that no PDL instruction is performed that does not correspond to a current value in the corresponding "xxx-supported" Printer attribute. In such a situation, the Printer MAY (a) abort the job, (b) stop the Printer, (c) hold the job, or (d) substitute on the fly with one of the current values of its "xxx-supported" attribute. This choice MAY be fixed by the implementation or configurable by the system administrator and MAY depend on the attribute.
  3) provides the "queue override" semantics as described below, if the "xxx-supported" Printer attribute is configured with only a single value.

See [RFC2911] for a full description of how this attribute interacts with and affects other IPP attributes, especially the "ipp-attribute-fidelity" attribute.

Additional semantics for 'guaranteed' value of "pdl-override-supported" attribute

If a Printer's "pdl-override-supported" is 'guaranteed', then the Printer MUST follow these additional semantics for Job Creation requests and Get-Job-Attributes or Get-Jobs responses.

Whether or not the client supplied the "xxx" Job attribute in the Job Creation operation, the Printer MUST accept or reject the job based on "ipp-attribute-fidelity" as usual. However, if "ipp-attribute-fidelity" was 'false' and the supplied "xxx" value did not match any of the "xxx-supported" values, the Printer MUST accept the job as usual, but MUST NOT store that attribute on the Job object. The Printer MUST return an "xxx" Job attribute in subsequent Get-Job-Attributes and Get-Jobs responses according to the following rules:

(a) If the client did not supply the "xxx" Job attribute in the Job Creation operation and the current value of the Printer's "xxx-supported" attribute has only a single value (the so-called queue override case), then the Printer MUST return an "xxx" attribute with that current single value in a Get-Job-Attributes or Get-Jobs, response, as if the client had supplied "xxx" Job attribute with that value.

Rationale for this new behavior: because the printed result will be the same as if the client had supplied the "xxx" Job attribute with that single value (since that value will override a corresponding PDL instruction), the response should reflect those (new) semantics.

(b) If the client did not supply the "xxx" Job attribute in the Job Creation operation and the current value of the Printer's "xxx-supported" attribute has more than one value, then the Printer MUST NOT return that "xxx" attribute in a Get-Job-Attributes or Get-Jobs response.

Rationale: Same behavior as for the other values of "pdl-override-supported".

(c) If the client did supply the "xxx" Job attribute in the Job Creation operation and the current value of the Printer's "xxx-supported" attribute has only a single value (the so-called queue override case), then the Printer MUST return that "xxx" Job attribute with the Printer's current "xxx-supported" value in a Get-Job-Attributes or Get-Jobs, response, as if the client had supplied "xxx" with that value.

Rationale for this *new* behavior: because the behavior of the Printer is to enforce that "xxx" Job attribute with the Printer's current single (override) value.

(d) If the client did supply the "xxx" attribute in the Job Creation operation and the current value of the Printer's "xxx-supported" attribute has more than one value and one of them match, then the Printer MUST return that "xxx" Job attribute with the client-supplied value in a Get-Job-Attributes or Get-Jobs response.

Rationale: Same behavior as for the other values of "pdl-override-supported".

(e) If the client did supply the "xxx" attribute in the Job Creation operation and the current value of the Printer's "xxx-supported" attribute has more than one value but none of them match, then the Printer MUST NOT return that "xxx" Job attribute with the client-supplied value in a Get-Job-Attributes or Get-Jobs response. It MUST either (1) not return the attribute at all or return it with one of the supported values, depending on implementation.

Rationale: Same behavior as for the other values of "pdl-override-supported".

The above five cases are represented in Table 7.

Table 7 - Rules for 'guaranteed' value of "pdl-override-supported" attribute

Client supplies	Printer supports "xxx-supported"	Printer's response to Job query request	Queue override ?
omits "xxx"	'a'	"xxx" = 'a'	yes
omits "xxx"	'a', 'b'	nothing is returned	no
supplies "xxx" = ?	'a'	"xxx" = 'a'	yes
supplies "xxx" = 'a'	'a', 'b'	"xxx" = 'a'	no
supplies "xxx" = 'c'	'a', 'b'	either nothing is returned or "xxx" = one of the supported values, depending on implementation	no

Examples of queue override

The queue override extension allows an implementation to guarantee that only advertised supported attribute values are actually performed, and that unsupported values embedded as instructions in the document data, will not be performed. This extension is also a further step in making a Printer object indicate what it really supports in both the protocol and the PDL and what it does not, i.e., "truth in advertising". In other words, the absence of a value in the Printer's "xxx-supported" attribute indicates a guarantee that the value cannot be performed.

For example, an administrator that wants to force two sided printing could set the Printer's "sides-supported" to the two values: 'two-sided-long', 'two-sided-short' (see [RFC2911]). Thus a PDL that contained an embedded simplex (one-sided) instruction would be overridden in one of the following ways: (1) aborted, (2) held, or (3) automatically forced to two-sided, depending on implementation and/or site policy.

As another example, an administrator could set up an IPP Printer that always staples by setting the "finishings-supported" to 'staple' only, i.e. by removing the 'none' keyword value (see [RFC2911]). In order to allow jobs not to be stapled on the same physical piece of hardware, the administrator would use fan-in (if supported) to set up another IPP Printer that does include the 'none' keyword value in its "finishings-supported" attribute.

The values of the Printer's "xxx-supported" attribute are not only what the Printer will accept in the print protocol (e.g. IPP) as attributes, but also as corresponding values of embedded PDL instructions. If a value that is not in the "xxx-supported" list is encountered in either the print protocol or the PDL, it MUST NOT be performed (no matter what the value of "ipp-attribute-fidelity" is).

