

# ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

---

## STANDARD ECMA-140

# DOCUMENT PRINTING APPLICATION (DPA)

---

Part 1 - Abstract-Service Definition and Procedures

Part 2 - Protocol Specification

June 1990

Free copies of this document are available from ECMA,  
European Computer Manufacturers Association  
114 Rue du Rhône - CH-1204 Geneva (Switzerland)

Phone: + 41 22 735 36 34 Fax: + 41 22 786 52 31

# ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

---

## STANDARD ECMA-140

# DOCUMENT PRINTING APPLICATION (DPA)

---

Part 1 - Abstract-Service Definition and Procedures

Part 2 - Protocol Specification

June 1990

## BRIEF HISTORY

The Document Printing Application is one component of a coordinated set of facilities and standards needed to satisfy the printing requirements of the modern distributed office. Together, the capabilities provided can enable users to create and produce high-quality office documents in a consistent and unambiguous manner within a distributed open systems environment.

Specifically, this Standard addresses those aspects of document processing that enable users in a distributed open systems environment to send electronic documents to shared, possibly geographically-dispersed printers, and to cause the documents to be printed in accordance with their desires. For the purposes of this Standard, it is assumed that such documents have been composed in a form that is compatible with the destination printing system prior to their introduction to the Document Printing Application.

This ECMA Standard has been developed by ECMA/TC32 in close cooperation with ECMA/TC29 and ISO/IEC JTC1/SC18. A similar standard is under development in SC18.

Adopted as an ECMA Standard by the General Assembly of 14th December 1989.

## Table of Contents

### Part I - Abstract Service Definition and Procedures

#### SECTION ONE - INTRODUCTION

	Page
1. SCOPE AND FIELD OF APPLICATION	5
1.1 Scope	5
1.2 Field of Application	5
2. REFERENCES	6
3. DEFINITIONS	7
3.1 Document	7
3.1.1 Final form document	7
3.1.2 Revisable (form) document	7
3.1.3 SPDL document	7
3.2 Print format	7
3.3 Page description language	8
3.4 Creator	8
3.5 Printing	8
3.6 Physical printer	8
3.7 Print server	8
3.8 Print client	8
3.9 Print access protocol	8
3.10 Print service	8
3.11 Print system	8
3.12 Logical printer	8
3.13 Print request	8
3.14 Service request	9
3.15 Print job	9
3.16 Print operation parameters	9
3.16.1 Print management instructions	9
3.16.2 Document production instructions	9
3.17 Imposition	10
3.18 Signature	10

3.19	System administrator	10
3.20	Assured reproduction area	10
4.	ACRONYMS	10
5.	CONVENTIONS	10
5.1	Conventions for Abstract-services	10
5.2	Conventions for Documentation in General	11
<b>SECTION TWO - DP ABSTRACT-SERVICE DEFINITION</b>		
6.	DP ABSTRACT MODEL	15
6.1.	Objects in the Document Printing Environment	16
6.1.1	DP-Server Object	16
6.1.2	DP-User Object	16
6.1.3	DP-Administrator Object	16
6.2	Ports Used in the Document Print Service Model	17
6.2.1	DP-User Port	17
6.2.2	DP-Administration Port	17
6.3	DPA Object Classes	17
6.3.1	Job object class	18
6.3.2	Printer object class	18
6.3.3	Server object class	19
6.3.4	Media object class	19
6.3.5	Font resource class	19
6.3.6	Named colour object class	20
6.3.7	Transfer method object class	20
6.3.8	Delivery method object class	20
6.3.9	Job sheet object class	20
6.3.10	Finishing method object class	20
6.3.11	Output method object class	20
6.3.12	Imposition function object class	20
6.3.13	Scheduler object class	21
6.4	Attributes	21
6.4.1	Introduction	21
6.4.2	Attribute-Type	22
6.4.3	Attribute-values	22
6.4.4	Attribute-type definition	22
6.4.5	Filters	25
6.4.6	Qualified Attribute	27
6.5	Document Processing Model	27
6.5.1	Objects in the DP-User Environment	28
6.5.2	Objects in the DP-Server Environment	29

6.6	Security in DPA	29
6.6.1	Authentication	30
6.6.2	Access authorization	30
7.	ABSTRACT ASSOCIATION INFORMATION	31
7.1	Abstract-bind-operation	31
7.1.1	Bind-argument Parameters	31
7.1.2	Bind-result Parameters	32
7.1.3	Bind-error parameters	32
7.2	Abstract-unbind operation	33
8.	ABSTRACT OPERATIONS	33
8.1	Common data-types and values used in DPA abstract operations	33
8.1.1	Imported data-types	33
8.1.2	Data-types common for most DPA abstract operations	34
8.2	User Port Abstract Operations	35
8.2.1	Print	35
8.2.2	Modify-job	40
8.2.3	Cancel-job	42
8.2.4	List-object-attributes	43
8.3	DP-administration Port Operations Definitions	45
8.3.1	Promote-job	45
8.3.2	Interrupt	45
8.3.3	Pause	46
8.3.4	Resume	48
8.4	Abstract-Errors	49
8.4.1	Access-error	49
8.4.2	Attribute-error	49
8.4.3	Document-access-error	51
8.4.4	Name-error	51
8.4.5	Printer-error	52
8.4.6	Security-error	52
8.4.7	Selection-error	53
8.4.8	Service-error	53
8.4.9	Update-error	54
8.4.10	Error Precedence	54
8.5	Abstract-Events	55
8.5.1	Abort-event	55
8.5.2	Error-event	55
8.5.3	Warning-event	56
8.5.4	Report-event	56
8.5.5	Other-event	57
8.5.6	Event Identifiers	57

## SECTION THREE - PRINT SERVICE ATTRIBUTES

9.	PRINT SERVICE ATTRIBUTES	61
9.1.	Overview of the Attribute Concept	61
9.2	Generic Object attribute syntaxes	61
9.2.1	Identifier	61
9.2.2	Descriptor	61
9.2.3	Name	62
9.2.4	Aliases	62
9.2.5	State	62
9.2.6	Message	62
9.2.7	Availability	63
9.2.8	List of managers	63
9.2.9	Test	63
9.3	Job Object attributes	64
9.3.1	Job-information	64
9.3.2	Job-results-handling	65
9.3.3	Job-event-handling	66
9.3.4	Job-scheduling-instructions	68
9.3.5	Printer-identification	71
9.3.6	Production-instructions	71
9.3.7	Document-description	78
9.3.8	Document-attributes	80
9.3.9	Access-and-accounting	86
9.3.10	Job-states	87
9.3.11	Job-security-handling	93
9.4	Printer Object attributes	94
9.4.1	Printer-identifier	94
9.4.2	Printer-name	94
9.4.3	Printer-location	94
9.4.4	Printer-model	94
9.4.5	Printer-type	94
9.4.6	Printer-security	95
9.4.7	Printer-speed	96
9.4.8	Printer-resolution	96
9.4.9	Printer-state	96
9.4.10	Printer-message	97
9.4.11	Attributes-supported-by-printer	97
9.4.12	Object-classes-supported	97
9.4.13	Printers-supported	97
9.4.14	Printers-active	97
9.4.15	Fonts-supported	98
9.4.16	Fonts-active	98
9.4.17	Font-default	98
9.4.18	Media-supported	98
9.4.19	Media-active	98
9.4.20	Medium-default	99
9.4.21	Named-colours-supported	99



9.4.22	Named-colours-active	99
9.4.23	Named-colour-default	99
9.4.24	Finishing-methods-supported	99
9.4.25	Finishing-methods-active	99
9.4.26	Finishing-methods-default	100
9.4.27	Output-methods-supported	100
9.4.28	Output-methods-active	100
9.4.29	Output-methods-default	100
9.4.30	Output-bias-supported	100
9.4.31	Output-bias-active	101
9.4.32	Output-bias-default	101
9.4.33	Imposition-functions-supported	101
9.4.34	Imposition-functions-enabled	101
9.4.35	Imposition-functions-default	101
9.4.36	Print-qualities-supported	102
9.4.37	Print-qualities-active	102
9.4.38	Print-quality-default	102
9.4.39	Job-sheets-supported	102
9.4.40	Job-sheets-active	102
9.4.41	Job-sheet-defaults	103
9.4.42	Character-sets-supported	103
9.4.43	Character-sets-active	103
9.4.44	Character-set-default	103
9.4.45	Character-mappings-supported	104
9.4.46	Character-mappings-active	104
9.4.47	Character-mapping-default	104
9.4.48	Character-repertoires-supported	104
9.4.49	Character-repertoires-active	104
9.4.50	Character-repertoire-default	105
9.4.51	Start-message-supported	105
9.4.52	End-message-supported	105
9.4.53	Sides-supported	105
9.4.54	Sides-default	105
9.4.55	Page-select-supported	106
9.4.56	Maximum-copies-supported	106
9.4.57	Content-formats-supported	106
9.4.58	Content-format-default	106
9.4.59	Document-formats-supported	107
9.4.60	Document-format-default	107
9.4.61	Araz-supported	107
9.4.62	Ara-default	107
9.4.63	Page-sizes-supported	107
9.4.64	Orientations-supported	108
9.4.65	Orientation-default	108
9.4.66	Page-order-assumed	108
9.4.67	Page-independence-assumed	108
9.4.68	Associated-server	108
9.4.69	Sheet-count	109
9.4.70	Printer-needs-attention-time	109
9.4.71	Printer-needs-key-operator-attention-time	109

9.5	Server Object attributes	109
9.5.1	Server-identifier	109
9.5.2	Server-name	109
9.5.3	Server-address	110
9.5.4	Server-state	110
9.5.5	Server-message	110
9.5.6	Object-classes-supported	110
9.5.7	Attributes-supported-by-server	110
9.5.8	Physical-printers-supported	111
9.5.9	Physical-printers-active	111
9.5.10	Logical-printers-supported	111
9.5.11	Logical-printers-active	111
9.5.12	Translators-supported	112
9.5.13	Translators-active	112
9.5.14	Delivery-methods-supported	112
9.5.15	Delivery-methods-active	112
9.5.16	Scheduling-algorithms-supported	112
9.5.17	Scheduling-algorithm-active	112
9.5.18	System-hints-supported	112
9.6	Media Object attributes	113
9.6.1	Medium-identifier	115
9.6.2	Medium-state	115
9.6.3	Medium-message	116
9.6.4	Medium-type	116
9.6.5	Medium-size	116
9.6.6	Medium-Dimensions	116
9.6.7	Medium-colour	116
9.6.8	Medium-tooth	117
9.6.9	Medium-grain	117
9.6.10	Medium-sides	117
9.6.11	Medium-weight	117
9.6.12	Medium-assured-reproduction-area	117
9.6.13	Medium-associated-media	117
9.6.14	Medium-form-parts	118
9.7	Font Resource Object attributes	118
9.7.1	Font-identifier	118
9.7.2	Font-type	118
9.7.3	Font-state	118
9.7.4	Font-message	119
9.8	Named-colour Object attributes	119
9.8.1	Named-colour-identifier	119
9.8.2	Named-colour-state	119
9.8.3	Named-colour-message	119
9.9	Transfer-method Object attributes	119
9.9.1	Transfer-method-identifier	120
9.9.2	Document-reference-type	120
9.9.3	Transfer-method-message	120

9.10	Delivery Method Object attributes	121
9.10.1	Delivery-method-identifier	122
9.10.2	Delivery-method-message	123
9.11	Job Sheet Object attributes	123
9.11.1	Job-sheet-identifier	123
9.11.2	Job-sheet-message	124
9.12	Finishing Method Object attributes	124
9.12.1	Finishing-method-identifier	125
9.12.2	Finishing-method-message	126
9.13	Output Method Object attributes	126
9.13.1	Output-method-identifier	127
9.13.2	Output-method-message	128
9.14	Imposition-Function Object attributes	128
9.14.1	Imposition-function-identifier	129
9.14.2	Imposition-function-message	130
9.15	Scheduler Object attributes	130
9.15.1	Scheduler-identifier	130
9.15.2	Scheduler-message	130
<b>SECTION FOUR - DPA REALIZATION</b>		
10.	OVERVIEW	133
11.	SUPPLY OF THE DP-USER ABSTRACT-SERVICE	133
11.1	Supply of the DP-user Port Abstract-services	133
11.1.1	Performance of the Print Abstract-operation	133
11.1.2	Performance of the ModifyJob Abstract-operation	133
11.1.3	Performance of the CancelJob Abstract-operation	134
11.1.4	Performance of the ListObjectAttributes Abstract-operation	134
12.	SUPPLY OF THE DP-ADMINISTRATOR ABSTRACT-SERVICE	134
12.1	Supply of the DP-administration Port Abstract-services	134
12.1.1	Performance of the PromoteJob Abstract-operation	135
12.1.2	Performance of the Interrupt Abstract-operation	135
12.1.3	Performance of the Pause Abstract-operation	135
12.1.4	Performance of the Resume Abstract-operation	135
13.	PORT REALIZATION	136
<b>APPENDICES PART 1</b>		
APPENDIX A - FORMAL ASSIGNMENT OF DPA STANDARD OBJECT IDENTIFIERS		143
APPENDIX B - FORMAL DEFINITION OF THE DPA ABSTRACT SERVICE		57

APPENDIX C - FORMAL DEFINITION OF DPA STANDARD ATTRIBUTES	175
APPENDIX D - FORMAL DEFINITION OF DPA STANDARD PARAMETER UPPER BOUND	205

## Part 2 - Protocol Specification

### SECTION ONE - INTRODUCTION

1. SCOPE AND FIELD OF APPLICATION	211
1.1 Scope	211
1.2 Field of Application	211
2. REFERENCES	212

### SECTION TWO - DPA ACCESS PROTOCOL SPECIFICATION

3. OVERVIEW OF THE PROTOCOL	215
3.1 DPA Access Protocol Model	215
3.2 Services Provided by the DPA Access Protocol	216
3.3 Use of Underlying Services	216
3.3.1 Use of ROSE Services	216
3.3.2 Use of RTSE Services	217
3.3.3 Use of ACSE Services	217
4. DPA ACCESS PROTOCOL ABSTRACT SYNTAX DEFINITION	217
5. CONFORMANCE	220
5.1 Statement Requirements	220
5.2 Static Requirements	220
5.3 Dynamic Requirements	221

### APPENDIX PART 2

APPENDIX A - FORMAL ASSIGNMENT OF PROTOCOL OBJECT IDENTIFIERS	225
---	-----

---

**Part I**  
**Abstract Service Definition Procedures**

---

1977

1977



SECTION ONE - INTRODUCTION





## 1. SCOPE AND FIELD OF APPLICATION

### 1.1 Scope

This Standard ECMA-140 consists of two parts:

Part 1: Document Printing Application - Abstract-service Definitions and Procedures

Part 2: Document Printing Application - Protocol Specification

The Document Printing Application is one component of a coordinated set of facilities and standards needed to satisfy the printing requirements of the modern distributed office. Together, the capabilities provided can enable users to create and produce high-quality office documents in a consistent and unambiguous manner within a distributed open systems environment.

Specifically, this Standard addresses those aspects of document processing that enable users in a distributed open systems environment to send electronic documents to shared, possibly geographically-dispersed printers, and to cause the documents to be printed in accordance with their desires. For the purposes of this Standard, it is assumed that such documents have been composed in a form that is compatible with the destination printing system prior to their introduction to the Document Printing Application.

Other Standards deal with related aspects of document processing, such as the creation and formatting of electronic documents, and the underlying protocols used to transport electronic documents to a printing system. This Standard is aligned with these related Standards as appropriate, and shares some information in common with them. Clause 2 identifies those standards that are directly applicable to this one.

The Document Printing Application defined in this Standard is consistent with the model, architectural framework and design principles of the Distributed Office Application Model (ISO DP 10031-1). This Document Printing Application Standard defines services and specifies access protocols available within the application layer of the Reference Model (ISO 7498).

### 1.2 Field of Application

The Document Printing Application constitutes the final phase of the document processing cycle, i.e., the queuing, preparation, rendering and finishing of the fully composed form of the document on marking engines and other image generation devices. This cycle includes other processes such as document creation and interchange through public and private networks.

This Standard is oriented toward satisfying the following subset of the overall document processing functional requirements:

- an ability for multiple users to share access to distributed printers;
- an ability for users to convey information to a printing system to influence the scheduling and processing requirements of a print job;
- a capability for users to monitor and manage the progress of their print jobs;
- a capability for printing systems, and associated facilities, to protect against unauthorized printing of documents.

Many different document print formats have been developed, and are in wide use. For this reason, the Document Printing Application has been developed with a view toward supporting arbitrary print formats in a transparent manner. That is, the specific content or format of an electronic document is independent of the access protocol defined by the Document Printing Application Standard. The only requirement is that the destination printing system be capable of

dealing with the format of the transmitted document, and possess the features and functionality needed to successfully render the document.

However, in spite of this generality of focus, this Standard is particularly oriented toward providing the features needed to assist in the transport and faithful rendering of documents formatted in the Standard Page Description Language (SPDL - ISO/IEC 10180).