The four combinations of unsupported IPP attribute values and unsupported PDL instruction values are:

(1) unsupported attribute value in the protocol and "ipp-attribute-fidelity" = 'false' (independent of "pdl-override-supported"):

The Printer object MUST accept the Job with the unsupported attribute value. However, the Printer object will always return only one of its "xxx-supported" values in response to Get-Jobs or Get-Job-Attributes. Depending on implementation, the Printer can perform this substitution once at job submission time, or each time the job is queried.

(2) unsupported attribute value in the protocol and "ipp-attribute-fidelity" = 'true' (independent of "pdl-override-supported"):

The Printer MUST reject the job.

(3) unsupported embedded instruction value in the PDL and "pdl-override-supported" = 'attempted' or 'not-attempted' (and independent of "ipp-attribute-fidelity"):

When an implementation encounters a value that is not in the "xxx-supported" list, it makes no special action and allows it to be performed. See the definition of "Supports" in [RFC2911] that indicates that performing a PDL instruction that is not supported in the protocol is an allowed behavior.

(4) (new case) unsupported embedded instruction value in the PDL and "pdl-override-supported" = 'guaranteed' (and independent of "ipp-attribute-fidelity"):

The unsupported embedded instruction value MUST NOT be performed. The behavior is implementation-dependent if an unsupported embedded instruction value is encountered. For example, the Printer MAY abort the job or substitute on the fly with one of the current values of its "xxx-supported" attribute. This choice MAY be fixed by the implementation or configurable by the system administrator and MAY depend on the Job Template attribute.

8.2 Additional values for the "which-jobs" Operation attribute and the “which-jobs-supported” Printer Description attribute

This section defines additional values for the IPP "which-jobs" (type2 keyword) Operation attribute of the Get-Jobs operation (see [RFC2911] section 3.2.6) and the “which-jobs-supported” (1setOf type2 keyword) Printer Description attribute (see section 7.8) to include keywords corresponding to the symbolic names of all IPP “job-state” enum values (see [RFC2911] section 4.3.7) and an ‘all’ value.

The additional values are represented by revisions to the original description for “which-jobs” from [RFC2911]. The [RFC2911] values for “which-jobs” are ‘completed’ and ‘not-completed’.

8.2.1 which-jobs (type2 keyword)

The client OPTIONALLY supplies this attribute in the Get-Jobs operation. The Printer object MUST support this attribute. It indicates which Job objects MUST be returned by the Printer object in the Get-Jobs Request.

The values for this attribute are:

'completed': This includes any Job object whose state is 'completed', 'canceled', or 'aborted'.

'not-completed': This includes any Job object whose state is 'pending', 'processing', 'processing-stopped', or 'pending-held'.

'pending': This includes any Job object whose state is 'pending'. This value corresponds to the symbolic name of the corresponding value for the “job-state” attribute.

'pending-held': This includes any Job object whose state is 'pending-held'. This value corresponds to the symbolic name of the corresponding value for the “job-state” attribute.

'processing': This includes any Job object whose state is 'processing'. This value corresponds to the symbolic name of the corresponding value for the “job-state” attribute.

'processing-stopped': This includes any Job object whose state is 'processing-stopped'. This value corresponds to the symbolic name of the corresponding value for the “job-state” attribute.

'canceled': This includes any Job object whose state is 'canceled'. This value corresponds to the symbolic name of the corresponding value for the “job-state” attribute.

'aborted': This includes any Job object whose state is 'aborted'. This value corresponds to the symbolic name of the corresponding value for the “job-state” attribute.

'all': This includes all Job objects, that is any Job object that is considered 'completed' or 'not-completed as defined by these values above.

A Printer object MUST support both the 'completed' and 'not-completed' values. However, if the implementation does not keep jobs in the 'completed', 'canceled', and 'aborted' states, then it returns no jobs when the 'completed' value is supplied.

If a client supplies some other value, the Printer object MUST copy the attribute and the unsupported value to the Unsupported Attributes response group, reject the request, and return the 'client-error-attributes-or-values-not-supported' status code.

If the client does not supply this attribute, the Printer object MUST respond as if the client had supplied the attribute with a value of 'not-completed'.

For the order that Jobs are returned, see the description of the Get-Jobs Response in [RFC2911].

8.3 Additional values and semantics for the "job-state-reasons" Job Description attribute

The additional values and semantics are represented by revisions to the original description for “job-state-reasons” in [RFC2911] section 4.3.8 and the values defined in [ops-set2], [prod-print], and [override].

8.3.1 job-state-reasons (1setOf type2 keyword)

This REQUIRED attribute provides additional information about the job's current state, i.e., information that augments the value of the job's "job-state" attribute.

These values MAY be used with any job state or states for which the reason makes sense. Some of these value definitions indicate conformance requirements; the rest are OPTIONAL. Furthermore, when implemented, the Printer MUST return these values when the reason applies and MUST NOT return them when the reason no longer applies whether the value of the Job's "job-state" attribute changed or not. When the Job does not have any reasons for being in its current state, the value of the Job's "job-state-reasons" attribute MUST be 'none'.

Note: While values cannot be added to the 'job-state' attribute without impacting deployed clients that take actions upon receiving "job-state" values, it is the intent that additional "job-state-reasons" values can be defined and registered without impacting such deployed clients. In other words, the "job-state-reasons" attribute is intended to be extensible.

The following standard keyword values are defined. For ease of understanding, the values are presented in the order in which the reasons are likely to occur (if implemented), starting with the 'none' value:

'none': There are no reasons for the job's current state. This state reason is semantically equivalent to "job-state-reasons" without any value and MUST be used when there is no other value, since the 1setOf attribute syntax requires at least one value. [RFC2911]

'job-hold-until-specified': The value of the job's "job-hold-until" attribute was specified with a time period that is still in the future. The job MUST NOT be a candidate for processing until this reason is removed and there are no other reasons to hold the job. This value SHOULD be supported if the "job-hold-until" Job Template attribute is supported. [RFC2911]

'job-password-wait': The job is currently being held until the correct password is entered at the device. If the "job-password" Operation attribute (See section 4.1) is supported, this value MUST be supported.

'proof-print-wait': The job is currently being held until the operator verifies the output of the proof print and performs a Release-Job operation. If the "proof-print" Job Template attribute (See section 5.9) is supported, this value MUST be supported.

'resources-are-not-supported': At least one of the resources needed by the job, such as media, fonts, resource objects, etc., is not supported on any of the physical printers for which

the job is a candidate. This condition MAY be detected when the job is accepted, or subsequently while the job is pending or processing, depending on implementation. The job may (1) remain in its current state, (2) be moved to the 'pending-held' state, depending on implementation and/or job scheduling policy, or (3) scheduled normally, but the Printer is put into the 'stopped' state when the job is attempted to be processed on the Printer. This value is intended for use with an implementation that supports the "user-defined-values-supported" Printer attribute (defined in [prod-print]) which allows a job to be accepted with an unsupported 'name' value. [prod-print]

'resources-are-not-ready': At least one of the resources needed by the job, such as media, fonts, resource objects, etc., is not ready on any of the physical printer's for which the job is a candidate. This condition MAY be detected when the job is accepted, or subsequently while the job is pending or processing, depending on implementation. The job may remain in its current state or be moved to the 'pending-held' state, depending on implementation and/or job scheduling policy. [RFC2911]

'job-incoming': Either (1) the Printer has accepted the Create-Job operation and is expecting additional Send-Document and/or Send-URI operations or (2) the Printer is retrieving/accepting document data as a result of a Print-Job, Print-URI, Send-Document or Send-URI operation. [RFC2911]

'job-spooling': Same as 'job-incoming' with the specialization that the Printer is spooling the document data before processing it. This value corresponds to the 'spool' or 'automatic' value of the "job-spooling-supported" Printer Description attribute.

'job-streaming': Same as 'job-incoming' with the specialization that the Printer is processing the document data as it is being received (that is, the job is not being spooled, but rather is being processed in chunks by the output device and is being imaged during reception). This value corresponds to the 'stream' or 'automatic' value of the "job-spooling-supported" Printer Description attribute.

'job-data-insufficient': The Create-Job operation has been accepted by the Printer, but the Printer is expecting additional document data before it can move the job into the 'processing' state. If a Printer starts processing before it has received all data, the Printer removes the 'job-data-insufficient' reason, but the 'job-incoming' remains. If a Printer starts processing after it has received all data, the Printer removes the 'job-data-insufficient' reason and the 'job-incoming' at the same time. [RFC2911]

'unsupported-compression': The job was aborted by the system because the Printer determined while attempting to decompress the document-data's that the compression is actually not among those supported by the Printer. This value MUST be supported, since "compressions is a REQUIRED operation attribute. [RFC2911]

'compression-error': The job was aborted by the system because the Printer encountered an error in the document-data while decompressing it. If the Printer posts this reason, the document-data has already passed any tests that would have led to the 'unsupported-compression' job-state-reason. [RFC2911]

'document-access-error': After accepting a Print-URI or Send-URI request, the Printer could not access one or more documents passed by reference. This reason is intended to cover

any file access problem, including file does not exist and access denied because of an access control problem. The Printer MAY also indicate the document access error using the "job-document-access-errors" Job Description attribute (see the description of "job-document-access-errors" above). Whether the Printer aborts the job and moves the job to the 'aborted' job state or prints all documents that are accessible and moves the job to the 'completed' job state and adds the 'completed-with-errors' value in the job's "job-state-reasons" attribute depends on implementation and/or site policy. This value SHOULD be supported if the Print-URI or Send-URI operations are supported. [RFC2911]

'**unsupported-document-format**': The job was aborted by the system because the document-data's document-format is not among those supported by the Printer. If the client specifies the document-format as 'application/octet-stream', the printer MAY abort the job and post this reason even though the format is a member of the "document-format-supported" printer attribute, but not among the auto-sensed document-formats. This value MUST be supported, since "document-format" is a REQUIRED operation attribute. [RFC2911]

'**document-format-error**': The job was aborted by the system because the Printer encountered an error in the document-data while processing it. If the Printer posts this reason, the document-data has already passed any tests that would have led to the 'unsupported-document-format' job-state-reason. [RFC2911]

'**submission-interrupted**': The job was not completely submitted for some unforeseen reason, such as: (1) the Printer has crashed before the job was closed by the client, (2) the Printer or the document transfer method has crashed in some non-recoverable way before the document data was entirely transferred to the Printer, (3) the client crashed or failed to close the job before the time-out period. See the description of the “multiple-operation-time-out” Printer Description attribute in [RFC2911] section 4.3.31. [RFC2911]

'**job-outgoing**': The Printer is transmitting the job to the output device. [RFC2911]

'**job-scheduling**': The printer is in the process of scheduling the processing for the job.