The access protocol defined by this standard enables a user to convey document files to a document print server, along with the parameters needed to express the user's desires regarding the scheduling and production of the ensuing print job. In addition, the protocol permits a user to inquire about the status, capabilities and characteristics of a document print server in order to choose from a variety of printing devices, depending on capabilities, formats, logistic convenience, cost, ownership and availability.

The protocol also allows users to inquire about jobs, modify the characteristics and progress of jobs, and obtain feedback about a job.

## 2. REFERENCES

- |                      |   |
|----------------------|---|
| ISO 7498             | Information Processing Systems - Open Systems Interconnection - Part 1: Basic Reference Model.  |
| ECMA-101             | Open Document Architecture (ODA) and Interchange Format - Part 1: Introduction and General Principles (ISO 8613-1) - Part 4: Document Profile (ISO 8613-4) - Part 5: Office Document Interchange Format (ODIF) (ISO 8613-5) |
| ECMA-137             | Document Filing and Retrieval (DFR)   |
| ISO 8649             | Information Processing Systems - Open Systems Interconnection - Service Definition for the Association Control Service Definition.  |
| ISO 8824             | Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).  |
| ISO 8824 DAD1        | Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1): Draft Addendum 1: ASN.1 Extensions.  |
| ISO 8571             | Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management - Concepts, virtual-filestore model, Services and Protocol.  |
| ISO/IEC 9066         | Information Processing - Text Communication - Reliable Transfer - Part 1: Model and Service Definition.   |
| ISO/IEC 9072         | Information Processing - Text Communication - Remote Operations - Part 1: Model, Notation and Service Definition.   |
| ISO/IEC 9541-1 (DIS) | Information Processing - Font Information Interchange - Part 1: Architecture.   |
| ISO/IEC 9541-2 (DIS) | Information Processing - Font Information Interchange - Part 2: Interchange Format.   |
| ISO/IEC 9594-2       | Information Processing Systems - The Directory - Models.  |
| ISO/IEC 9594-3       | Information Processing Systems - The Directory - Abstract Service Definition.   |

ISO/IEC 10021-3	Information Processing Systems - Text Communication - MOTIS - Part 3: Abstract Service Definition Conventions.
ISO/IEC 10021-4	Information Processing Systems - Text Communication - MOTIS - Part 4: Message Transfer System: Abstract Service Definition.
ISO/IEC 10031-1 (DIS)	Information Processing - Text Communication - Distributed Office Applications Model - Part 1: General Model
ISO/IEC 10031-2 (DIS)	Information Processing - Text Communication - Distributed Office Applications Model - Part 2: Referenced Data Transfer
ISO/IEC 10106-1 (DIS)	Information Processing - Text Communication - Document Filing and Retrieval (DFR) - Part 1: Abstract Service Definition and Procedures.
ISO/IEC 10180 (DP)	Information Processing - Text Communication - Standard Page Description Language (SPDL).

### 3. DEFINITIONS

The definitions of ISO/IEC 10031 apply to this Standard. In addition, for the purpose of this Standard, the following definitions apply:

#### 3.1 Document

In this Standard, when not otherwise qualified, a document is assumed to be the electronic form of a document, in a form acceptable to the print server(s) to which the document is (to be) sent. Qualified terms that may be used include:

##### 3.1.1 Final form document

An electronic document which is in a form suitable for printing by some destination print server(s); this term generally implies that the document is no longer in a form that may be edited under normal circumstances, but this is not required.

##### 3.1.2 Revisable (form) document

An electronic document which is in a form that may be edited by some properly equipped system; this term generally implies that the document is not in a final form for printing, but this is not required.

*Note 1*

There may be data formats that allow both forms to be present at the same time in a single document (e.g., ISO 8633). In this case these terms refer to the particular aspects of information being present in a document.

##### 3.1.3 SPDL document

A particular instance of a final form document, encoded in the Standard Page Description Language, and thus in a form suitable for printing by any print server(s) capable of rendering an SPDL document.

#### 3.2 Print format

The generic name for any data syntax with which a final form document may be encoded.

**3.3 Page description language**

A class of print format wherein the data syntax is based on the concepts of the modern programming languages, e.g., SPDL.

**3.4 Creator**

The software process that converts a revisable form document into a final form document, or adds final form information to the revisable form information already present in the document description.

**3.5 Printing**

Unless otherwise qualified or restricted, the act of printing may include any form of output which transforms the electronically encoded and stored form of a document into a visible representation. This includes, but is not limited to, hard copy printers, plotters, display devices, and microfilm/microfiche generators.

**3.6 Physical printer**

A physical device which is capable of rendering documents.

**3.7 Print server**

The entity that accepts print requests, together with accompanying documents (or references to them), from print clients and prints the documents in accordance with parameters delivered with the print request and/or incorporated within the electronic document.

**3.8 Print client**

The software entity that acts on behalf of a user to convey service requests to a print server and to deal appropriately with responses from the print server that arise from the submitted requests.

**3.9 Print access protocol**

The application-level protocol between any print client and any print server.

**3.10 Print service**

The functionality defined for the Document Printing Application, as visible between a (document) print-client and a (document) print server. For this particular application, the service is provided by the server.

**3.11 Print system**

An aggregation of one or more print servers and one or more printing devices, which may or may not be located in the same geographical area. The components of a print system are assumed to be interconnected in some manner, providing at least one network interface to print clients, and acting synergistically to supply the defined document print service.

**3.12 Logical printer**

A logical entity that permits a particular set of characteristics and capabilities (e.g., location, resolution, media available, etc.) to be grouped and referenced as a single entity. The logical printer may represent one or more printing devices having the same capabilities and characteristics; alternatively, a single printing device may be represented by more than one logical printer, where selection of a particular logical printer implies some specific form of processing to be applied to the document, or a particular set of resources that may be made available, e.g. media weights that may be (manually) loaded if the relevant logical printer is selected.

**3.13 Print request**

An operation invoked by a print client to cause a print server to print a document(s). Parameters are sent with the print request to identify the document, and to specify and control the act of

printing and producing the document(s), such as number of copies to print, whether to staple the hard copy document, etc. The document to be printed may accompany the print request, or the print request may include a reference (pointer) to the document, indicating that the print server is to fetch the document using some other protocol. Actual access and transport of the document to a print server may be accomplished by other protocols that are outside the scope of this Standard.

**3.14 Service request**

Within the print access protocol (standard), this term refers to a request for any type of operation implemented by a print server.

**3.15 Print job**

A class of data object created by a print server to collect information related to the progress of a particular print request. When a print server accepts a print request from a client, the print server generates a print job identifier that is unique within its environment, and returns it to the client to signal acceptance of the request, and to enable the client to interrogate status of the job subsequently.

**3.16 Print operation parameters**

Parameters sent to a print server to control or guide and assist the print server in producing the requested document in the form requested by the user. Print Operation parameters are defined by means of the attribute notation conventions presented in clause 6 of this standard. The set of print operation parameters are subdivided into two subsets based on how they may be conveyed to a print server. This classification scheme is depicted graphically in figure 1, and the terms are defined in the following subparagraphs.



Figure 1. Classification of print operation parameters

**3.16.1 Print management instructions**

Print operation parameters that may be communicated only within the print operation's arguments. These parameters are generally concerned with the management of print jobs or of the other objects related to the management of jobs. They do not directly affect the appearance of the printed documents.

**3.16.2 Document production instructions**

May be delivered with the print request or in the document, or both; if a specific instruction appears in both places for a given request and document, the print server will resolve the po-

tential conflict by means of an algorithm that is beyond the scope of this Standard. Different subsets of document production instructions may be defined for different specific print formats.

**3.17 Imposition**

Arrangement of page images on one side of a sheet so that when cut, folded and trimmed they will be presented in the proper orientation and in numerical sequence.

**3.18 Signature**

Sheet of paper printed on both sides and folded to make up part of a publication. For example, a sheet of paper with 2 printed pages on each side is folded once to form a 4-page signature. One with 4 pages on each side is folded twice to form an 8-page signature, and so on. The proper arrangement of pages in signatures is called imposition.

**3.19 System administrator**

A person with administrative responsibility for one or more components in a distributed system. Within this Standard, it is assumed that a system administrator has responsibility for a print server(s), and may be authorized to invoke some operations, or to access certain information, not available to the normal user.

*Note 2*

*The decision as to whether a system administrator will be accorded special privileges is a site and/or enterprise specific issue, not mandated by this Standard.*

**3.20 Assured reproduction area**

That area of a medium upon which a printing device is capable of rendering an image. The term may be applied either to the capability of a specific printer type to render an image on some arbitrary medium, or to the capacity of a specific medium to accept an image from some arbitrary printer, in a form acceptable to a user.

**4. ACRONYMS**

ARA	Assured reproduction area
ASN.1	Abstract Syntax Notation One
DOAM	Distributed Office Application Model
DFR	Document filing and retrieval
DPA	Document printing application
DPI	Document production instructions
DPS	Document print service
FTAM	File transfer and access method
PDL	Page description language
PMI	Print management instructions
POP	Print operation parameters
RDT	Referenced Data Transfer
SPDL	Standard Page Description Language

**5. CONVENTIONS**

**5.1 Conventions for Abstract-services**

This Part 1 of this ECMA Standard uses the following ASN.1-based descriptive conventions for the indicated purposes:

- 1) ASN.1 itself, to specify the abstract-syntax of information-objects and their components, common data-types, and state-variables.
- 2) The ASN.1 OBJECT and PORT macro and associated abstract-service definition conventions of ISO/IEC 10021-3, to specify the DPA Port.
- 3) The ASN.1 ABSTRACT-BIND, ABSTRACT-UNBIND, ABSTRACT-OPERATION, and ABSTRACT-ERROR macros of ISO/IEC 10021-3, to specify the DPA abstract-service.

*Note J*

*ASN.1 specifications in this Standard make full use of the ASN.1 DADI features, especially such symmetrical constructs as "WITH COMPONENTS" (sub-typing of sequences, sets and choices). All specifications are written using "IMPLICIT TAGS" convention, which means systematic omission, at the time of ASN.1 encoding, of all unnecessary "nested" tags, especially those "recovered" by context-specific ones.*

## 5.2 Conventions for Documentation in General

Throughout this document,

- terms are rendered in bold when defined, and in italic when referenced prior to their definitions;
- terms that are proper nouns, rather than generic terms, are capitalized;
- multi-word generic terms are hyphenated;
- labels are lower case; hyphens separate components;
- types have each component capitalized, but hyphens are not used.

The following characters are used in this Standard to indicate whether a parameter, an attribute, or other item described are mandatory, optional or conditional.

- M (Mandatory) indicates that an item shall be present in any case (must be supported by DPA);
- O (Optional) indicates that an item shall be present at the discretion of a DPA implementor;
- C (Conditional) indicates that an item shall be present under some circumstances defined by this Standard.





SECTION TWO - DP ABSTRACT-SERVICE DEFINITION



## 6. DP ABSTRACT MODEL

This clause provides an abstract functional model of Document Printing. For an introduction and description of the abstract-service concept and its definition conventions, see ISO/IEC 10021-3.

The Document Printing Application (DPA) enables a DP-User to send electronic documents to a DP-Server, and to request the printing of those documents. In addition, the DPA provides the means for a DP-User to make inquiries of the DP-Server, in order to determine, for example, the status of a previously-submitted print request, or the progress of the print job created to execute and manage the request, or to determine the status or capabilities of the DP-Server itself.

Additional facilities are included in the DPA for the purpose of managing a DP-Server. These server management facilities may be constrained by a particular DP-Server to limit their use to members of a certain class of users, referred to as DP-Administrators.

The DPA is described using an abstract model in order to define the services provided by the DPA as a whole -- the DP Abstract-Service. Figure 2 illustrates the DP model.

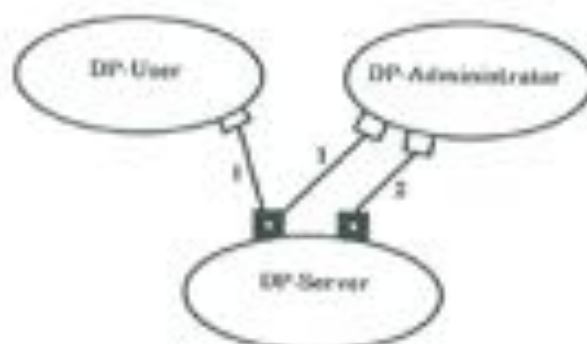


Figure 2. Document Printing Abstract Model

A DP-Server is modelled as an object whose overall behaviour can be described without reference to its internal structure. The services provided by the DP-Server object are made available at ports, where a type of port corresponds to a set of abstract-operations that can be accessed at the port (in this case, those abstract-operations which can be performed by the DP-Server object).

A DP-User and a DP-Administrator are each modelled as objects which obtain the services provided by a DP-Server through ports that are paired with DP-Server ports of the same respective types. A DP-User is associated with a single port type, the DP-User port; a DP-Administrator, on the other hand, may access two types of port: the same port type available to a DP-User, and an additional port type that is reserved for administrative functions, the DP-Administrator port.

A port type may be defined to be either symmetrical or asymmetrical. A symmetrical port type allows any of its defined abstract-operations to be invoked or performed by either of two objects associated by paired ports of that type. Conversely, an asymmetrical port type permits its abstract-operations to be invoked by just one of the two objects, called the consumer, and performed only by the other object, the supplier.

Note that the terms consumer and supplier are used only to distinguish between the roles assumed by a pair of ports in invoking or performing abstract-operations.

The assignment of the terms is usually intuitive where one object is providing a service used by another object. For example, a print server would usually be regarded as a supplier, and its users as consumers. Accordingly, the DP-User port is defined to be of the asymmetrical type, with the DP-

Server object and the DP-User object defined to be the supplier and the consumer, respectively, with respect to that port.

Similarly, the DP-Administration port is also of the asymmetrical type, with the DP-Server object as the supplier and the DP-Administrator object as the consumer.

Before an object can invoke abstract-operations on another, the two objects must be bound into an abstract-association. The binding of an abstract-association between objects establishes a relationship between the objects which lasts until the abstract-association is released. An abstract-association is always released by the initiator of the abstract-association.

The binding of an abstract-association validates the credentials needed for two objects to interact, and establishes the application-context and security-context of an abstract-association. The application-context of an abstract-association may comprise one or more types of ports paired between the two objects.

Because the model presented is abstract, it is not always possible for an outside observer to identify the boundaries between objects, or to decide on the moment or means by which abstract-operations occur. However, in some cases the abstract model will be realized. For example, a pair of objects which communicate through paired ports may be located in different open systems. In this case, the boundary between the objects is visible, the ports are exposed, and the operations may be supported by instances of OSI communication.

#### 6.1. Objects in the Document Printing Environment

The Document Printing environment can be decomposed into one central object, the Document Print Server (DP-Server), one or more Document Printing User objects (DP-Users), and one or more Document Printing Administrator objects (DP-Administrators).

##### 6.1.1 DP-Server Object

This functional entity performs abstract-operations invoked by DP-Users and DP-Administrators in the Printing Environment via ports. Some of these operations are related to the printing of documents, others are used for inquiry and management purposes, and the rest are used for administrative purposes.

The formal definition of the DP-Server object is as follows:

```
dp-server OBJECT
    PORTS { dp-user [S], dp-administration [S] }
    ::= id-dp-server
```

##### 6.1.2 DP-User Object

This functional entity may invoke operations that are related to the printing of documents and those that are used for inquiry and management purposes related to the progress of a print job(s) or to the characteristics or status of a print server.

The formal definition of the DP-User object is as follows:

```
dp-user OBJECT
    PORTS { dp-user [C] }
    ::= id-dp-user
```

##### 6.1.3 DP-Administrator Object

This functional entity may invoke all operations supplied by the DP-Server object.

The formal definition of the DP-Administrator object is as follows:

```
dp-administrator OBJECT
    PORTS { dp-user [C], dp-administration [C] }
    ::= id-dp-administrator
```

## 6.2 Ports Used in the Document Print Service Model

### 6.2.1 DP-User Port

This port is intended for submitting service requests that are directly related to the primary function of the print server, i.e., the printing of documents. This includes service requests related to retrieving and altering information about the status of print jobs. This port also provides facilities for inquiring about the status and characteristics of the print server.

This port is formally defined as follows:

```
dp-user PORT
    CONSUMER INVOKES {
        Print,
        ModifyJob,
        CancelJob,
        ListObjectAttributes }
    ::= id-pt-dp-user
```

### 6.2.2 DP-Administration Port

This port is intended for performing administrative functions (e.g., to exercise control over a print server). The use of this port is intended for use only by administrative users. [But note that this is an implementation and site management policy decision, not mandated by this Standard.]