'**job-queued**': Job is in the 'processing' state, but more specifically, the Printer has queued the document data. [RFC2911]

'**job-transforming**': Job is in the 'processing' state, but more specifically, the Printer is interpreting document data and producing another electronic representation. [RFC2911]

'**job-interpreting**': Job is in the 'processing' state, but more specifically, the Printer is interpreting the document data. [RFC2911]

'**job-queued-for-marker**': Job is in any of the 'pending-held', 'pending', or 'processing' states, but more specifically, the Printer has completed enough processing of the document to be able to start marking and the job is waiting for the marker. Systems that require human intervention to release jobs using the Release-Job operation, put the job into the 'pending-held' job state. Systems that automatically select a job to use the marker put the job into the 'pending' job state or keep the job in the 'processing' job state while

waiting for the marker, depending on implementation. All implementations put the job into (or back into) the 'processing' state when marking does begin. [RFC2911]

'job-printing': The output device is marking media. This value is useful for Printers which spend a great deal of time processing (1) when no marking is happening and then want to show that marking is now happening or (2) when the job is in the process of being canceled or aborted while the job remains in the 'processing' state, but the marking has not yet stopped so that impression or sheet counts are still increasing for the job. [RFC2911]

'printer-stopped-partly': The value of the Printer's "printer-state-reasons" attribute contains the value 'stopped-partly'. [RFC2911]

'printer-stopped': The value of the Printer's "printer-state" attribute is 'stopped'. [RFC2911]

'job-suspending': The printer is in the process of moving the job from a processing condition to a suspended condition where other jobs can be processed.

'job-suspended': The job has been suspended while processing using the Suspend-Current-Job operation and other jobs can be processed on the Printer. The Job can be resumed using the Resume-Job operation which removes this value. [ops-set2]

'job-suspended-by-operator': The job has been indefinitely suspended by the printer operator.

'job-suspended-by-user': The job has been indefinitely suspended by the user.

'job-suspended-by-system': The job has been indefinitely suspended by the Printer's system software during normal processing of the job.

'job-resuming': The printer is in the process of moving the job from a suspended condition to a candidate for processing.

'job-canceled-by-user': The job was canceled by the owner of the job using the Cancel-Job request, i.e., by a user whose authenticated identity is the same as the value of the originating user that created the Job object, or by some other authorized end-user, such as a member of the job owner's security group. This value SHOULD be supported. [RFC2911]

'job-canceled-by-operator': The job was canceled by the operator using the Cancel-Job request, i.e., by a user who has been authenticated as having operator privileges (whether local or remote). If the security policy is to allow anyone to cancel anyone's job, then this value may be used when the job is canceled by other than the owner of the job. For such a security policy, in effect, everyone is an operator as far as canceling jobs is concerned. This value SHOULD be supported if the implementation permits canceling by other than the owner of the job. [RFC2911]

'job-canceled-at-device': The job was canceled by an unidentified local user, i.e., a user at a console at the device. This value SHOULD be supported if the implementation supports canceling jobs at the console. [RFC2911]

'aborted-by-system': The job (1) is in the process of being aborted, (2) has been aborted by the system and placed in the 'aborted' state, or (3) has been aborted by the system and

placed in the 'pending-held' state, so that a user or operator can manually try the job again. This value SHOULD be supported. [RFC2911]

'processing-to-stop-point': The requester has issued a Cancel-Job operation or the Printer object has aborted the job, but is still performing some actions on the job until a specified stop point occurs or job termination/cleanup is completed.
If the implementation requires some measurable time to cancel the job in the 'processing' or 'processing-stopped' job states, the IPP object MUST use this value to indicate that the Printer object is still performing some actions on the job while the job remains in the 'processing' or 'processing-stopped' state. After all the job's job description attributes have stopped incrementing, the Printer object moves the job from the 'processing' state to the 'canceled' or 'aborted' job states. [RFC2911]

'service-off-line': The Printer is off-line and accepting no jobs. All 'pending' jobs are put into the 'pending-held' state. This situation could be true if the service's or document transform's input is impaired or broken. [RFC2911]

'job-warnings-detected': If the Printer supports the value ‘job-warnings-detected’, the Printer MUST add it to “job-state-reasons” when it generates the first warning message. That is, a single occurrence of this value is present in the “job-state-reasons” if the Printer has generated one or more warnings. [override]

'job-completed-successfully': The job completed successfully. There were no warnings or errors in printing and no errors in saving. This value SHOULD be supported. [RFC2911]

'job-completed-with-warnings': The print part of the job completed with warnings (whether or not there were save errors). This value SHOULD be supported if the implementation detects warnings. [RFC2911]

'job-completed-with-errors': The print part of the job completed with errors (and possibly warnings too) (whether or not there were save errors). This value SHOULD be supported if the implementation detects errors. [RFC2911]

'job-saving': The printer is transmitting the job to the save location. This is similar to the 'job-printing' value.

'job-saved-successfully': The job was successfully saved.

'job-save-error': The job did not save successfully (whether or not it was printed successfully, printed with warnings, printed with errors, or not printed). The Printer MUST ensure that none of the saved job is accessible, if it was unable to successfully save all of the job.

'job-restartable': This job is retained (see the description of "job-state" above, specifically "Partitioning of Job States") and is currently able to be restarted using the Restart-Job operation (see [RFC2911]). If 'job-restartable' is a value of the job's "job-state-reasons" attribute, then the IPP object MUST accept a Restart-Job operation for that job. This value SHOULD be supported if the Restart-Job operation is supported. [RFC2911]

'**queued-in-device**': The job has been forwarded to a device or print system that is unable to send back status. The Printer sets the job's "job-state " attribute to 'completed' and adds the 'queued-in-device' value to the job's "job-state-reasons" attribute to indicate that the Printer has no additional information about the job and never will have any better information. See the description of the "job-state" attribute above, specifically the portion on "Forwarding Servers". [RFC2911]

NOTE: The semantics of this attribute or the set of valid values may be different for different document formats.

Additional Semantics for "job-state-reasons" attribute for Job Save and Reprint Capability

This section defines additional values that are defined in support of the Job Save and Reprint Capability (see section 3.3, and the description of the "job-save-disposition" attribute in section 5.7).

Many of the existing "job-state-reasons" attribute values can apply to a ‘save-only’ job as well as a job submitted for printing. If the value of the “job-disposition” attribute is 'save-only', the "job-state-reasons" attribute MAY be set with existing values such as 'job-incoming', 'resources-are-not-ready' (such as a font), 'job-interpreting', and 'job-queued'. Likewise, errors that occur on a saved job MAY have existing "job-state-reasons" attribute values such as 'job-data-insufficient', 'document-access-error', 'submission-interrupted', 'job-canceled-by-user', 'aborted-by-system', etc.

There are also new two new "job-state-reasons" keyword values required that support a saved job -- 'job-saving' and 'job-save-error' -- and these have been added to the list defined above. The existing ‘job-completed-successfully’, job-completed-with-warnings’, and ‘job-completed-with-errors’ values are also clarified in the list above for implementations that support the Job Save and Reprint Capability, in such a way that these values remain compatible with Printers and clients that do not support job saving.

Table 8 enumerates the possible combinations of print and save success, warning, and error conditions. Note that two values occur only when there are print warnings/errors and a save error.

Table 8 - Values of "job-state-reasons" attribute for various job conditions

	Not saving	**Save was successful**	**Save was unsuccessful**
Not printing	*<not possible>*	'job-completed-successfully'	'job-save-error'
Print successful	'job-completed-successfully'	'job-completed-successfully'	'job-save-error'
Print warnings	'job-completed-with-warnings'	'job-completed-with-warnings'	'job-completed-with-warnings', 'job-save-error'
Print errors	'job-completed-with-errors'	'job-completed-with-errors'	'job-completed-with-errors', 'job-save-error'
Print warnings	'job-completed-	'job-completed-	'job-completed-

and errors	with-errors'	with-errors'	with-errors', 'job-save-error'

8.4 Additional semantics for the IPP "media-col" Job Template Attribute

The following member attributes in Table 9 and their semantic descriptions have been added to the existing IPP “media-col” Job Template attribute.

Table 9 – Additional "media-col" member attributes

Member Attribute Name	Attribute Syntax	Request	Printer Support
media-tooth	type3 keyword | name(MAX)	MAY	MAY
media-grain	type3 keyword | name(MAX)	MAY	MAY
media-material	type3 keyword | name(MAX)	MAY	MAY
media-thickness	integer(1:MAX)	MAY	MAY

The "media-col" collection member attributes definitions are:

8.4.1 media-tooth (type3 keyword | name(MAX))

The "media-tooth" member attribute indicates the desired tooth (or roughness) of the media being specified. The source for this attribute is DPA.

The tooth of a medium is particularly important for those marking engines that use pens (e.g. plotters) to mark the medium. The type of pen used shall match the tooth of the medium for best resolution.

Standard keyword values for "media-tooth" are:

Keyword	Description
'fine'	The specified media should have a fine tooth or smooth finish.
'medium'	The specified media should have a medium tooth or regular finish.
'coarse'	The specified media should have a coarse tooth or rough finish.

The Administrator can define a custom media tooth using the 'name' (MAX) attribute syntax of the "media-tooth-supported" (1setOf (type3 keyword | name(MAX))) Printer attribute. Note: as with other Job Template and member attributes, the user can also supply user-defined tooth names that

are not among the values of the “media-tooth-supported” Printer attribute, if the Administrator has configured the Printer's "user-defined-values-supported" attribute (see [prod-print]) to contain the 'media-tooth' attribute keyword value.

8.4.1.1 media-tooth-supported (1setOf (type3 keyword | name(MAX)))

The "media-tooth-supported" (1setOf (type3 keyword | name(MAX))) Printer attribute identifies the values of this "media-tooth" member attribute that the Printer supports, i.e., the media tooth supported.

8.4.2 media-grain (type3 keyword | name(MAX))

The "media-grain" member attribute indicates the desired grain of the media being specified. The source for this attribute is DPA.

Grain affects the curl and the folding of the medium. Some marking engines are sensitive to the resulting curl.

Standard keyword values for "media-grain" are:

Keyword	Description
'x-direction'	The direction of the paper fibers is in the short dimension (i.e. the x-direction) of the medium.
'y-direction'	The direction of the paper fibers is in the long dimension (i.e. the y-direction) of the medium.

See section 2.4 of [prod-print] regarding the coordinate system.

The Administrator can define a custom media grain using the 'name' (MAX) attribute syntax of the "media-grain-supported" (1setOf (type3 keyword | name(MAX))) Printer attribute. Note: as with other Job Template and member attributes, the user can also supply user-defined grain names that are not among the values of the “media-grain-supported” Printer attribute, if the Administrator has configured the Printer's "user-defined-values-supported" attribute (see [prod-print]) to contain the 'media-grain' attribute keyword value.

8.4.2.1 media-grain-supported (1setOf (type3 keyword | name(MAX)))

The "media-grain-supported" (1setOf (type3 keyword | name(MAX))) Printer attribute identifies the values of this "media-grain" member attribute that the Printer supports, i.e., the media grain supported.

8.4.3 media-material (type3 keyword | name(MAX))

The "media-material" member attribute indicates the desired material of the media being specified. The source for this attribute is JDF v1.0.