This port is formally defined as follows:

```
dp-administration PORT
    CONSUMER INVOKES {
        PromoteJob,
        Interrupt,
        Pause,
        Resume }
    ::= id-pt-dp-administration
```

## 6.3 DPA Object Classes

As described in the preceding clauses, a DP-User (or DP-Administrator) is able to access the services offered by a DP-Server by invoking one or more of the operations defined for the port associated with that class of user. In order to specify completely the manner in which an operation is to be performed, the invoker generally must provide additional data in the form of an argument to the operation. The DP-Server provides details of the status and outcome of an operation in the form of results.

The DP-Server accomplishes the action(s) that has been requested via an operation invocation by managing and manipulating data entities that are referred to as DPA-Objects. These DPA-Objects are simply convenient collections of data that may represent objects that are defined elsewhere. The invoker of an operation provides many of the necessary additional argument details in the form of values for some of the DPA-Object components; similarly, the server may

inform a user of the status or outcome of an operation by providing values for DPA-Object components in the operation results.

The DPA-Objects are grouped into the following classes:

- a) Job
- b) Printer
- c) Server
- d) Media
- e) Font
- f) Named colour
- g) Transfer method
- h) Delivery method
- i) Job sheet
- j) Finishing method
- k) Output method
- l) Imposition function
- m) Scheduler

The following subclasses provide an informal description of each of these object classes, along with their purposes and relationships to the defined operations. In this narrative, and in the remainder of this document, the term user will be employed to refer to both DP-Users and DP-Administrators, except when necessary to distinguish between the two. Details of the objects that comprise the respective classes are to be found in clause 9.

#### 6.3.1 Job object class

When a user invokes a Print operation, the DP-Server (hereafter referred to generically as a print server) creates a body of information to be used in the management and control of that operation. The action of invoking the Print operation is referred to as a print request, and the body of information created as a result of that request is termed a print job, or simply, job.

Various components of a job object convey and record information about the status, progress, and other characteristics of the job. In addition, the document(s) to be printed is treated as an attribute of the job object. That is, the document(s) is treated as a component of the job object to be transferred transparently during the processing of the job.

#### 6.3.2 Printer object class

The principal object managed by a print server is a printer. A printer object is the repository of information about the printer that the object represents. Examples of the information that may be recorded by a printer object are the printer's identity, installed features, available media and form, etc.

In abstract terms, a printer object can be considered a grouping of certain attributes within the associated server's environment. Indeed, in the document processing model presented subsequently, the actual printer is not referenced at all, but is simply considered a latent component of the document presentation process. In this context, a printer object may thus represent either a physical printer or a logical printer.

A physical printer is expected to be some form of physical device capable of rendering graphical output on some physical medium; however, this (draft) standard makes no stipulation as to the actual physical characteristics embodied in a particular physical printer.

A logical printer is defined as some collection of printer characteristics that have been defined and grouped together as a convenience to users of a print server. A single logical printer may represent several physical printers; conversely, a single physical printer may be represented by several logical printers. By directing a print request to a particular logical printer, the user is

informing the associated print server that a certain predefined set of printer and server features and facilities are to be applied to the requested print operation.

The logical printer concept can also aid a print server in managing several physical printers in a manner that is transparent to the user population. That is, while a user might be directing a print request to a (logical) printer that embodies a certain set of features and functionality, the associated print server might actually assign the resulting job to any one of several physical printers possessing the features implied by the logical printer attributes. This would enable the server to balance the load among its associated printers, and to remove printers from service for maintenance reasons, while still offering the same functionality to its clients.

### 6.3.3 Server object class

The print server is the entity that provides print services to clients in a distributed system. Users desiring to print documents submit their print requests to a print server and receive from the print server information concerning the status and outcome of their requests.

A server object represents a particular print server, and is the repository of information about that particular server.

A print server is associated with one or more printers. To the normal user, the distinction between a print server and a physical printer may not be visually apparent, since a printer and the server associated with it may be housed in one physical entity. However, a print server may be capable of supporting multiple printers; and the printers supported by one server need not be of the same type, model, or capabilities, and there is no need that the server and its printer(s) be collocated.

#### NOTE 4

*This Standard deals only with the application level interface between a print server and the users of that print server; it does not mandate any particular server/printer configuration, nor does it address the physical interconnection between a server and the printer(s) associated with it. The physical connection and access protocol(s) used between server and printer are considered implementation-specific; while it is possible to utilize the same methodology and protocol(s) to interconnect associated servers and printers as are employed in the interface between the server and the user of that server, that is not addressed nor required by this standard.*

*In practical terms, the printer and the associated server may be one and the same. And a physical printer may be capable of functioning as a server if so assigned by the local administrator.*

### 6.3.4 Media object class

A media object represents a medium on which a document may be rendered. The most common example is paper, but also included are transparencies and virtually any physical entity on which an image may be rendered. Two different categories of media are addressed: logical and absolute media.

A logical medium incorporates a generic set of characteristics which indicate the approximate desired effect, and which may be realized by one or more absolute media.

An absolute medium uniquely specifies the realization of a certain medium; which physical medium is to be used and which optional electronic form or overlay is to be combined with it. A physical medium is a medium that has distinct physical characteristics such as size, weight, type or preprinted information.

### 6.3.5 Font resource class

As defined in ISO/IEC DIS 9541, a font resource consists of the total collection of information representing an abstract font in a device-independent manner, while a font object is a device-specific or process-specific instance of a font resource.

A DPA-Object of the Font resource Class is a summary collection of information about one particular font resource. By means of operation defined subsequently in clause 8, a DP-User may interrogate a DP-Server to determine the font resources supported by, or available through, the DP-Server.

Note that while the DP-Server may actually be providing the desired features by means of font object, communication with the DP-User about the font will be in terms of the font resource, since this relates to the desired effect rather than to implementation-specific details which should be of no consequence to the DP-User.

#### 6.3.6 Named colour object class

A Named colour object represents some particular colour effect that may be available at a printer, and that a user may wish to invoke in the printing of a document. The manner in which a printer actually realize some colour effect is not of concern to this standard, nor does this standard address the specific aspects of the representation of colour; these topics are the subject of other international standards, and are beyond the scope of this standard.

#### 6.3.7 Transfer method object class

A transfer method object represents one specific method available for transferring electronic documents to a print server (and printer). For example, a print request may convey the subject document immediately and directly, or the request argument may provide a reference to an object stored in some remote file system, which the print server is to fetch by some other means, as indicated in the attributes of the transfer method.

#### 6.3.8 Delivery method object class

A delivery method object represents some particular means available to the print server for delivering print job outputs, logs and notifications. For example, a particular print server might offer a secure-pickup facility with controlled access; or some means might be available for delivering the output by postal mail. Such details are recorded as attributes of the delivery method object.

#### 6.3.9 Job sheet object class

A Job sheet object represents a particular form of job sheet that may be printed at the beginning of the requested print output, or at the end of the output, or separating different parts of a print job output. Different print servers will have different numbers and formats of such things available, summarized by the job sheet objects.

#### 6.3.10 Finishing method object class

Finishing encompasses the different kinds of operations that may be applied to printed output to produce the final document(s). This includes such things as stapling (with various characteristics), binding, hole-drilling, etc. Each specific kind of finishing available on a particular printer is represented by a finishing method object, which records the characteristics of the method.

#### 6.3.11 Output method object class

Output method objects represent an additional family of operations that may be applied to the output of a print job. Such actions as sorting and collating are examples of output methods.

#### 6.3.12 Imposition function object class

Some print servers may provide one or more utility functions that can be applied to a composed (electronic) document to alter its description prior to imaging and rendering the document on the output media. Such utility functions are termed convenience impositions, and each is represented by an Imposition function object. Examples include bind-shifting; i.e.,



shifting the image to be placed on the physical page to allow for some binding operation; a more complicated example is "2-up signatures", where two page images are selected from the composed document, and transformed such that the page images will be printed side-by-side, and oriented so the printed sheet may be folded to make up part of a booklet.

### 6.3.13 Scheduler object class

Different print servers may support different scheduling algorithms for selecting and processing print jobs. The details of such algorithms is outside the scope of this standard. Summary information about such algorithms, however, may be collected in scheduler object, enabling administrators to create and select different algorithms, and users to inquire about those available and currently in effect.

## 6.4 Attributes

Each DPA-Object is represented by a set of Attributes which characterise that object. That is, each attribute provides a piece of information about, or derived from, the object to which it corresponds. DPA employs attributes to affect and modify the manner in which an operation is performed, to identify the objects upon which to operate, and to return details of the status and outcome of an operation.

For details of the DPA-Attributes see clause 9.

Some selected attributes are intended for ordering the objects by the value of these attributes. In the case of textual attributes the definition of collating sequences for specific languages is outside the scope of this standard.

The attribute concept and the associated ASN.1 specifications are imported from the OSI Directory standard (ISO/IEC 9594-2). They are documented in the following subclasses as an aid to the reader of this standard.

### 6.4.1 Introduction

An attribute consists of an attribute-type, which identifies the class of information given by an attribute, and the corresponding attribute-value(s), which is a particular instance of that class appearing in the entry (see Figure 3).

```
Attribute ::= SEQUENCE {  
    type      AttributeType,  
    values    SET OF AttributeValue -- at least one value is required -- }
```

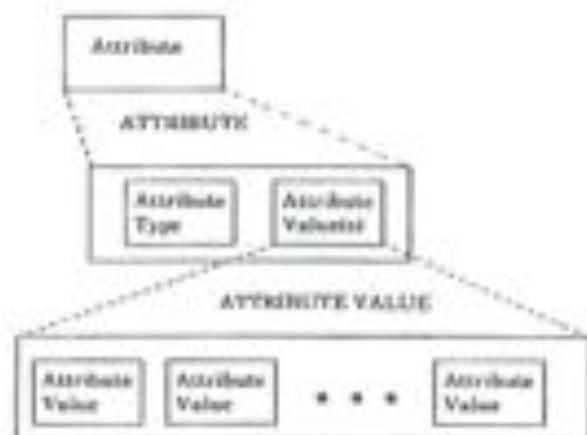


Figure 3. General Structure of Attributes

All attributes of a given object must be of distinct attribute-types.

For some attribute-types, an attribute may only contain a single attribute-value. Such an attribute-type is said to be single-valued. For others, an attribute may contain one or more attribute-values, all of the same ASN.1 data-type. Such an attribute-type is said to be multi-valued. Whether an attribute-type is single-valued or multi-valued is stated when the attribute-type is defined (see 6.3.7.2).

#### 6.4.2 Attribute-Type

Some attribute-types will be standardized. Other attribute-types will be defined by national administrative authorities and private organizations. Some externally defined attribute types will not be specific to DPA, for example security attributes. This implies that a number of separate authorities will be responsible for assigning types in a way that ensures that each is distinct from all other assigned types. This is accomplished by identifying each attribute-type with an object identifier when the type is defined.

AttributeType ::= OBJECT IDENTIFIER

Attribute-types defined in this Standard are specified in clause 9.

#### 6.4.3 Attribute-values

Defining an attribute-type also involves specifying the ASN.1 data-type to which every value in such attributes must conform. The data-type of an attribute-value for the attribute-type is defined through the object-identifier for the attribute-type.

AttributeValue ::= ANY -- DEFINED BY type --

#### 6.4.4 Attribute-type definition

The definition of an attribute-type involves :

- a) assigning an object identifier to the attribute-type;
- b) indicating or defining the data-type of an attribute-value;
- c) indicating whether an attribute of this attribute-type may have more than one value;
- d) indicating whether an attribute of this attribute-type may be used for filtering using equality, subclassing, and/or ordering relations (see 6.4.5).

*NOTE 3*

*A filter may always test for the presence or absence of an attribute of a particular attribute-type.*

The DPA ensures that the indicated attribute syntax is used for every attribute of this type. The DPA also ensures that attributes of this type will have one and only one value in entries if attributes of this type are defined to have only one value.

##### 6.4.4.1 The ATTRIBUTE Macro

The following ASN.1 macro is used to define an attribute-type:

ATTRIBUTE MACRO ::=

BEGIN

TYPE NOTATION ::= AttributeSyntax Multivalued | empty  
 VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)  
 AttributeSyntax ::= "WITH ATTRIBUTE-SYNTAX" SyntaxChoice  
 Multivalued ::= "SINGLE VALUE" | "MULTI VALUE" | empty  
 SyntaxChoice ::= value (ATTRIBUTE-SYNTAX)  
                   Constraint | type MatchTypes  
 Constraint ::= "(" ConstraintAlternative ")" | empty  
 ConstraintAlternative ::= StringConstraint | IntegerConstraint  
 StringConstraint ::= "SIZE" "(" SizeConstraint ")"  
 SizeConstraint ::= SingleValue | Range  
 SingleValue ::= value (INTEGER)  
 Range ::= value (INTEGER) "..." value (INTEGER)  
 IntegerConstraint ::= Range  
 MatchTypes ::= "MATCHES FOR" Matches | empty  
 Matches ::= Match Matches | Match  
 Match ::= "EQUALITY" | "SUBSTRINGS" | "ORDERING"

END

The correspondence between the parts of the definition and the various pieces of the notation introduced by the macro is as follows:

- (a) the object identifier assigned to the attribute type is the value supplied in the value assignment of the MACRO;
- (b) the attribute syntax for the attribute type is that identified by the AttributeSyntax production: either separately defined attribute syntax denoted by an object identifier, or explicitly the attribute syntax with ASN.1 type and matching rules. If a separately defined attribute syntax is employed, a size constraint for underlying string types or a value range for an underlying integer type may optionally be indicated;
- (c) the attribute is single valued if the Multivalued production is "SINGLE VALUE" and may have one or more values if it is "MULTI VALUE" or empty.

#### 6.4.4.2 Attribute-syntax definitions

The definition of an attribute syntax involves:

- a) optionally, assigning an object identifier to the attribute syntax;
- b) indicating the data type, in ASN.1, of the attribute syntax;
- c) defining appropriate rules for matching a presented value with a target attribute value. None, some, or all of the following matching rules may be defined for a particular attribute syntax:
  - i) equality. Applicable to any attribute syntax. The presented value must conform to the data type of the attribute syntax.

- ii) substrings. Applicable to any attribute syntax with a string data type. The presented value must be a sequence ("SEQUENCE OF"), each of whose elements conforms to the data type.
- iii) ordering. Applicable to any attribute syntax for which a rule can be defined that will allow a presented value to be described as less than, equal to, or greater than a target value. The presented value must conform to the data type of the attribute syntax.

If no equality matching rule is defined, the DP-Server:

- a) treats values as attributes of this attribute syntax as having type of ANY, i.e., the DPA does not check that those values conform with the data type indicated for the attribute syntax.
- b) will not attempt to match presented values against target values of such an attribute type.

If an equality matching rule is defined, the DP-Server:

- a) treats values of attributes of this attribute syntax as having type ANY defined by the data type indicated for the attribute syntax.
- b) will only match according to the matching rules defined for that attribute syntax.
- c) will only match a presented value of a suitable data type.

#### 6.4.4.3 The ATTRIBUTE-SYNTAX Macro

The following ASN.1 macro is used to define attribute syntaxes:

```

ATTRIBUTE-SYNTAX MACRO ::=
BEGIN
  TYPE NOTATION ::= Syntax MatchTypes | empty
  VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)

  Syntax ::= type
  MatchTypes ::= "MATCHES FOR" Matches | empty
  Matches ::= Match Matches | Match
  Match ::= "EQUALITY" | "SUBSTRINGS" | "ORDERING"
END

```

The correspondence between the parts of the definition and the various pieces of the notation introduced by the macro is as follows:

- (a) the object identifier assigned to the attribute syntax is the value supplied in the value assignment of the MACRO.
- (b) the data type of the attribute syntax is that identified by the Syntax production, i.e., that following the macro name.
- (c) the defined matching rules are equality, if "EQUALITY" appears in the MatchTypes production, substrings if "SUBSTRINGS" appears, and ordering if "ORDERING" appears. If the production is empty, then no matching rules are defined. Should the "empty" alternative of the notation be selected, the resulting notation ("ATTRIBUTE-SYNTAX") can be used to denote any possible attribute syntax.

*NOTE 6*

*No support is provided in the macro for actually defining the matching rules themselves; this must be done by natural language or by other means.*

**6.4.3 Filters**

A filter parameter applies a test that is either satisfied by a particular DPA-Object or not. The Filter is expressed in terms of assertions about the presence or value of certain attributes of the DPA-Object, and is satisfied if and only if it evaluates to true.

Filter ::= CHOICE {  
    item [0]FilterItem,  
    and [1]SET OF Filter,  
    or [2]SET OF Filter,  
    not [3]Filter}

A Filter is either a filter-item, or an expression involving simpler filters composed together using the logical operators and, or, and not.

Where the filter is:

- a) an item, it is true if and only if the corresponding filter-item is true;
- b) an and, it is true unless any of the filters in the SET are false;  
*NOTE 7*  
*Thus, if there are no filters in the SET, the and evaluates to true.*
- c) an or, it is false unless any of the filters in the SET are true;  
*NOTE 8*  
*Thus, if there are no filters in the SET, the or evaluates to false.*
- d) a not, it is true if and only if the filter is false.