Standard keyword values for "media-material" are:

Keyword	Description
'aluminum'	The specified media should be aluminum.
'dry-film'	The specified media should be dry film.
'paper'	The specified media should be paper.
'polyester'	The specified media should be polyester.
'wet-film'	The specified media should be wet film.

Note: The standard keyword values for the "media-material" attribute are the JDF standard values .

The Administrator can define custom media materials using the 'name' (MAX) attribute syntax of the "media-material-supported" (1setOf (type3 keyword | name(MAX))) Printer attribute. Note: as with other Job Template and member attributes, the user can also supply user-defined material names that are not among the values of the “media-material-supported” Printer attribute, if the Administrator has configured the Printer's "user-defined-values-supported" attribute (see [prod-print]) to contain the 'media-material' attribute keyword value.

8.4.3.1 media-material-supported (1setOf (type3 keyword | name(MAX)))

The "media-material-supported" (1setOf (type3 keyword | name(MAX))) Printer attribute identifies the values of this "media-material" member attribute that the Printer supports, i.e., the media materials supported.

8.4.4 media-thickness (integer(1:MAX))

The "media-thickness" member attribute indicates the thickness of the media being specified. The source for this attribute is JDF v1.0.

The unit of measure for the "media-thickness" member attribute is one hundredth of a millimeter. This unit is equivalent to 1/2540 th of an inch resolution.

8.4.4.1 media-thickness-supported (rangeOfInteger(1:MAX))

The "media-thickness-supported" (rangeOfInteger(1:MAX)) Printer attribute identifies the values of this "media-thickness" member attribute that the Printer supports.

9 Conformance Requirements

This section summarizes the Conformance Requirements detailed in the definitions in this document for clients and Printer objects (servers or devices).

9.1 Conformance Requirements for Printer objects

In general each of the attributes defined in this document are OPTIONAL for a Printer to support, so that Printer implementers MAY implement any combination of attributes. Only the following conditional conformance requirements are defined:

If the Printer supports:	then the Printer MUST also support (but vice-versa is OPTIONAL):
"job-hold-until-time"	"job-hold-until" (see [RFC2911] section 4.2.2)

Each of the collection attribute definitions indicate which member attributes are REQUIRED and which are OPTIONAL for a Printer to support and is not repeated here.

If a Printer supports the 'collection' attribute syntax of a Job Template attribute , then it MUST support the distinguished none value defined for that collection. See section 2.7 of [prod-print].

Support of the 'name' attribute syntax for Job Template attributes and collection member attributes is OPTIONAL, as in IPP/1.1 [RFC2911].

9.2 Conformance Requirements for clients

Clients that support two Job Template attributes that control the same aspect, such as "job-hold-until" and "job-hold-until-time", MUST NOT supply both in a Job Creation request as indicated in the definitions of these attributes.

Clients that support a "xxx" collection Job Template attribute SHOULD use the Get-Printer-Attributes request to obtain the "xxx-default" collection and display that to the user, so that the user can make any changes before submitting the Job. Then the client submits values for all member attributes, rather than depending on the Printer's defaulting for omitted member attributes, since such defaulting is implementation dependent and will vary from Printer to Printer.

9.3 Conformance Requirements for the Job Save and Reprint Capability

This section specifies the conformance requirements for Printers and clients pertaining to the Job Save and Reprint Capability. It also defines the conformance requirements for Job Save Format specifications.

9.3.1.1 Client Conformance Requirements for Job Save and Reprint Capability

Conforming clients:

1. MUST support the “job-save-disposition” (collection) attribute according to the description in section 5.7, including the member attributes as required by Table 2.

2. MUST support the “save-info” (collection) member attribute according to the "job-save-disposition" attribute description in section 5.7, including the member attributes as required by Table 3.

3. MUST support the Print-URI operation for printing saved jobs according to section 3.3.1.1 above.

9.3.1.2 Printer Conformance Requirements for Job Save and Reprint Capability

Conforming Printers:

1. MUST support the “job-save-disposition” (collection) Job Template attribute according to the description in section 5.7, including the member attributes as required by Table 2 and the “job-save-disposition-default” attribute and the “job-save-disposition-supported” attribute.

2. MUST support all of the “save-disposition” member attribute values defined in the "job-save-disposition" description section 5.7.

3. MUST support the “save-info” (collection) member attribute according to the "job-save-disposition" attribute description in section 5.7, including the member attributes as required by Table 3 and the “save-location-supported”, “save-name-supported”, “save-document-format-supported”, and “save-info-supported” Printer attributes.

4. SHOULD support both the ‘file:’ and ‘ftp:’ schemes for use in the “save-location” member attribute as defined in the "job-save-disposition" attribute description in section 5.7.

5. MUST support at least one “save-document-format” (mimeMediaType) attribute value, as defined in the "job-save-disposition" attribute description in section 5.7, for use in both the “save-document-format” member attribute and “document-format” operation attribute in the Print-URI operation as described in section 3.3.1.1.

6. SHOULD support a superset of the values of the Printer’s “save-document-format-supported” attribute as the values of the Printer’s “document-format-supported” attribute.

7. MUST support the ‘job-saving’ and ‘job-save-error’ values of the “job-state-reasons” Job Description attribute as defined in the "job-state-reasons" attribute description in section 8.3.

8. MUST support the Print-URI operation that references saved jobs as defined in section 3.3.1.1.

9.3.1.3 Job Save Format Specification Conformance Requirements

Documents that define a Job Save format MUST contain the following information:

1. MUST specify the mimeMediaType value that identifies the Job Save Format, e.g., 'application/X-single-page-tiff'.

2. MUST define the representation for PDL data.

3. MUST specify the formats of the document content and how the Job Save Format file references the document content, e.g., using Relative URI value (see [RFC2396]) to reference the document content in the “document-uri” operation attribute.