**6.4.3.1 Filter-item**

A FilterItem is an assertion about the presence or value(s) of an attribute of a particular type in the DPA-Object under test. Each such assertion is either true or false.

FilterItem ::= CHOICE {  
    equality [0]Attribute/ValueAssertion,  
    substrings [1]SEQUENCE {  
        type AttributeType,  
        strings SEQUENCE OF CHOICE {  
            initial [0]Attribute/Value,  
            any [1] Attribute/Value,  
            final [2] Attribute/Value }},  
    greater-or-equal [2]Attribute/ValueAssertion,  
    less-or-equal [3]Attribute/ValueAssertion,  
    present [4]AttributeType,  
    approximate-match [5]Attribute/ValueAssertion}

Every filter-item includes an attribute-type which identifies the particular attribute concerned.

Any assertion about the value of such an attribute is only evaluated if the attribute-type is defined, and the purported attribute-value(x) conforms to the attribute syntax defined for that attribute type.

**NOTE 9**

*In the event of non-conformity, an AttributeError is reported.*

Assertions about the value of an attribute are evaluated using the matching rules associated with the attribute syntax defined for that attribute type. A matching rule not defined for a particular attribute syntax cannot be used to make assertions about that attribute.

**NOTE 10**

*In the event of non-conformity, an AttributeError is reported.*

Where the filter-item asserts:

- a) **equality**, it is true if and only if there is a value of the attribute which is equal to that asserted;
- b) **substrings**, it is true if and only if there is a value of the attribute in which the specified substrings appear in the given order. The substrings must be non-overlapping, and may be separated from the ends of the attribute-value and from one another by zero or more string elements.  

The first character in *initial*, if present, shall match the first character in the attribute value; the last character in *final*, if present, shall match the last character in the attribute value; *any*, if present, may match any substring in the attribute value;
- c) **greater-or-equal**, it is true if and only if the supplied value is greater than or equal to any value of the attribute;
- d) **less-or-equal**, it is true if and only if the supplied value is less than or equal to any value of the attribute;
- e) **present**, it is true if and only if such an attribute is present in the entry;
- f) **approximate-match**, it is TRUE if and only if there is a value of the attribute which matches that which is asserted by some locally-defined approximate matching algorithm (e.g. spelling variations, phonetic match, etc). There are no specific guidelines for approximate matching in this version of the Standard. If approximate matching is not supported, this FilterItem should be treated as a match for equality.

**NOTE 11**

*In the case of an attribute whose value is of character type, ordering (greater-or-equal and less-or-equal) is defined by the lexicographic order based upon a fixed collating sequence.*

6.4.5.2

**Attribute-value-assertion**

An **AttributeValueAssertion** is a proposition, which may be true, false, or undefined, concerning the values of a DPA-Object. It involves an attribute-type and an attribute-value:

**AttributeValueAssertion** ::= SEQUENCE {  
    type            AttributeType,  
    value           AttributeValue }

and is:

- a) **undefined**, if any of the following holds:
  - 1) the attribute-type is not present in the DPA-Object,

- 2) the definition of the attribute-type cannot be matched for equality.
  - 3) the attribute-value does not conform to the data type of the attribute-values;
- b) true, if the entry contains an attribute of that attribute-type, one of whose attribute-values matches that attribute-value;
  - c) false, otherwise.

#### 6.4.6 Qualified Attribute

The Document Printing Application assumes that there will exist many different configurations of print server, with potentially a large variation in the set of features and resources available at a particular print server. Likewise, because DPA utilizes the inherent extensibility of the Attribute type to model argument and result parameters, it is assumed that many different attributes will be defined over time, and it is assumed that not all of these attributes will be understood by all print servers or print clients.

In order to deal with this flexibility consistently, a print server is expected to make a best-effort attempt at honoring a print request; that is, unless instructed otherwise, a print server will ignore optional argument parameters that are not understood, or requests for features or resources that it is unable to supply.

In some cases, however, the print client requires that a server explicitly honour one or more such attributes. For this reason, the qualified attribute is defined, below, to enable a client to indicate to the server which optional attributes or extension attributes must be strictly obeyed.

```
QualifiedAttribute ::= SEQUENCE {  
    attribute  Attribute,  
    qualifier  Compsuory DEFAULT FALSE }  
Compsuory ::= BOOLEAN
```

If the qualifier is set to TRUE:

- 1) the server must understand the semantics of the attribute and its value,
- 2) the requested attribute and value must be supported (in the case of several compulsory attributes, all on the assigned printer(s)),
- 3) the server and printer(s) must be able to perform the requested service; e.g., any requested resources must be available.

If any of the above conditions cannot be satisfied, the server must return an error and abort the job.

If the qualifier is set to FALSE, the server is allowed to ignore the attribute or the attribute value, or take a default action, but ONLY if it does not understand or support the semantics of the attribute or its value.

In the case where not all non-compulsory attributes are supported on any associated printer, such attributes or values may be ignored in an implementation-dependent but consistent manner.

#### 6.5 Document Processing Model

Another view of the Document Printing Application is provided by the Document Processing Model, which shows the relationships between the different forms that a document may take, and the different processes involved in creating, formatting and printing a document.

Figure 4 illustrates the Document Processing Model as an informal refinement of the DPA environment. Its purpose is to aid in the understanding of the concepts and abstract entities involved;

it is not intended to specify any particular implementation. The major processes are characterized, along with their interrelationships, in terms of the flow of data and control among them.

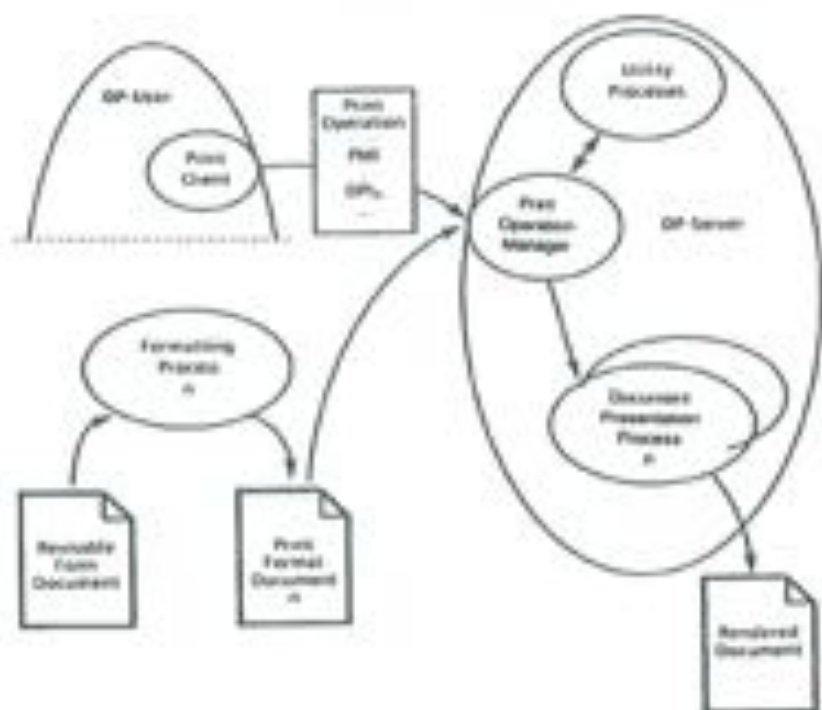


Figure 4. Document Processing Model

#### 6.5.1 Objects in the DP-User Environment

One major process is shown in the DP-User environment, the *Print Client*. The *Formatting Process* is very closely associated with the DP-User, but appears outside the DP-User's immediate environment since it may occur elsewhere. Other processes may be in, or related to, the DP-User's environment as well, but they are outside the scope of this discussion.

The *Print Client* is the interface between the initial requestor of printing and the print server itself. The print client accepts and acquires details of the requested printing action, transforms them into the form expected by the print server, and collects them into a print request consisting of the identity of the abstract-operation to be invoked, and the argument parameters to that operation. The completed request is then transferred to the desired print server by the supporting protocol layers. The argument parameters included with the print request are referred to collectively as *Print Operation Parameters (POP)*.

As illustrated in Figure 4, *Print Operation Parameters* may be subdivided into *Print Operation Management Instructions (PMI)* and *Document Production Instructions (DPI)*.

The *Formatting Process* is concerned with transforming the document(s) to be printed from its original revisable form into a form that is suitable for a printer to deal with. In the case of modern page description languages, it is the formatting process that supplies the additional information concerning precise placement of text and graphics on the pages, information about fonts to be used, etc. Note that some revisable formats may actually be in a form that some class of printers can deal with directly, without requiring a separate formatting process to be invoked. The rationale and details concerning different print formats is the subject of other standards, and is outside the scope of this Standard.



The formatting process does not necessarily reside in the same physical system as the Print Client, nor is the formatting process necessarily invoked at the same time a print request is made; the only requirement is that the document be available in the required printable form at the time it is conveyed to the document presentation process. It is also not a requirement that the document to be printed be resident on the same physical system as the Print Client. The Print Client may supply a reference to the document when the print request is invoked, enabling the transfer to be made independently of the association between the Print Client and the Print Server. Thus, in those systems that support the necessary facilities, the Print Client may supply a reference to the document that will enable the Print Server to acquire the document from a remote filestore, for example, by some means independent of the protocol used to transfer the print request.

#### 6.5.2 Objects in the DP-Server Environment

The abstract DP-Server environment includes one process that may be called a *Print Operation Manager*, one or more *Document Presentation Processes*, and optionally, one or more *Utility Processes*. Other processes may be included as well, but they are outside the scope of this discussion.

The Print Operation Manager is the counterpart of the Print Client, and is that part of the print server that interfaces with the print client for the purpose of accepting print requests and providing the client with status and completion information regarding print requests. The Print Operation Manager coordinates the activities of the other processes involved in the processing of print requests, scheduling print jobs, managing print queues, passing acquired data to the other processes, etc. The Print Operation Manager deals with the print operation parameters passed in a print request and may relay to one of the other processes for action.

A document presentation process is the counterpart of a specific type of formatting process. For example, an SPDL formatting process formats a document in accordance with the SPDL standard; likewise there must be a corresponding SPDL presentation process in order to interpret the content of an SPDL document, and prepare it for output to a printer or other presentation device.

In addition to dealing with a specific print format, a presentation process also may be passed a subset of the print operation parameters, referred to collectively as *document production instructions*. This subset concerns the final structure and appearance of the document to be printed, and may include binding and finishing instructions as well. Some print formats permit document production instructions to be carried within the print format document itself, as well as in the print request parameters that are transferred outside of the document. In those cases where a specific type of production instruction appears in both the print request and the print format document, it is the responsibility of the presentation process to resolve precedence. Such precedence rules must be specified in conjunction with the particular print format specification, and is outside the scope of this standard.

Some print servers may include one or more utility processes to provide some additional functions for clients. Examples include convenience imposition functions capable of altering the structure of a formatted (electronic) document such that multiple page images may be rendered on each sheet of paper; such functions facilitate the production of booklets, etc. In addition, an utility process could be included to format a document for printing, e.g. in the case of an ODA Processable Form document.

#### 6.6 Security in DPA

Two specific security mechanisms are addressed here: Authentication and Access authorization.

### 6.6.1 Authentication

At BIND time, the security subject represented by the DP-User is authenticated either by using password(s) or, in the case of a user who has been previously authenticated elsewhere, by checking a certified identity. The authentication mechanism verifies the credentials of the DP-User requesting access to the DP-Server. This does not, however, qualify the DP-User to access all DPA-Objects stored in the DP-Server, or to access the abstract-services implemented by the DP-Server.

Certified identities can also be presented subsequent to the BIND operation, as arguments to individual operations. This permits the BIND to be shared between multiple security subjects, the DP-User representing a different security subject for each of the operations concerned.

*NOTE 12*

*The object designator DP-Administrator will be used throughout this Standard primarily to identify those specific cases where it is necessary to distinguish between users with normal access privileges and those with some form of administrative privilege. Unless otherwise qualified, this and following discussions concerning DP-Users apply equally to DP-Administrators.*

### 6.6.2 Access authorization

Access authorization is controlled by a group of security-related attributes assigned to DPA-Objects, known collectively as a Control Attribute Package (CAP).

In order to provide the design freedom necessary to enable DP-Servers to operate under a wide variety of security policies, the DPA implementor should be free to choose control attribute types appropriate to the security regimes for which the implementation is targeted. In the longer term, standards will be defined for access authorization rules and for control attribute types, from which users of DPA will make their choices. Examples of such attributes might be the identity, the access rights, and the role of the initiator of a print job.

In the short term these standards are not available, and this standard defines one specific control attribute, the DpAccessList.

```
DpAccessListElement ::=
  SEQUENCE {
    access-id      AccessId,
    access-rights  DpAccessRights }
AccessId         ::= DistinguishedName
DpAccessRights   ::=
  ENUMERATED {
    dp-user      (0),
    dp-administrator (1) }
```

If the accessing DP-User has presented a Privilege Attribute Certificate (PAC) containing privilege attributes for a DPA-Object (e.g., a print job), these attributes will be used where appropriate, in conjunction with the attributes in the CAP, in order to determine what form of access the user may have to that object. Privilege attributes are optionally presented either at abstract bind time via a PAC type credentials argument (see 7.1.1) or explicitly at the time of the operation request in the privileges argument (see 8.1.2.3) or both. The DP-User may also specify a proxy PAC in a specific operation request. This proxy PAC is used by the DP-Server on behalf of the requesting DP-User (see 8.1.2.3).

```
PrivilegeAttributeCertificate ::= EXTERNAL
```

*NOTE 13*

*The detailed syntax of the PrivilegeAttributeCertificate is under study elsewhere.*

Each specific DPA realization may determine which CAP attributes it is intended to support according to some defined matching algorithms.

7. ABSTRACT ASSOCIATION INFORMATION

This clause specifies the information that must be defined or supplied to effect the BIND operation.

7.1 Abstract-bind-operation

An abstract-bind-operation opens the DP-Server ports. The initiator is the DP-User or DP-Administrator, while the responder is the DP-Server.

```

DpBind ::= ABSTRACT-BIND
          TO ( dp-user[S], dp-administration[S] )
          BIND
          ARGUMENT   DpBindArgument
          RESULT     DpBindResult
          BIND-ERROR DpBindError

```

The abstract-bind parameters needed by the DP-Server Port are defined and described in the present clause.

7.1.1 Bind-argument Parameters

The dp-bind-argument parameter identifies or authenticates the DP-User. It also accepts a set of restrictions for criteria to be returned as result of an abstract-operation.

The definition is:

```

DpBindArgument ::= SEQUENCE [
  credentials          [0] Credentials,
  retrieve-restrictions [1] Restrictions OPTIONAL, --default is none--
  bind-security        [2] BindSecurity OPTIONAL ]

```

1) credentials (M): Credentials may be exchanged between the DP-User and the DP-Server. Whether the DP-User is representing a specific end user or not is of no concern to the DP-Server. The DP-Server's view of the accessing subject's identity and access privileges is obtained from the credentials passed (though this view may be modified for a particular operation). Credentials serve to identify a user, authenticate a user, or certify the identity of a user previously authenticated externally. In the latter case, access control privileges associated with the user can also be passed. The full syntax and semantics of credentials when used for authentication purposes are outside the scope of this standard (since these matters are common to all bind operations). Use of certified security attributes is described in outline below. Semantic details are however not defined since this is dependent on the actual security policy formulated and implemented by the organisation operating the DP-Server.

The authentication of the user may have taken place external to the DP-Server; the resulting access control attributes will then be passed in the abstract bind process to allow the DP-Server to make subsequent access control decisions. This is the function of the PAC construct. Either the DP-User or the DP-Server may abort the abstract bind process if the authentication parameters do not justify successful completion of the abstract bind operation.

**Credentials** ::= CHOICE {  
     **simple** [0] **Creds**. -- used for initial authentication --  
     **certified** [1] **PrivilegeAttributeCertificate** }  
 -- used when initial authentication has already taken place external to the DP-Server --

A **Creds** contains a password associated with the DP-User.

A **PrivilegeAttributeCertificate** contains attributes associated with the DP-User, for example the user's name or security clearance. These can be used in making access control decisions (see 6.6.2).

2) **retrieve-restrictions** (O): This contains the restrictions on objects to be returned as a result of an abstract-operation. The restrictions remain until an abstract-unbind-operation is issued.

In the absence of this argument, the default is that no restrictions need to be performed.

This argument consists of the following components:

**Restrictions** ::= SET {  
     **maximum-result-length** [1] **ResultLength** OPTIONAL -- default is no restriction -- }

a) **maximum-result-length** (C): The maximum-result-length that the DP-User is prepared to accept as result of a list-object-attributes abstract-operation. Any result with a result-length exceeding the one specified will not be returned, but will result in an error unless the abstract-operation has explicitly overridden the restriction.

In the absence of this component, the default is that no read-restrictions on result-length need to be performed.

**ResultLength** ::= INTEGER (1..ub-integer)

3) **bind-security** (O): This specifies OSI security services required in the bind for example peer entity authentication of the software entities involved, or confidentiality, or integrity protection.

**BindSecurity** ::= EXTERNAL

### 7.1.2 Bind-result Parameters

The **dp-bind-result** parameter returns any authentication-attributes needed.

The definition is:

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

1) **authentication-attributes** (C): Information returned as confirmation of an authentication check, which is not constrained by this Part of this Standard.

**AuthenticationAttribute** ::= EXTERNAL

### 7.1.3 Bind-error parameters

The DP-Server may report:

- a **SecurityError** which indicates that the identity of the DP-User cannot be established on the basis of the information the DP-User supplied,
- a **ServiceError** which indicates an error related to the provision of the service.

```
DpBindError ::= CHOICE {  
    service-error    [0] ServiceProblem, -- see 3.4.5 --  
    security-error   [1] SecurityProblem -- see 3.4.6 -- }
```

The same errors may occur in DP abstract operations, and are described in 3.4.

## 7.2 Abstract-unbind operation

An abstract-unbind operation closes the user and/or administration port(s). There are no arguments or errors associated with the abstract-unbind operation.

```
DPUndbind ::= ABSTRACT-UNBIND  
    FROM { dp-user[S], dp-administration[S] }
```

## 8. ABSTRACT OPERATIONS

This clause defines the following abstract-operations:

- Print
- ModifyJob
- CancelJob
- ListObjectAttributes
- PromoteJob
- Interrupt
- Pause
- Resume

Certain print servers may choose to permit some of these operations to be invoked only by users possessing the appropriate access privilege level, e.g., system administrators.

### 8.1 Common data-types and values used in DPA abstract operations

#### 8.1.1 Imported data-types

Some data-types described in clause 6, as well as in this clause, are actually defined in other Standards. These imported data-types are:

From the OSI Directory (ISO/IEC 9594-2):

- Attribute,
- AttributeType,
- AttributeValue,
- AttributeValueAssertion,
- DistinguishedName.

From the OSI Directory (ISO/IEC 9594-3):

- Filter,
- FilterItem

From DOAM (ISO/IEC 10031-2 (DIS)):

- RDT-Reference

From MOTIS (ISO/IEC 10021-4)

- OFAAddressAndOrDirectoryName

## 8.1.2 Data-types common for most DPA abstract operations

CommonArguments ::= SET {  
    extensions [25] SET OF QualifiedAttribute OPTIONAL,  
    error-handling [27] ErrorHandlingMode DEFAULT all-or-nothing,  
    priority [28] Priority DEFAULT medium,  
    privileges [30] Privileges DEFAULT ( ) }

CommonResults ::= SEQUENCE {  
    extensions [25] SET OF QualifiedAttribute OPTIONAL }

### 8.1.2.1 ErrorHandlingMode

ErrorHandlingMode ::= ENUMERATED {  
    all-or-nothing (0),  
    until-first-error (1),  
    do-maximum (2) }

This parameter is only applicable to abstract operations that may perform multiple actions such as list-object-attributes.

### 8.1.2.2 Priority

Priority ::= ENUMERATED {  
    low (0),  
    medium (1),  
    high (2) }

This common parameter may be specified for any DPA abstract operation. It is useful in the context of a heavily loaded server to allow better access to some privileged requests, according to a locally applied priority policy.

This concept of operational priority is not necessarily related to any communication priority. This means that specification of this parameter may, but need not, influence the quality of service of the underlying communication layers.

The server does not mandatorily grant the abstract operation with the priority requested. The assignment of and association of any particular priority interpretation with any numeric value is site specific. The values defined in this subclass are intended to be advisory. This priority concept should not be confused with job-priority (9.3.4.6).

### 8.1.2.3 Privileges

Privileges ::= SEQUENCE {  
    operation-pac [0] PrivilegeAttributeCertificate OPTIONAL, -- see 6.6.2 --  
    proxy-pac [1] PrivilegeAttributeCertificate OPTIONAL }

The Privileges enable access certificates to be associated directly with an abstract-operation request replacing or supplementing those available from the abstract-bind within which the request is being made.

The operation-pac may be required for either of two reasons:

- a) When the access privileges established by the user during the bind are not sufficient to permit the requested operation.

- b) when the bind is being multiplexed between a number of users, and each operation is potentially under the control of a different user, who must present his own access privileges.

The proxy-pac may be required because the DP-Server shall make further access to another application on behalf of the DP-User, and itself would have insufficient access rights unless supplemented by those in the proxy-pac. The DP-Server that receives the proxy-pac itself then uses this as an operation-pac to the other application.

## 8.2 User Port Abstract Operations

The following ASN.1 productions make use of data types described in sub-clauses of 6.4.

### 8.2.1 Print

This operation allows a user to submit a print job request to the print server. A print job request contains the information needed by the print server to print a particular document. Multiple documents may be printed as a single print job by making multiple print operation invocations.

This abstract operation is formally defined as follows:

```
Print ::= ABSTRACT-OPERATION
  ARGUMENT PrintArgument
  RESULT PrintResult
  ERRORS ( AttributeError, -- see 3.4 --
          DocumentAccessError,
          PrinterError,
          SecurityError,
          SelectionError,
          ServiceError )
```

When the print job request is accepted by the print server, it creates a print job and responds with a print job identifier unique within the environment. Otherwise the print job request is rejected and the reason is reported.

For a single document print job or the first document of a multiple document print job, the first invocation of the Print operation will allow a client to pass all the job attributes, including the "production-instructions", "document-attributes" and "document-description" attributes of one document. The "job-identifier" attribute must be omitted. The Print operation will return a "job-identifier" attribute.

For a multiple document print job, each additional document requires an additional invocation of the Print operation. The argument for each additional invocation of the Print operation shall include the "job-identifier" attribute (returned by the server on the first invocation), plus the "production-instructions", "document-attributes" and "document-description" attributes for the additional document. Any other job attributes passed with these additional invocations will be ignored and the additional documents will inherit these additional job attributes from the first document submitted. "Production-instruction", "document-attributes" and "document-description" shall not be inherited from previously submitted documents.

The "document-sequence-number" attribute is used to specify one of the following on each print operation invocation:

- 1) this is a single document job;
- 2) this is the first document of a multiple document job;

- 3) this is one of the intervening documents in a multiple document job;
- 4) this is the last document in a multiple document job;
- 5) the last document in a multiple document job was submitted with the previous print operation invocation for this job.

Multiple document print jobs are not scheduled for printing until all documents have been submitted.

This standard assumes that printers and other server components are reset or returned to appropriate states prior to processing each job. The "reset-printer" attribute permits the client to suspend the resetting of the printer after particular documents, if desired for down-line loading of fonts, forms, PDL prologues, etc.

The parameters of a job and its associated documents completely (and independently) define the processing and printing of the job. Only scheduling is expected to be affected by the existence of other jobs and their parameters.

#### 8.2.1.1 Print-argument

The argument of this operation is composed of the following parameters:

```
PrintArgument ::= SEQUENCE {
  job-attributes          [0] SET OF JobAttribute OPTIONAL,
  document-description   [1] SET OF DocumentDescription,
  -- This set must include all possible components of DocumentDescription --
  COMPONENTS OF CommonArguments }
```

```
JobAttribute ::= QualifiedAttribute
  (WITH COMPONENTS {
    attribute
    (WITH COMPONENTS {
      type (
        id-at-job-identifier          -- see 9.3.1.1 --
        | id-at-document-sequence-number -- see 9.3.7.3 --
        | INCLUDES JobInformationTypes
        | INCLUDES JobResultsHandlingTypes
        | INCLUDES JobEventHandlingTypes
        | INCLUDES JobSchedulingInstructionsTypes
        | INCLUDES PrinterIdentificationTypes
        | INCLUDES ProductionInstructionsTypes
        | INCLUDES DocumentAttributeTypes
        | INCLUDES AccessAndAccountingTypes
        | INCLUDES JobSecurityTypes ),
      values
      qualifier
    })
  )
```



```
DocumentDescription ::= Attribute -- see 9.3.7 --  
  (WITH COMPONENTS {  
    type ( id-att-transfer-method  
          | id-att-document-format  
          | id-att-document-content ),  
    values  
  })
```

The following type productions outline the structure of each of the parameters of the print operation argument, and indicates the subclause in which they are defined.

```
DocumentAttributeTypes ::= AttributeType -- see 9.3.5 --  
  ( id-att-content-formats-used  
  | id-att-assured-reproduction-area  
  | id-att-content-orientation  
  | id-att-page-order  
  | id-att-page-size  
  | id-att-page-independent  
  | id-att-character-sets-used  
  | id-att-character-mappings-used  
  | id-att-fonts-used  
  | id-att-character-repertoires-used  
  | id-att-media-used  
  | id-att-named-colours-used  
  | id-att-document-descriptor  
  | id-att-document-authors  
  | id-att-document-message  
  | id-att-document-revision-date  
  | id-att-ocfl-count  
  | id-att-page-count  
  | id-att-glyph-count  
  | id-att-font-count  
  | id-att-font-change-count  
  | id-att-maximum-fonts-per-page  
  | id-att-percent-graphics  
  | id-att-percent-images )
```

```
JobEventHandlingTypes ::= AttributeType -- see 9.3.3 --  
  ( id-att-notification-profile  
  | id-att-logging-profile  
  | id-att-job-abort-criteria  
  | id-att-job-warning-criteria )
```

```
JobInformationTypes ::= AttributeType -- sec 9.3.1 --
    ( id-at-job-name
    | id-at-job-originator
    | id-at-job-owner
    | id-at-job-comment
    | id-at-job-start-message
    | id-at-job-end-message )

JobResultsHandlingTypes ::= AttributeType -- sec 9.3.2 --
    ( id-at-results-profile )

JobSchedulingInstructionsTypes ::= AttributeType -- sec 9.3.4 --
    ( id-at-job-message-to-operator
    | id-at-job-start-wait
    | id-at-job-end-wait
    | id-at-job-hold
    | id-at-job-password
    | id-at-job-print-after
    | id-at-job-priority
    | id-at-job-deadline-time
    | id-at-job-retention-time
    | id-at-job-retention-period )

PrinterIdentificationTypes ::= AttributeType -- sec 9.3.5 --
    ( id-at-printer-identifier
    | id-at-printer-location
    | id-at-printer-model
    | id-at-printer-type
    | id-at-printer-security
    | id-at-printer-speed
    | id-at-printer-resolution )

ProductionInstructionsTypes ::= AttributeType -- sec 9.3.6 --
    ( id-at-font-substitution
    | id-at-media-substitution
    | id-at-named-colour-substitution
    | id-at-finishing-specification
    | id-at-output-specification
    | id-at-output-bin
    | id-at-imposition-specification
    | id-at-print-quality
    | id-at-font-fidelity
    | id-at-media-fidelity
    | id-at-colour-fidelity )
```

```

| id-att-sides
| id-att-page-select
| id-att-page-media-select
| id-att-copy-count
| id-att-start-sheet
| id-att-separator-sheet
| id-att-end-sheet
| id-att-reset-printer)

```

```

AccessAndAccountingTypes ::= AttributeType -- see 9.3.9 --
( id-att-user-name
| id-att-accounting-information )

```

```

JobSecurityTypes ::= AttributeType -- see 9.3.11 --
( id-att-job-security-handling )

```

### 8.2.1.2 Print-result

The result of this operation is composed of the following parameters:

```

PrintResult ::= SEQUENCE {
  job-identification [0] JobIdentification,
  returned-information [1] SET OF ReturnedInformation OPTIONAL,
  COMPONENTS OF CommonResults }

```

```

JobIdentification ::= Attribute
( WITH COMPONENTS {
  type ( id-att-job-identifier ),
  values } )

```

```

ReturnedInformation ::= Attribute
( WITH COMPONENTS {
  type ( id-att-server-state -- see 9.3.4 --
| id-att-server-message -- see 9.3.3 --
| id-att-file-reference -- see 9.3.7.5 --
| INCLUDES JobStatusTypes),
  values } )

```

```

JobStatusTypes ::= AttributeType -- see 9.3.10 --
( id-att-current-job-state
| id-att-previous-job-state
| id-att-printers-assigned
| id-att-estimated-completion-time
| id-att-submission-time
| id-att-modification-time
| id-att-started-printing-time
| id-att-copies-completed
| id-att-pages-completed
| id-att-octets-completed

```

- | id-att-intervening-pages
- | id-att-intervening-octets
- | id-att-intervening-jobs
- | id-att-print-checkpoint
- | id-att-new-server-name
- | id-att-on-request-resources-required
- | id-att-job-message-from-administrator
- | id-att-last-job-events
- | id-att-id-of-last-accessor
- | id-att-interrupt-at
- | id-att-pause-at )

In the case of multiple-document jobs, the returned job status information applies to the job as a whole, not to any individual part of the job.

**NOTE 14**

*JobStatusTypes* defines a collection of attribute types relating to job status, some of which may be returned by different operation types. However, many attributes included in *JobStatusTypes* will not be returned by the Print operation because values are not yet available.

**8.2.1.3 Print Abstract-errors**

Should the request fail, one of the listed abstract-errors will be reported. The circumstances under which the particular abstract-errors will be reported are defined in 8.4.

**8.2.2 Modify-job**

The modify-job abstract-operation is used to alter the values of specified print job argument parameters after the print job has been accepted by the print server.

This abstract operation is formally defined as follows:

```

ModifyJob ::= ABSTRACT-OPERATION
  ARGUMENT      ModifyJobArgument
  RESULT        ModifyJobResult
  ERRORS        ( AccessError,      -- see 8.4 --
                  AttributeError,
                  DocumentAccessError,
                  PrinterError,
                  SecurityError,
                  SelectionError,
                  ServiceError,
                  UpdateError )

```

The effect is the same as if the job had been submitted originally with the modified parameter values, although scheduling of the original job may be impacted.

Not all parameters associated with a print job may be modified. That is, support for modification of other attributes is dependent on the implementation.

"AccessAndAccountingTypes" and "JobSecurityTypes" attributes cannot be modified. If any specified modification cannot be fulfilled (e.g., due to conflicting requirements, unavailability of a feature, or current status of the job or system), the modify-job abstract-operation is rejected and processing of the print job continues as accepted originally.

### 8.2.2.1 Modify-job-argument

The argument of this operation is formally defined as follows:

```

ModifyJobArgument ::= SEQUENCE {
    job-identification      [0] JobIdentification, -- see 8.2.1.2 --
    document-identification [1] DocumentIdentification OPTIONAL,
    -- required for addressing individual documents in a multiple document print job --
    job-atr-modification    [2] SET OF JobAtrModification OPTIONAL,
    COMPONENTS OF CommonArguments }

```

```

DocumentIdentification ::= Attribute
    ( WITH COMPONENTS {
        type ( id-atr-document-sequence-number ),
        values } )

```

```

JobAtrModification ::= QualifiedAttribute
    ( WITH COMPONENTS {
        attribute
        ( WITH COMPONENTS {
            type ( id-atr-job-identifier
                | id-atr-document-sequence-number
                | INCLUDES JobInformationTypes
                | INCLUDES JobResultsHandlingTypes
                | INCLUDES JobEventHandlingTypes
                | INCLUDES JobSchedulingInstructionsTypes
                | INCLUDES PrinterIdentificationTypes
                | INCLUDES ProductionInstructionsTypes ),
            values },
        qualifier } )

```

If the optional "document-sequence-number" element is omitted, any "production-instructions" specified for modification, will apply to all documents in the print job. If a "document-sequence-number" is supplied, only the "production-instructions" for that document will be affected.

Support for the ability to modify parameters of individual documents of a multiple document print job is implementation dependent. Servers not implementing this shall return an error if "document-sequence-number" is supplied, rather than modifying all the documents in the entire job.

### 8.2.2.2 Modify-job-result

The result of this operation is composed of the following parameters:

```

ModifyJobResult ::= SEQUENCE {
    returned-information [0] SET OF ReturnedInformation OPTIONAL,
    -- see 8.2.1.2 for ReturnedInformation definition --
    COMPONENTS OF CommonResults }

```

### 8.2.2.3 Modify-job Abstract-errors

Should the request fail, one of the listed abstract-errors will be reported. The circumstances under which the particular abstract-errors will be reported are defined in 8.4.

### 8.2.3 Cancel-job

This operation allows a user to cancel one specific print job or a document of a multi-document job, before printing has been completed. Depending on when the cancel request is received by the print service, some pages may be printed before a job is successfully terminated.

This abstract operation is formally defined as follows:

```
CancelJob ::= ABSTRACT-OPERATION
  ARGUMENT      CancelJobArgument
  RESULT        CancelJobResult
  ERRORS        ( AccessError,
                  NameError,
                  SecurityError,
                  SelectionError,
                  ServiceError,
                  UpdateError )
```

Normally a print job can be cancelled only by the user who initiated the print job by means of a print operation. However, this is an implementation and site-specific decision, not mandated by this standard.