4. MUST define whether or not Printers with differing values of their “printer-make-and-model” Printer attributes are expected to print saved jobs with the same appearance, i.e., whether or not a Printer MUST omit the “job-printer-make-and-model” Job Description attribute when saving the job in the defined format (see the "job-printer-make-and-model" attribute description in section 6.1).

10 IANA Considerations

10.1 Attribute Registration

The attributes defined in this document will be published by IANA according to the procedures in RFC 2911 [RFC2911] section 6.2 in the following file:

```
http://www.iana.org/assignments/ipp-registrations
```

The registry entry will contain the following information:

```
Job Template attributes:                                    Section:
feed-orientation  (type3 keyword)                               5.1
font-name-requested  (name(MAX))                                5.2
font-size-requested  (integer (1:MAX))                          5.3
job-hold-until-time  (dateTime)                                 5.4
job-phone-number  (text(127))                                   5.5
job-recipient-name  (name(MAX))                                 5.6
job-save-disposition  (collection)                              5.7
     ISSUE 1: We need to register all member attributes. TBD.
pdl-init-file  (1setOf collection)                              5.8
     ISSUE 2: We need to register all member attributes. TBD.
proof-print  (collection)                                       5.9
     ISSUE 3: We need to register all member attributes. TBD.

Operation attributes:                                       Section:
job-password  (octetString(255))                                4.1
job-password-encryption  (type3 keyword | name(MAX))            4.2
```



```
Job Description attributes:                                    Section:
job-printer-make-and-model  (text(127))                             6.1

Printer Description attributes:                                Section:
job-creation-attributes-supported  (1setOf type2 keyword)           7.1
job-password-supported  (integer(0:127))                            7.2
job-password-encryption-supported  (1setOf (type3 keyword |
        name(MAX)))                                                 7.3
job-spooling-supported  (type2 keyword)                             7.4
max-save-info-supported  (integer(1:MAX))                           7.5
media-col-database  (1setOf collection)                             7.6
printer-detailed-status-messages (1setOf text(MAX))                 7.7
which-jobs-supported  (1setOf type2 keyword)                        7.8
```

10.2 Attribute Value Registration

The “document-format” mimeMediaType attribute values, the “pdl-override-supported” type2, “which-jobs” type2, and “job-state-reasons” type2 keyword attribute values defined in this document will be published by IANA according to the procedures in RFC 2911 [RFC2911] section 6.1 in the following file:

```
     http://www.iana.org/assignments/ipp-registrations
```

The registry entries will contain the following information:

```
ISSUE 4:  We need to register all keyword values of all NEW
attributes in this spec, including the new “media-col” member
attributes.  TBD.

pdl-override-supported (type2 keyword):                      Section:
  guaranteed                                                      8.1

which-jobs (type2 keyword):      Section:
  pending                                                         8.2
  pending-held                                                    8.2
  processing                                                      8.2
  processing-stopped                                              8.2
  canceled                                                        8.2
  aborted                                                         8.2
  all                                                             8.2

job-state-reasons (type2 keyword):  Section:
  job-saved-successfully                                          8.3
  resources-are-not-supported                                     8.3
  job-warnings-detected                                           8.3
  job-saving                                                      8.3
  job-save-error                                                  8.3
  job-spooling                                                    8.3
  job-streaming                                                   8.3
  job-scheduling                                                  8.3
```


job-suspending	8.3
job-suspended	8.3
job-suspended-by-operator	8.3
job-suspended-by-user	8.3
job-suspended-by-system	8.3
job-resuming	8.3
job-password-wait	8.3
proof-print-wait	8.3

11 Internationalization Considerations

The IPP extensions defined in this document require the same internationalization considerations as any of the Job Template attributes defined in IPP/1.1 [RFC2911].

12 Security Considerations

The IPP extensions defined in this document require the same security considerations as any of the Job Template attributes defined in IPP/1.1 [RFC2911].

13 Normative References

This section lists references to documents whose implementation are required in order to conform to this specification.

[coll]
deBry, R., , Hastings, T., Herriot, R., Ocke, K., and P. Zehler, "Internet Printing Protocol (IPP): The ‘collection’ attribute syntax", <draft-ietf-ipp-collection-05.txt>, work in progress, July 17, 2001.

[prod-print]
Ocke, K., Hastings, T., "Internet Printing Protocol (IPP): Production Printing Attributes - Set1", IEEE-ISTO 5100.3-2001, February 12, 2001, ftp://ftp.pwg.org/pub/pwg/standards/pwg5100.3.pdf.

[RFC2119]
S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March 1997

[RFC2910]
Herriot, R., Butler, S., Moore, P., Turner, R., and J. Wenn, "Internet Printing Protocol/1.1: Encoding and Transport", RFC 2910, September 2000.

[RFC2911]
Hastings, T., Herriot, R., deBry, R., Isaacson, S., and P. Powell, "Internet Printing Protocol/1.1: Model and Semantics", RFC 2911, September 2000.

14 Informative References

This section lists references to documents that provide information, but whose implementation is not required in order to conform to this specification.

[admin-ops]
Kugler, C, Hastings, T., Lewis, H., "Internet Printing Protocol (IPP): Job and Printer Administrative Operations", <draft-ietf-ipp-ops-set2-03.txt>, July 17, 2001.

[override]
Herriot, R., Ocke, K., "Internet Printing Protocol (IPP): Override Attributes for Documents and Pages", IEEE-ISTO 5100.4-2001, February 7, 2001, ftp://ftp.pwg.org/pub/pwg/standards/pwg5100.4.pdf.