#### 8.2.3.1 Cancel-job-argument

The argument of this abstract operation is composed of the following parameters:

```
CancelJobArgument ::= SEQUENCE (
  job-identification      [0]JobIdentification,
  document-identification [1]DocumentIdentification OPTIONAL )
  -- required for addressing individual documents in a multiple document print job --
  COMPONENTS OF CommonArguments )
```

If the optional "document-identification" element is omitted, the entire print job is cancelled.

If "document-identification" is supplied with a value, the individual document specified will be cancelled. Support for the ability to cancel individual documents of a multiple document print job is implementation dependent. Servers not implementing this shall return an error if "document-number" is supplied and the CancelJob operation will be rejected. The identified job will not be affected in this case.

#### 8.2.3.2 Cancel-job-result

The result of this abstract operation is structured into parameters as follows:

```
CancelJobResult ::= SEQUENCE (
  returned-information [0] SET OF ReturnedInformation OPTIONAL, -- see 8.2.1.2 --
  COMPONENTS OF CommonResults )
```

#### 8.2.3.3 Cancel-job Abstract-errors

Should the request fail, one of the listed abstract-errors will be reported. The circumstances under which the particular abstract-errors will be reported are defined in 8.4.

## 8.2.4 List-object-attributes

This operation enables a user to obtain information from a print server concerning jobs, printers, and other objects known to the server. By means of the specified selection criteria, this operation enables a client to request and obtain values for one or more attributes, associated with one or more DPA-Objects, belonging to some object class.

This abstract operation is formally defined as follows:

```
ListObjectAttributes ::= ABSTRACT-OPERATION
    ARGUMENT      ListObjectAttributesArgument
    RESULT        ListObjectAttributesResult
    ERRORS        { AccessError,
                  AttributeError,
                  NameError,
                  SecurityError,
                  ServiceError }
```

### 8.2.4.1 List-object-attributes-argument

The argument of this abstract-operation is composed of the following parameters:

```
ListObjectAttributesArgument ::= SEQUENCE {
    object-class      [0] OBJECT IDENTIFIER,
    continuation     [1] ContinuationContext OPTIONAL,
    selector         [2] Selector OPTIONAL,
    requested-attributes [3] SEQUENCE OF AttributeType OPTIONAL,
    COMPONENTS OF CommonArguments }
```

```
ContinuationContext ::= OCTET STRING -- implementation-specific information --
```

```
Selector ::= SET {
    object-identification [0] SET OF ObjectIdentification OPTIONAL,
    -- There is only one member in this set, except for the case when a particular document --
    -- of a multiple documents print job is selected --
    object-filter         [1] Filter,
    time-limit           [2] DeltaTime OPTIONAL,
    count-limit          [3] INTEGER (formal-integer) OPTIONAL }
```

```
ObjectIdentification ::= Attribute
    [ WITH COMPONENTS {
        type (
            id-at-job-identifier
            id-at-document-sequence-number
            id-at-printer-identifier
            id-at-server-identifier
            id-at-medium-identifier
            id-at-font-identifier
            id-at-named-colour-identifier
            id-at-transfer-method-identifier
            id-at-delivery-method-identifier
            id-at-job-sheet-identifier
            id-at-finishing-method-identifier
            id-at-output-method-identifier
            id-at-imposition-function-identifier
            id-at-scheduler-identifier ),
        values
    } ]
```

```
DeltaTime ::= INTEGER -- time in seconds --
```

When the object-class is a print job, and if a document-sequence is specified, object-identification must be a "job-identifier" attribute, and is required. If the optional "document-sequence-number" element of "object-identification" is omitted, any "production-instruction", "document-attribute" or "document-description" attributes will be those of all the documents in the job. If a "document-sequence-number" element is supplied, any "production-instructions", "document-attribute" and "document-description" attributes requested will be those of the specified document.

The continuation-context enables a client to invoke a sequence of list operations in order to acquire a potentially large collection of data in the form of more manageable subsets of the total. The first (or only) list operation in a sequence will omit continuation-context, or specify a null value. Each return from the list operation indicates whether a limit has been reached and supplies a continuation-context value. The client includes this value in the next list operation argument to enable the server to resume the data collection from the point it had left off on the previous invocation.

Time-limit specifies the maximum time duration the user has allocated to the print server for completion of the requested list operation. If the server has not completed listing all object attributes that satisfy the criteria, the print server will return the information that has been collected, and will provide a continuation-context value to enable the client to resume the data collection with a subsequent list operation.

The count-limit value indicates the maximum number of object-instances for which attributes should be returned by the operation invocation. If not all of the data has been returned that satisfy the specified criteria, a continuation-context will be returned to permit the data collection to be resumed on a subsequent list operation invocation.

#### 8.2.4.2 List-object-attributes-result

The result of this abstract-operation is formally defined as follows.

```
ListObjectAttributesResult ::= SEQUENCE {
    answer-time      [1] UTCTime,
    continuation    [2] ContinuationContext OPTIONAL,
    limit-encountered [3] LimitEncountered OPTIONAL,
    attributes       [4] SEQUENCE OF ObjectResult }
```

```
LimitEncountered ::= ENUMERATED {
    time-limit [0],
    count-limit [1] }
```

```
ObjectResult ::= SEQUENCE {
    object-identification [0] SET OF ObjectIdentification
    -- There is only one member in this set, except for the case when a particular --
    -- document of a multiple document print job is selected --,
    attributes            [1] SET OF Attribute }
```

Answer-time represents the time at which the data was collected.

The other parameters and sub-parameters are defined in previous classes.

#### 8.2.4.3 List-object-attributes Abstract-errors

Should the request fail, one of the listed abstract errors will be reported. The circumstances under which the particular abstract-errors will be reported are defined in 8.4.



### 8.3 DP-administration Part Operations Definition

The operations described in the following classes may be used for administering and controlling the flow of print jobs through printers. Specific print server implementations may limit usage of one or more of these operations to users with sufficient access rights, and/or may vary the facilities provided by a given operation depending upon the access rights assigned to the individual users served by the print server.

It is not required that administration be possible in all cases between heterogeneous implementations over company boundaries.

The following ASN.1 productions make use of data types described in sub-classes of 6.4.

#### 8.3.1 Promote-job

This operation allows an administrative user (e.g., a system administrator) to indicate that the job associated with job-identifier is to be scheduled for printing on the next available matching printer. The mechanism used to provide this functionality is implementation-specific (e.g., it may be implemented by: increasing job priority; or putting all other jobs on hold; or moving job to the top of a queue; etc.).

This abstract operation is formally defined as follows:

```
PromoteJob ::= ABSTRACT-OPERATION
  ARGUMENT   PromoteJobArgument
  RESULT     PromoteJobResult
  ERRORS     ( AccessError,
               AttributeError,
               NameError,
               SecurityError,
               ServiceError )
```

##### 8.3.1.1 Promote-job-argument

```
PromoteJobArgument ::= SEQUENCE {
  job-identification [0] JobIdentification, -- see 8.2.1.2 --
  COMPONENTS OF CommonArguments }
```

##### 8.3.1.2 Promote-job-result

```
PromoteJobResult ::= SEQUENCE {
  returned-information [0] SET OF ReturnedInformation OPTIONAL, -- see 8.2.1.2 --
  COMPONENTS OF CommonResults }
```

##### 8.3.1.3 Promote-job Abstract-errors

Should the request fail, one of the listed abstract-errors will be reported. The circumstances under which the particular abstract-errors will be reported are defined in 8.4.

#### 8.3.2 Interrupt

This operation allows an administrator to interrupt a currently printing job or printer so that another job may be printed on the printer. The interrupted job will resume processing automatically at the point of interrupt checkpoint, upon completion of the interrupting job.

This operation shall not be used to interrupt a job that has already interrupted another job.

This abstract operation is formally defined as follows:

```

Interrupt ::= ABSTRACT-OPERATION
ARGUMENT      InterruptArgument
RESULT        InterruptResult
ERRORS        { AccessError,
                AttributeError,
                NameError,
                SecurityError,
                SelectorError,
                ServiceError }

```

**NOTE 13**

Normally a print job or printer can be interrupted only by an appropriately authorized user (such as an operator or system administrator). However, this is a site and/or enterprise-specific policy decision, not mandated by this standard.

**8.3.2.1 Interrupt-argument**

The argument of this abstract-operation is composed of the following parameters:

```

InterruptArgument ::= SEQUENCE {
  interrupted-object-identification [0] Attribute
    ( WITH COMPONENTS {
      type (id-at-job-identifier
            | id-at-printer-identifier ),
      values } ),
  interrupting-job-identification [1] Attribute
    ( WITH COMPONENTS {
      type (id-at-job-identifier ),
      values } )
  COMPONENTS OF CommonArguments }

```

**8.3.2.2 Interrupt-result**

The result of this abstract-operation is composed of the following parameters:

```

InterruptResult ::= SEQUENCE {
  interrupted-object-identification [0] Attribute
    ( WITH COMPONENTS {
      type (id-at-job-identifier
            | id-at-printer-identifier ),
      values } ),
  returned-information[1] SET OF ReturnedInformation OPTIONAL, -- see 8.2.1.2 --
  COMPONENTS OF CommonResults }

```

**8.3.2.3 Interrupt Abstract-errors**

Should the request fail, one of the listed abstract-errors will be reported. The circumstances under which the particular abstract-errors will be reported are defined in 8.4.

**8.3.3 Pause**

This operation allows an administrator to pause an idle or currently printing job or printer. The job or printer must be resumed using the Resume operation to return it to its pre-paused state.

Pausing a printer stops the printing of a job on the printer, leaves the printer paused without changing its internal state, and leaves the job assigned to the printer. Pausing a job stops the printing of this job on the printer, leaves the printer active, and allows other jobs to be assigned to this printer.

This abstract-operation is formally defined as follows:

```
Pause ::= ABSTRACT-OPERATION
  ARGUMENT PauseArgument
  RESULT   PauseResult
  ERRORS   ( AccessError,
             AttributeError,
             NameError,
             SecurityError,
             SelectionError,
             ServiceError )
```

Normally a print job or printer can be paused only by an appropriately authorized user (such as an operator or system administrator).

#### 8.3.3.1 Pause-argument

The argument of this abstract operation is structured into parameters as follows:

```
PauseArgument ::= SEQUENCE {
  paused-object-identification [0] Attribute
  ( WITH COMPONENTS {
    type ( id-att-job-identifier
          | id-att-printer-identifier ),
    values
  } ),
  pause-message [1] Attribute
  ( WITH COMPONENTS {
    type ( id-att-job-message-from-administrator
          | id-att-printer-message ),
    values
  } ) OPTIONAL,
  COMPONENTS OF CommonArguments }
```

#### 8.3.3.2 Pause-result

The result of this operation is composed of the following parameters:

```
PauseResult ::= SEQUENCE {
  paused-object-identification [0] Attribute
  ( WITH COMPONENTS {
    type ( id-att-job-identifier
          | id-att-printer-identifier ),
    values
  } ),
  returned-information [1] SET OF ReturnedInformation OPTIONAL,
  - see 8.2.1.2 for ReturnedInformation definition -
  checkpoint [2] PrintCheckpoint,
  COMPONENTS OF CommonResults }
```

```

PrintCheckpoint ::= SEQUENCE { -- see 9.3.10.4 --
  document-number [0] INTEGER (1..ub-integer) DEFAULT 1,
  page-number      [1] INTEGER (1..ub-integer) DEFAULT 1,
  copy-number      [2] INTEGER (1..ub-integer) DEFAULT 1,
  context-info     [3] OCTET STRING OPTIONAL -- implementation dependent --
}

```

The checkpoint information is supplied by the print server to indicate where the job may be resumed later. This information is returned to the print server subsequently to enable it to resume job processing.

### 8.3.3.3 Pause Abstract-errors

Should the request fail, one of the listed abstract-errors will be reported. The circumstances under which the particular abstract-errors will be reported are defined in 8.4.

### 8.3.4 Resume

This operation allows an administrator to resume a currently paused job or printer.

This abstract operation is formally defined as follows:

```

Resume ::= ABSTRACT-OPERATION
  ARGUMENT      ResumeArgument
  RESULT        ResumeResult
  ERRORS        { AccessError,
                  AttributeError,
                  NameError,
                  SecurityError,
                  SelectionError,
                  ServiceError }

```

Normally a print job or printer can be resumed only by an appropriately authorized user (such as an operator or system administrator).

#### 8.3.4.1 Resume-argument

The argument of this abstract-operation is structured into parameters as follows:

```

ResumeArgument ::= SEQUENCE {
  resumed-object-identification [0] Attribute
  ( WITH COMPONENTS {
    type ( id-att-job-identifier
           | id-att-printer-identifier ),
    value
  } ),
  checkpoint [1] PrintCheckpoint, -- see 8.3.1.2 --
  COMPONENTS OF CommonArguments }

```

#### 8.3.4.2 Resume-result

The result of this operation is composed of the following parameters:

```

ResumeResult ::= SEQUENCE {
  returned-information [1] SET OF ReturnedInformation OPTIONAL, -- see 8.2.1.2 --
  COMPONENTS OF CommonResults }

```

### 8.3.4.3 Resume Abstract-errors

Should the request fail, one of the listed abstract-errors will be reported. The circumstances under which the particular abstract-errors will be reported are defined in 8.4.

## 8.4 Abstract-Errors

This clause defines the abstract-errors associated with using the abstract-operations at the DP-Server ports.

If an error is returned as the result of invoking an abstract operation, this means that the abstract operation has changed nothing (this does not apply, however, to warnings, reported by the DP-Server as part of a normal result of an abstract operation).

Associated with each error there is a problem parameter. This parameter is a refinement of the related error; it may be either a standard-problem (that is, a value defined in this standard) or an extended-problem (that is, an implementation-defined value), in which case it is defined as an object identifier.

The following ASN.1 productions make use of data types described in sub-clauses of 6.4.

### 8.4.1 Access-error

An AccessError reports a problem encountered when attempting to access a DP-object specified in the argument of an abstract operation.

```
AccessError ::= ABSTRACT-ERROR
  PARAMETERSEQUENCE OF SEQUENCE {
    job-identifier [0] JobIdentifier, -- see 9.1.1.1 --
    problem [1] AccessProblem }
```

```
AccessProblem ::= ENUMERATED {
  inappropriate-object-class (1),
  insufficient-access-rights (2) }
```

An AccessProblem may be one of the following:

- a) inappropriate-object-class: the abstract-operation cannot be applied to this object.
- b) insufficient-access-rights: an attempt to access attribute values has been made by a DP-User who does not have sufficient access rights to a DPA-Object or the attributes.

### 8.4.2 Attribute-error

An AttributeError reports one or several problems encountered by the DP-Server while attempting to submit a print job or to list or modify attributes of one or more objects.

```
AttributeError ::= ABSTRACT-ERROR
  PARAMETER SEQUENCE {
    job-identifier [0] JobIdentifier OPTIONAL,
    problems [1] SEQUENCE OF SEQUENCE {
      problem [0] AttributeProblem,
      type [1] AttributeType,
      value [2] AttributeValue OPTIONAL } }
```

```

AttributeProblem ::= CHOICE {
  standard-problem    ENUMERATED {
    no-such-attribute          (1),
    invalid-attribute-syntax  (2),
    undefined-attribute-type  (3),
    inappropriate-matching   (4),
    constraint-violation      (5),
    attribute-or-value-already-exists (6),
    illegal-modification      (7),
    inconsistent-with-other-attributes (8),
    undefined-attribute-value  (9),
    unsupported-features      (10) },
  extended-problem    OBJECT IDENTIFIER }

```

The *job-identifier* parameter is absent in an *AttributeError* in the following cases:

- when the abstract operation causing this error was a *Print* abstract operation, and an *AttributeProblem* has been encountered when processing the attributes component of the abstract operation's argument;
- when the abstract operation causing this error was a *ListObjectAttributes* abstract operation, and an *AttributeProblem* has been encountered when processing the search criteria component of the abstract operation's argument.

The *problems* parameter specifies one or several *AttributeProblem*s encountered. Each problem (identified below) is accompanied by an indication of the *AttributeType*, and, if necessary to avoid ambiguity, the value, that caused the problem:

- a) *no-such-attribute*: The named object lacks one of the attributes specified as an argument of the abstract operation. (This does not apply when attempting to read an attribute which is absent);
- b) *invalid-attribute-syntax*: A purported attribute value, specified as an argument of the abstract operation, does not conform to the attribute syntax of the attribute type.
- c) *undefined-attribute-type*: An undefined attribute type was provided as an argument to the abstract operation;
- d) *inappropriate-matching*: An attempt was made, e.g. in a filter, to use a matching rule not defined for the attribute type concerned;
- e) *constraint-violation*: An attribute value supplied (or specified indirectly) in the argument of an abstract operation does not conform to the static constraints imposed by a functional standard or by the attribute definition (e.g. the value exceeds the maximum size allowed);
- f) *attribute-or-value-already-exists*: An attempt was made to add an attribute which already existed in the object, or a value which already existed in the attribute. (It does not apply, however, to the case when the values to be added have been specified indirectly);
- g) *illegal-modification*: An attempt was made to modify an attribute (i.e. to add or remove the entire attribute or some of its values), which has some specific behaviour in DPA, that is, either a DP-Server assigned attribute, or an attribute which, once assigned by the DP-User, may not be modified in the way specified in the operation;

- h) *inconsistent-with-other-attributes*: An attempt was made to modify an attribute in a way inconsistent with other attributes of the same print job. An inconsistency with some existent attribute is not reported if it is eliminated by further modifications specified in the same abstract operation. If two attributes enter in conflict, the DP-Server may report an *AttributeProblem* for either of them, or for both.
- i) *undefined-attribute-value*: An *AttributeValue* was specified which is not defined for the DP-Server.
- j) *unsupported-features*: An *AttributeType* was specified which is not defined for the DP-Server.

#### 8.4.3 Document-access-error

A *DocumentAccessError* reports a problem occurring when attempting to access the document referenced in the print job. This applies to internal as well as to external references.

```
DocumentAccessError ::= ABSTRACT-ERROR  
    PARAMETERSEQUENCE {  
        problem [0] DocumentAccessProblem }
```

```
DocumentAccessProblem ::= CHOICE {  
    standard-problem    ENUMERATED {  
        value-not-available (1),  
        referent-modified (2),  
        access-denied (3),  
        unknown-document (4) },  
    extended-problem    OBJECT IDENTIFIER }
```

A *DocumentAccessProblem* may be one of the following:

- a) *value-not-available*: The document specified in the abstract operation's argument is not available at that moment because of some (temporary) access problems (e.g. when attempting to read the content of the document).
- b) *access-denied*: An attempt to read the referenced document has been made by a DP-User with insufficient access rights.
- c) *unknown-document*: The document specified in the abstract operation's argument has been deleted ("dangling reference").
- d) *referent-modified*: The document specified in the abstract operation's argument has been modified after the produce-time of the reference (the latter being stored in the qos-level component of the RDT-reference content).

#### 8.4.4 Name-error

A *NameError* reports a problem related to a name of a DPA-Object specified in the argument of an abstract operation. The *DPAObjectName* which caused a problem is reported as it was specified, accompanied by an indication of the problem encountered.

```
NameError ::= ABSTRACT-ERROR  
    PARAMETERSEQUENCE OF SEQUENCE {  
        job-identifier [0] JobIdentifier, -- see 9.3.1.1 --  
        problem [1] NameProblem }
```

```
NameProblem ::= CHOICE {  
    standard-problem    ENUMERATED {  
        invalid-job-identifier (1)},  
    extended-problem    OBJECT IDENTIFIER }
```

A NameProblem may be one of the following:

- a) invalid-job-identifier: The JobIdentifier provided in the abstract operation's argument does not refer to any print job (either this job-identifier has never been assigned, or the related print job has been finished and deleted);

#### 8.4.5 Printer-error

A PrinterError reports a problem occurring when attempting to access the selected printer during the processing of the abstract operation in which it occurs.

```
PrinterError ::= ABSTRACT-ERROR  
    PARAMETERSEQUENCE {  
        problem    [0] PrinterProblem }
```

```
PrinterProblem ::= CHOICE {  
    standard-problem    ENUMERATED {  
        printer-error (1),  
        printer-attention (2),  
        printer-key-attention (3)},  
    extended-problem    OBJECT IDENTIFIER }
```

A PrinterProblem may be one of the following:

- a) printer-error: the printer allocated to that printer job has reported an error;
- b) printer-attention: the printer allocated to that printer job needs operator attention;
- c) printer-key-attention: the printer allocated to that printer job needs key operator attention.

#### 8.4.6 Security-error

A SecurityError reports a problem occurring when a DP-User presents credentials to a DP-Server. This may occur either when binding or when executing a DPA abstract operation bearing (proxy) credentials.

```
SecurityError ::= ABSTRACT-ERROR  
    PARAMETERSEQUENCE {  
        problem    [0] SecurityProblem }
```

```
SecurityProblem ::= CHOICE {  
    standard-problem    ENUMERATED {  
        inappropriate-authentication (1),  
        invalid-credentials (2),  
        insufficient-access-rights (3),  
        invalid-pac (4)},  
    extended-problem    OBJECT IDENTIFIER }
```

A SecurityProblem may be one of the following:



- a) *inappropriate-authentication*: The level of security associated with the requestor's credentials is inconsistent with the level of protection requested;
- b) *invalid-credentials*: the supplied credentials were invalid;
- c) *insufficient-access-rights*: The requestor does not have the right to carry out the requested abstract operation (e.g. Modify), independently of the DPA-Object involved in the abstract operation.
- d) *invalid-pac*: The supplied PAC is invalid

#### **E.4.7 Selection-error**

A *SelectionError* reports a problem encountered when attempting to select an object instance by means of a set of attributes.

```
SelectionError ::= ABSTRACT-ERROR  
  PARAMETERSEQUENCE {  
    problem [0] SelectionProblem }
```

```
SelectionProblem ::= CHOICE {  
  standard-problem ENUMERATED {  
    wrong-job-identifier (1),  
    wrong-printer-identifier (2),  
    wrong-document-sequence-number (3) },  
  extended-problem OBJECT IDENTIFIER }
```

- a) *wrong-job-identifier*: The requested job does not exist or is unknown.
- b) *wrong-printer-identifier*: The specified printer identifier is unknown or invalid.
- c) *wrong-document-sequence-number*: The specified job does not include a document with the specified document sequence number.

#### **E.4.8 Service-error**

A *ServiceError* reports a problem related to the provision of the service, which is not due to an incorrect abstract operation request or the requestor's access rights.

```
ServiceError ::= ABSTRACT-ERROR  
  PARAMETERSEQUENCE {  
    problem [0] ServiceProblem }
```

```
ServiceProblem ::= CHOICE {  
  standard-problem ENUMERATED {  
    server-busy (1),  
    server-unavailable (2),  
    operation-too-complex (3),  
    resource-limit-exceeded (4),  
    unclassified-server-error (5) },  
  extended-problem OBJECT IDENTIFIER }
```

A *ServiceProblem* reported may be one of the following:

- a) *server-busy*: The DP-Server is presently too busy to perform the requested abstract operation, but may be able to do so after a short while;

- b) `server-unavailable`: The DP-Server is currently unavailable;
- c) `operation-too-complex`: The requested abstract operation is too complex syntactically or semantically;
- d) `resource-limit-exceeded`: This may happen when a very large object is to be created or copied in the DP-Server
- e) `unclassified-server-error`: This is a place-holder for "any other" errors, primarily those due to software bugs in a yet unstable DP-Server implementation, to make it nevertheless possible to access it remotely before it becomes robust.

#### 8.4.9 Update-error

An `UpdateError` reports a problem encountered when attempting to modify (update), explicitly an existing DPA-Object. Deletion of a DPA-Object is also considered here as a modification.

```
UpdateError ::= ABSTRACT-ERROR  
              PARAMETERSEQUENCE (  
                problem                [0] UpdateProblem )
```

```
UpdateProblem ::= CHOICE (  
  standard-problem    ENUMERATED (  
    no-modifications-allowed    (1),  
    insufficient-access-rights   (2),  
    illegal-content-modification (3),  
    previous-operation-incomplete (4),  
    cancellation-not-possible    (5) ),  
  extended-problem    OBJECT IDENTIFIER )
```

An `UpdateProblem` may be one of the following:

- a) `no-modifications-allowed`: The print job referred by the `JobIdentifier` provided in the abstract operation's argument is executing and cannot be modified now.
- b) `insufficient-access-rights`: An attempt to modify a DPA-Object has been made by a DP-User with insufficient access rights to this object;
- c) `illegal-content-modification`: An attempt has been made to modify the content of a DPA-Object which is not subject to user specified modifications;
- d) `previous-operation-incomplete`: new operations are not allowed until previous finishes.
- e) `cancellation-not-possible`: cancellation not possible because the job is already completed.

#### 8.4.10 Error Precedence

Should several error conditions occur simultaneously for the same DPA abstract operation only one of them is reported to the requestor. The precedence of these error conditions is as follows beginning with the most important:

SecurityError  
ServiceError  
NameError  
AccessError  
PrinterError  
SelectionError  
DocumentAccessError  
AttributeError  
UpdateError

## 8.5 Abstract-Events

This clause defines the abstract events that may occur during the course of an abstract-operation invoked via a DP-Server port.

An event indicates that something has happened within the system. An event may be expected (e.g., the change in state of a job), or it may be an abnormal occurrence (e.g., an error). All errors are events, but not all events are errors.

Both Event Types and Event Identifiers are defined. The latter are used to identify the specific event while the former enable the user to identify the types of events for which the server is to provide notifications.

EventType ::= OBJECT IDENTIFIER

EventIdentifier ::= OBJECT IDENTIFIER

Clause 9.3.3 defines the job-event-handling attributes which are utilized to convey the event information.

### 8.5.1 Abort-event

An abort event occurs when a print job is aborted. Three event types are defined to indicate the originator of the abort.

- a) id-val-abort-event-by-server: The print job was aborted by the server.
- b) id-val-abort-event-by-operator: The print job was aborted by the operator.
- c) id-val-abort-event-by-user: The print job was aborted by the user.

### 8.5.2 Error-event

An error event occurs whenever (and in addition to an Abstract-error) an error occurs. The following error events are defined:

- a) id-val-error-event-deadline: This value specifies the event that the deadline time for the print job (established by the "job-deadline-time" parameter of the print or modify-job operation) has passed.
- b) id-val-error-event-retention: This identifier specifies the event that the retention time for the print job established by the "job-retention-time" parameter of the print or modify-job operation has passed.
- c) id-val-error-event-shutdown: This identifier specifies the event that the printer the job was printing on shut down before the job completed.
- d) id-val-error-event-noresource: This identifier specifies the event that a needed resource is not available or has become unavailable. This will include resources that are expected to become available after the deadline time established for the print job.

Resources whose time for becoming available again is 'unknown' will always be unavailable.

- c) `id-val-error-event-name-unrecognized`: This identifier specifies that a resource specified to be required for the print job is not known to the server.
- f) `id-val-error-event-nodocument`: This identifier specifies that a document specified to be printed for this Print Job is not accessible or available to the server.

### 8.5.3 Warning-event

A warning event occurs whenever a condition arises which affects the processing of the user's print job. The following warning events are defined.

- a) `id-val-warning-event-res-attr`: This identifier specifies that a resource needs the attention of person, not necessarily a key operator. Which conditions need a key operator vs. an untrained person are implementation dependent and may vary from resource to resource.
- b) `id-val-warning-event-res-operator`: This identifier specifies that a resource needs the attention of a key operator (a trained person for the particular resource). Which conditions need a key operator vs. an untrained person are implementation dependent and may vary from resource to resource.
- c) `id-val-warning-event-pr-shutdown`: This identifier specifies that the printer to be used for this job was shut down before the job was started.
- d) `id-val-warning-event-modified`: This identifier specifies the event that the print job was modified by the user or an operator using the `modify-job` operation.
- e) `id-val-warning-event-close-to-deadline`: This identifier specifies the event that the deadline time (specified by the print job parameter 'job-deadline-time') for printing the job is close. The amount of time remaining which triggers this event is implementation dependent.
- f) `id-val-warning-event-close-to-retention-time`: This identifier specifies the event that the retention time (specified by the print job parameter 'job-retention-time') is close. The amount of time remaining which triggers this event is implementation dependent.
- g) `id-val-warning-event-criteria-exceeded`: This identifier specifies the event that a previously specified counter or timer threshold value has been exceeded.

### 8.5.4 Report-event

A report event occurs whenever a significant point in the processing of the job occurs. The following report events are defined.

- a) `id-val-report-event-completed`: This identifier specifies the event that the print job is complete, and has completed successfully.
- b) `id-val-report-event-checkpoint`: This identifier specifies the event that a job checkpoint has been taken.
- c) `id-val-report-event-resource`: This identifier specifies that a message about a resource which is used by this job should be read.
- d) `id-val-report-event-discarded`: This identifier specifies that the print job was discarded because the time specified by the print job parameter 'retain-after-printing' has expired or the time specified by 'job-retention-time' has passed (the job must have printed successfully; otherwise, an error-event would occur).

- e) `id-val-report-event-file-transferred`: This identifier specifies that a file transfer has been completed. This event may be indicated by a client or third party transfer service for files transferred to a print service, or by the print service or a third party transfer service for files transferred to a print client.

#### 8.5.5 Other-event

An other event occurs whenever an event occurs which cannot be categorized by the previously specified event categories. The following other event is defined:

- a) `id-val-other-event-unknown`: This identifier specifies the occurrence of any event that cannot be categorized by the server.

#### 8.5.6 Event Identifiers

The following Event Identifiers are used in the Notification Profile or the Logging Profile to indicate the types of events the user (or designate) is to be notified.

- a) `id-val-abort-event-id`: This identifier can be used in Notification and Logging Profiles to indicate that the user wants all abort events to be notified/logged.
- b) `id-val-error-event-id`: This identifier should be used in Notification and Logging Profiles to indicate that the user wants all error events to be notified/logged.
- c) `id-val-warning-event-id`: This identifier should be used in Notification and Logging Profiles to indicate that the user wants all warning events to be notified/logged.
- d) `id-val-report-event-id`: This identifier should be used in Notification and Logging Profiles to indicate that the user wants all report events to be notified/logged.
- e) `id-val-other-event-id`: This identifier should be used in Notification and Logging Profiles to indicate that the user wants all other events to be notified/logged.



**SECTION THREE - PRINT SERVICE ATTRIBUTES**





## 9. PRINT SERVICE ATTRIBUTES

### 9.1. Overview of the Attribute Concept

An attribute is a characteristic of an object. The attribute concept, introduced in 6.4, permits the open-ended definition and use of new characteristics. By modelling argument and result parameters as attributes, implementors of print server and print client software are able to incorporate new features readily, and hence, to make available new and/or implementation-specific features for access by users in a timely fashion.

The attribute definition consists of a type, an indication of whether it has one or multiple values, and a value(s). An attribute type component carries a class of information that has meaning to the user of a print service, and serves to indicate exactly which aspect of the printing function the value component applies to. The value component consists of an interpretation and the actual value. The interpretation indicates what datatype the actual value conforms to, and may be either simple (e.g., BOOLEAN, INTEGER, ...) or structured. Attributes are defined by means of the ATTRIBUTE macro. See 6.4.

Two DPA-Attribute-Sets are defined: the DPA-Basic-Attribute-Set and the DPA-Extension-Attribute-Set. DPA supports mandatorily the DPA-Basic-Attribute-Set and, optionally, members of the DPA-Extension-Attribute-Set. Operations to the DPA-Basic-Attribute-Set are the same as for the DPA-Extension-Attribute-Set.

Table 1 shows the attributes that constitute the DPA-Basic-Attribute-Set. The remaining attributes defined in this section are all part of the DPA-Extension-Attribute-Set.

Class	Attribute name
9.3.1.1	job-identifier
9.3.1.2	job-name
9.3.1.3	job-originator
9.3.6.14	copy-count
9.3.7	document-description
9.3.7.1	transfer-method
9.3.7.2	document-format
9.3.7.3	document-sequence-number
9.3.7.4	document-content
9.3.10.1	current-job-state
9.3.10.3	printers-assigned
9.4.1	printer-identifier

Table 1 DPA-Basic-Attribute-Set

### 9.2. Generic Object attribute syntaxes

Many attributes associated with different object classes share a common form and set of values. This subclause defines a generic set of attribute syntaxes and standard values to be used in the definitions of such attributes.

#### 9.2.1. Identifier

A globally unique identifier for the object instance.

```

identifierSyntax ATTRIBUTE-SYNTAX
  OBJECT IDENTIFIER
  ::= id-syn-identifier
  
```

#### 9.2.2. Descriptor

A descriptor provides human-readable text which serves to describe an information object. The text is not an unambiguous identification of the information object, but identical text for different information objects is intended to be uncommon.

descriptorSyntax ATTRIBUTE-SYNTAX

IASString (1..ub-descriptor-string)

::= id-syn-descriptor

9.2.3 Name

The name provides a short human readable string for the associated object. This string is used for reporting to the client, which the client can in turn display to its human user. It also provides some consistency between client implementations for the communication between the client and its human user.

nameSyntax ATTRIBUTE-SYNTAX

IASString (1..ub-name-string)

::= id-syn-name

The name does not provide a unique identification of the object.

9.2.4 Aliases

Aliases provide a list of names that may be used to refer to the object.

aliasesSyntax ATTRIBUTE-SYNTAX

OBJECT IDENTIFIER

::= id-syn-aliases

Aliases may include the unique object identifier.

9.2.5 State

This attribute identifies the current state of the object.

stateSyntax ATTRIBUTE-SYNTAX

OBJECT IDENTIFIER

::= id-syn-state

These generic state values apply to all object classes except job, server and printer. State values for these objects are defined with the other Attributes that pertain to these objects.