[RFC2565]
Herriot, R., Butler, S., Moore, P., and R. Turner, "Internet Printing Protocol/1.0: Encoding and Transport", RFC 2565, April 1999.

[RFC2566]
deBry, R., , Hastings, T., Herriot, R., Isaacson, S., Powell, P., "Internet Printing Protocol/1.0: Model and Semantics", RFC 2566, April 1999.

[set-ops]
Hastings, T., Herriot, R., Kugler, C., Lewis, H., "Internet Printing Protocol (IPP): Job and Printer Set Operations", August 28, 2001.

15 Author's Addresses

Tom Hastings
Xerox Corporation
701 South Aviation Blvd, ESAE-210
El Segundo, CA 90245

Phone: 310-333-6413
Fax: 310-333-5514
e-mail: hastings@cp10.es.xerox.com

Don Fullman
Xerox Corporation
701 South Aviation Blvd, ESAE-210
El Segundo, CA 90245

Phone: 310-333-8342
Fax: 310-333-5514
e-mail: fullman@cp10.es.xerox.com

IPP Web Page: http://www.pwg.org/ipp/
IPP Mailing List: ipp@pwg.org

To subscribe to the ipp mailing list, send the following email:

1) send it to majordomo@pwg.org
2) leave the subject line blank
3) put the following two lines in the message body:

subscribe ipp
end

Implementers of this specification document are encouraged to join IPP Mailing List in order to participate in any discussions of clarification issues and review of registration proposals for additional attributes and values.

Other Participants:

Ron Bergman - Hitachi Koki Imaging Systems
Dan Calle - Digital Paper
Weihai Chen - Microsoft
Lee Farrell - Canon Information Systems
Satoshi Fujitani - Ricoh
Roelof Hamberg - Océ
Tom Hastings - Xerox
Bob Herriot - Xerox
David Kellerman - Northlake Software
Carl Kugler - IBM
Harry Lewis - IBM
Carl-Uno Manros - Xerox
Satoshi Matsushita - Brother
Ira McDonald - High North Inc.
Paul Moore - Netreon
Hugo Parra, Novell
Stuart Rowley - Kyocera
Gail Songer - Netreon
Geoff Sorod - Software 2000
Jerry Thrasher - Lexmark
Shinichi Tsuruyama - Epson
Atsushi Uchino - Epson
Shigeru Ueda - Canon
William Wagner - NetSilicon/DPI
Mark Vander Wiele - IBM
Don Wright - Lexmark
Michael Wu - Heidelberg Digital
Peter Zehler - Xerox

16 Appendix B: Summary of other IPP documents

The full set of IPP documents includes:

1. Design Goals for an Internet Printing Protocol [RFC2567]
2. Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]

3. Internet Printing Protocol/1.1: Model and Semantics (this document)
4. Internet Printing Protocol/1.1: Encoding and Transport [RFC2910]
5. Internet Printing Protocol/1.1: Implementer's Guide [IIG]
6. Mapping between LPD and IPP Protocols [RFC2569]

The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed printing functionality, and it enumerates real-life scenarios that help to clarify the features that need to be included in a printing protocol for the Internet. It identifies requirements for three types of users: end users, operators, and administrators. It calls out a subset of end user requirements that are satisfied in IPP/1.0. A few OPTIONAL operator operations have been added to IPP/1.1.

The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document describes IPP from a high level view, defines a roadmap for the various documents that form the suite of IPP specification documents, and gives background and rationale for the IETF working group's major decisions.

The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the abstract operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It defines the encoding rules for a new Internet MIME media type called "application/ipp". This document also defines the rules for transporting over HTTP a message body whose Content-Type is "application/ipp". This document defines a new scheme named 'ipp' for identifying IPP printers and jobs.

The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some of the considerations that may assist them in the design of their client and/or IPP object implementations. For example, a typical order of processing requests is given, including error checking. Motivation for some of the specification decisions is also included.

The "Mapping between LPD and IPP Protocols" document gives some advice to implementers of gateways between IPP and LPD (Line Printer Daemon) implementations.

17 Appendix C: Description of the IEEE Industry Standards and Technology (ISTO)

The IEEE-ISTO is a not-for-profit corporation offering industry groups an innovative and flexible operational forum and support services. The IEEE-ISTO provides a forum not only to develop standards, but also to facilitate activities that support the implementation and acceptance of standards in the marketplace. The organization is affiliated with the IEEE (http://www.ieee.org/) and the IEEE Standards Association (http://standards.ieee.org/).

For additional information regarding the IEEE-ISTO and its industry programs visit:

http://www.ieee-isto.org.

18 Appendix D: Description of the IEEE-ISTO PWG

The Printer Working Group (or PWG) is a Program of the IEEE Industry Standards and Technology Organization (ISTO) with member organizations including printer manufacturers, print server developers, operating system providers, network operating systems providers, network connectivity vendors, and print management application developers. The group is chartered to make printers and the applications and operating systems supporting them work together better. All references to the PWG in this document implicitly mean "The Printer Working Group, a Program of the IEEE ISTO." In order to meet this objective, the PWG will document the results of their work as open standards that define print related protocols, interfaces, procedures and conventions. Printer manufacturers and vendors of printer related software will benefit from the interoperability provided by voluntary conformance to these standards.

In general, a PWG standard is a specification that is stable, well understood, and is technically competent, has multiple, independent and interoperable implementations with substantial operational experience, and enjoys significant public support.

For additional information regarding the Printer Working Group visit:

http://www.pwg.org