The following standard values are defined:

Identifier	Descriptor Text
id-val-state-available	Indicates that the object is available for use without any human intervention
id-val-state-busy	Indicates that the object is being used, but will become available without human intervention
id-val-state-on-request	Indicates that the object requires human intervention in order to become available
id-val-state-unavailable	Indicates that the object is not available for use even with human intervention
id-val-state-unknown	Indicates that the state of the object is not known

9.2.6 Message

The message is a user readable string associated with an object that may be presented when the user requests this object.

**messageSyntax ATTRIBUTE-SYNTAX**

IAString (1..ub-message-string)

::= id-syn-message

This attribute may be used to indicate to users, for example, why an object is not actively available or when it is expected to be available.

**9.2.7 Availability**

This attribute identifies the current availability of an object determined by its load.

**availabilitySyntax ATTRIBUTE-SYNTAX**

OBJECT IDENTIFIER

::= id-syn-availability

This attribute may be used to indicate the load of servers, printers, etc. and it may also be used to indicate the supply of media, font, resources.

The following standard values are defined:

Identifier	Descriptor Text
id-val-availability-immediate	Indicates the object is currently not in use and is available for use immediately
id-val-availability-high	Indicates that the availability of the object is high e.g., a server is not busy, or there is plenty of the given media
id-val-availability-normal	Indicates that the availability of the object is normal e.g. the load on the server is reasonable or as expected
id-val-availability-low	Indicates that the availability of the object is low e.g., a server is busy, or a supply of media is low
id-val-availability-restricted	Indicates that the use of this object is restricted, for example because of security reasons, cost, or low supply
id-val-availability-none	Indicates that the object is not available
id-val-availability-unknown	Indicates that the availability of the object is not known

**9.2.8 List of managers**

This attribute contains the list of managers responsible for the maintenance and operation of the object.

**listOfManagersSyntax ATTRIBUTE-SYNTAX**

IAString (1..ub-name-string)

::= id-syn-list-of-managers

The "manager" indicated may not necessarily identify an individual (e.g. group of users, or a programmatic object).

**9.2.9 Text**

Text provides a human-readable text associated with a user or application. Print servers must support a maximum length of at least 127 characters.

**textSyntax ATTRIBUTE-SYNTAX**

IAString (1..ub-text-string)

::= id-syn-text

### 9.3 Job Object attributes

#### 9.3.1 Job-information

These attributes provide information to identify a print job. (e.g., for searching).

##### 9.3.1.1 Job-identifier

Provides the unique job-identifier for this job. The server-identifier is unique within the network environment. The local-identifier is a unique identifier of this job on the identified server.

```
JobIdentifier ::= SEQUENCE {  
    server-identifier OBJECT IDENTIFIER,  
    local-identifier OCTET STRING}  
  
job-identifier ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX JobIdentifier  
    SINGLE VALUE  
    ::= id-att-job-identifier
```

##### 9.3.1.2 Job-name

Supplies a human readable string for the print-job. This string is used for naming the print job in human-readable "free-form" fashion.

If this attribute is not specified, no job name is assumed, but implementation specific defaults are allowed.

```
job-name ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX nameSyntax  
    SINGLE VALUE  
    ::= id-att-job-name
```

##### 9.3.1.3 Job-originator

Supplies a human-readable name for the originator of the print job.

```
job-originator ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX textSyntax  
    SINGLE VALUE  
    ::= id-att-job-originator
```

If this attribute is not specified, no value for job originator is assumed, but implementation specific defaults are allowed.

##### 9.3.1.4 Job-owner

Supplies the name of a human or programmatic owner of the print job.

```
job-owner ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX textSyntax  
    SINGLE VALUE  
    ::= id-att-job-owner
```

The value of job-owner will often be the same as job-originator. The job-owner will be different from job-originator when the job has been submitted by the originator on behalf of

the owner. This is particularly useful when an operator must resubmit a misprinted job on behalf of the original job owner. This is not to take the place of the security attributes.

If this attribute is not specified, the value of job-originator should be used for any circumstances which require a value for job-owner.

#### 9.3.1.5 Job-comment

Supplies an arbitrary human-readable text string associated with the print job.

job-comment ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-at-job-comment

#### 9.3.1.6 Job-start-message

Supplies a human-readable message string associated with the print job.

job-start-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-at-job-start-message

This message is forwarded to the operator (in an implementation dependent manner) before processing starts.

#### 9.3.1.7 Job-end-message

Supplies a human-readable message string associated with the print job.

job-end-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-at-job-end-message

This message is forwarded to the operator (in an implementation dependent manner) after machine processing ends.

### 9.3.2 Job-results-handling

These attributes specify the actions to be undertaken after a print job has been completed.

#### 9.3.2.1 Results-profile

The ResultsProfile includes the specification of a delivery method for delivery of the print output, specification of the address to which the results should be delivered and a human readable descriptor representing the intended recipient of the results.

ResultsProfile ::= SEQUENCE {  
  delivery-method OBJECT IDENTIFIER,  
  delivery-descriptor IA5String (1..ub-text-string),  
  delivery-address DeliveryAddress}

results-profile ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX ResultsProfile  
SINGLE VALUE  
::= id-at-results-profile

Values for 'delivery-method' are defined with the Delivery Method Object attributes, in 9.10.

The omission of a value for delivery-method indicates the default delivery method of a particular implementation is to be used (e.g., pick-up). Some of the values indicated may require other attributes to become mandatory.

Delivery-descriptor is a human-readable string supplied for clarification to the human user. It is not intended to be used as part of the address for delivery of results.

If delivery-descriptor is not specified, no name is assumed, but implementation specific defaults are allowed.

```
DeliveryAddress ::= CHOICE {  
    ORAddressAndOrDirectoryName,  
    DistinguishedName,  
    IASString (1..ub-text-string)}
```

*NOTE 16*

*ORAddress is imported from ISO/IEC 10021-4, and DistinguishedName is imported from ISO/IEC 9594-2.*

Delivery-address is programmatically interpretable data that fully specifies the address of the recipient of results. The structure and content of delivery-address is specific to the delivery-method.

For example, delivery-address can be used as a postal mail address, printed on the start sheet and/or end sheet, etc. If no value is specified for delivery-address, no address is assumed, but implementation specific defaults are allowed.

### 9.3.3 Job-event-handling

These attributes provide information for the actions to be undertaken when significant events occur during the processing of the print job.

#### 9.3.3.1 Notification-profile

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

```
NotificationProfile ::= SEQUENCE {  
    event-identifiers SET OF EventIdentifier,  
    delivery-method OBJECT IDENTIFIER,  
    delivery-descriptor IASString (1..ub-text-string),  
    delivery-address DeliveryAddress}
```

notification-profile ATTRIBUTE

WITH ATTRIBUTE-SYNTAX NotificationProfile

SINGLE VALUE

::= id-all-notification-profile

Event-identifiers is a list of identifiers of events of interest that may occur during the processing of the print job. This attribute must always be provided as a list element of "notification-profile".

Standard values for delivery-method are defined with the Delivery Method Object attributes, in 9.10.

The omission of *delivery-method* indicates the standard delivery method of a particular implementation is to be used (e.g., electronic mail). Some of the values indicated may require other attributes to become mandatory.

*Delivery-descriptor* is a human-readable string supplied for clarification to the human user. It is not intended to be used as part of the address for delivery of notifications.

*Delivery-address* is programmatically interpretable data that fully specifies the address of the recipient of notifications. The structure and content of *delivery-address* is specific to the *delivery-method*.

### 9.3.3.2 Logging-profile

This attribute is a specification of events which are to be logged. In addition, this attribute specifies how the logs are to be delivered.

```
LoggingProfile ::= SEQUENCE {  
    event-identifiers SET OF EventIdentifier,  
    delivery-method OBJECT IDENTIFIER,  
    delivery-descriptor IAString (1..ub-text-string),  
    delivery-address DeliveryAddress}
```

```
logging-profile ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX LoggingProfile  
    SINGLE VALUE  
    ::= id-att-logging-profile
```

*Event-identifiers* is a list of identifiers of events of interest that may occur during the processing of the print job. This attribute must always be provided as a list element of "logging-profile".

Standard values for *delivery-method* are defined with the *Delivery Method Object* attributes in 9.10.

The omission of *delivery-method* indicates the standard delivery method of a particular implementation is to be used (e.g., pick-up). Some of the values indicated may require other attributes to become mandatory.

*Delivery-descriptor* is a human-readable string supplied for clarification to the human user. It is not intended to be used as part of the address for delivery of logs.

*Delivery-address* is programmatically interpretable data that fully specifies the address of the recipient of logs. The structure and content of *delivery-address* is specific to the *delivery-method*.

### 9.3.3.3 Job-abort-criteria

Specifies a list of limit specifications which, when exceeded, force the abort of the print job.

```
Criteria ::= SEQUENCE {  
    counter-identifier AttributeType,  
    counter-threshold Threshold}
```

```
Threshold ::= CHOICE {INTEGER (1..ub-integer), UTCTime}
```

job-abort-criteria ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX Criteria  
MULTI VALUE  
::= id-att-job-abort-criteria

Counter-identifier specifies an attribute associated with the processing of a print job, which has a type of INTEGER or UTCTime for the value component. For example, the identified attribute value could be DeltaTime or ExpirationTime or a counter of event occurrences.

Counter-threshold provides a value to be used by the server in determining when to abort a print job; if the actual value of the identified attribute exceeds the specified threshold value, the job is aborted.

#### 9.3.3.4 Job-warning-criteria

Supplies a list of limit specifications which, when exceeded the first time, cause a warning to be created for the print job.

job-warning-criteria ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX Criteria  
MULTI VALUE  
::= id-att-job-warning-criteria

Counter-threshold provides a value to be used by the server in determining when to issue a warning; if the actual value of the identified attribute exceeds the specified threshold value, the warning is issued.

#### 9.3.4 Job-scheduling-instructions

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

##### 9.3.4.1 Job-start-wait

This attribute specifies that processing of the print job is to await operator action before commencing.

job-start-wait ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX Boolean  
SINGLE VALUE  
::= id-att-job-start-wait

The value 'TRUE' is defined to cause a printer to pause at the start of printing this job. The job may continue when a user 'resumes' the job. The value 'FALSE' is defined to mean that a job will not pause at the start of printing.

A start-wait is expected to occur once the job is scheduled to be printed on a specific printer. This wait allows operator intervention on behalf of the job (e.g., for changing media and other resources). The "job-start-message" may be sent to the operator at this time to request the necessary intervention.

##### NOTE 17

*If it desired that job-start-wait not significantly impede the progress of other jobs; i.e., a print server should be able to time out and pause a waiting job if action is not taken. This is an implementation-specific issue.*



#### 9.3.4.2 Job-end-wait

This attribute specifies that processing of the print job is to await operator action before concluding.

job-end-wait ATTRIBUTE

WITH ATTRIBUTE-SYNTAX Boolean

SINGLE VALUE

:= id-att-job-end-wait

The value 'TRUE' is defined to cause a printer to pause at the end of printing this job. The job may continue when a user or operator 'resumes' the job. The value 'FALSE' is defined to mean that a job will not pause at the end of printing.

A job-end-wait is expected to occur once the job has completed printing. This wait allows operator intervention on behalf of the job (e.g., for changing media and other resources). The "job-end-message" may be sent to the operator at this time to request the necessary intervention.

##### NOTE 18

*It is desired that job-end-wait not significantly impede the progress of other jobs; i.e., a print server should be able to time out and complete a waiting job if action is not taken. This is an implementation-specific issue.*

#### 9.3.4.3 Job-hold

This attribute specifies that the print job is not to be released for printing until the "job-hold" attribute has been reset. "Pause" and "Resume" can be used in combination with this attribute.

job-hold ATTRIBUTE

WITH ATTRIBUTE-SYNTAX Boolean

SINGLE VALUE

:= id-att-job-hold

The value 'TRUE' is defined to cause a print job to not be released for printing. The value 'FALSE' is defined to mean that a job is not required to be held from printing.

The resetting of the "job-hold" attribute is performed by the modify-job operation.

#### 9.3.4.4 Job-password

This attribute provides a password that must be entered before the print job commences.

job-password ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OCTET STRING

SINGLE VALUE

:= id-att-job-password

If a "job-password" is provided processing of the job on a particular printer waits for the password to be re-entered.

##### NOTE 19

*This is independent of start-wait and will occur whether or not the start-wait attribute is set. It is desired that a job-password wait not significantly impede the progress of other jobs; i.e., a print server should be able to time out and pause a waiting job if action is not taken. This is an implementation-specific issue.*

*NOTE 20*

*The encryption scheme to be used for this attribute is defined in the application-context.*

**9.3.4.5 Job-print-after**

This attribute specifies the time after which the print job may be scheduled for printing.

job-print-after ATTRIBUTE

WITH ATTRIBUTE-SYNTAX UTCTime

SINGLE VALUE

::= id-att-job-print-after

**9.3.4.6 Job-priority**

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

job-priority ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (1..100)

SINGLE VALUE

::= id-att-job-priority

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

**9.3.4.7 Job-deadline-time**

This attribute specifies the time by which the user desires the print job to be completed.

job-deadline-time ATTRIBUTE

WITH ATTRIBUTE-SYNTAX UTCTime

SINGLE VALUE

::= id-att-job-deadline-time

The print output may or may not include logs and other associated information. Support (or no-support) of deadline scheduling is implementation-dependent.

**9.3.4.8 Job-retention-time**

This attribute specifies the time at which the print job should be discarded, whether or not the job has printed.

job-retention-time ATTRIBUTE

WITH ATTRIBUTE-SYNTAX UTCTime

SINGLE VALUE

::= id-att-job-retention-time

If this attribute is supported, the server is required to retain job attributes and status. The server is not required to retain job document files if the job has completed processing. Whether or not the print job is discarded if the retention-time event occurs while the job is being printed is implementation dependent.

### 9.3.4.9 Job-retention-period

This attribute specifies the period of time for which the print job should be retained after the printing and processing of the job are completed.

job-retention-period ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX DateTime  
SINGLE VALUE  
::= id-att-job-retention-period

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

If this attribute is supported, the server is only required to retain job attributes and status. The server is not required to retain job document files.

### 9.3.4.10 Job-message-to-operator

This attribute indicates a message from the user to the operator to indicate something about the processing of this print job. This message - unlike the job-start-message - is not necessarily related to other job attributes.

job-message-to-operator ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-job-message-to-operator

## 9.3.5 Printer-identification

These attributes provide information to help select a particular printer. Definitions for the printer-identification attributes are explicitly those given for Attributes of the Printer Object Class (see 9.4).

## 9.3.6 Production-instructions

These attributes provide information that affect the rendering and finishing of the document. This information may also be contained in the document to be printed, in which case the attributes are referred to as document production instructions.

### 9.3.6.1 Font-substitution

This attribute specifies a list of pairs of font references (see ISO/IEC 9541). Each of these pairs identifies a document font, followed by the font to be substituted for it.

FontSubstitution ::= SEQUENCE {  
    original-font FontReference,  
    substitution-font FontReference }

font-substitution ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX FontSubstitution  
MULTI VALUE  
::= id-att-font-substitution

Omission of this attribute does not imply that no substitution will occur.

### 9.3.6.2 Media-substitution

This attribute specifies a list of pairs of media identifiers. Each of these pairs identifies a document medium, followed by the medium to be substituted for it.

MediaSubstitution ::= SEQUENCE (  
    original-medium MediumIdentifier,  
    substitution-medium MediumIdentifier )

media-substitution ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX MediaSubstitution  
    MULTI VALUE  
    ::= id-att-media-substitution

Standard values are defined in 9.6.1.

Omission of this attribute does not imply that no substitution will occur.

### 9.3.6.3 Named-colour-substitution

This attribute specifies a list of pairs of named colour identifiers. Each of these pairs identifies a document named-colour, followed by the named-colour to be substituted for it.

NamedColourSubstitution ::= SEQUENCE (  
    original-colour OBJECT IDENTIFIER,  
    substitution-colour OBJECT IDENTIFIER )

named-colour-substitution ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX NamedColourSubstitution  
    MULTI VALUE  
    ::= id-att-named-colour-substitution

Standard values for this attribute are not defined in this standard.

Omission of this attribute does not imply that no substitution will occur.

### 9.3.6.4 Finishing-specification

This attribute identifies user-requested finishing options to be applied to the printed output.

finishing-specification ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
    MULTI VALUE  
    ::= id-att-finishing-specification

Standard values for this attribute are defined in 9.12.1.

The finished printed output should not include logs and other accompanying material. Some or all of the specified finishing actions may be performed by a human instead of a machine. Omission does not exclude some form of site-defined finishing.

### 9.3.6.5 Output-specification

This attribute identifies the output processing for the media on which the job is to be printed.

output-specification ATTRIBUTE  
 WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
 MULTI VALUE  
 ::= id-att-output-specification

Standard values for this attribute are defined in 9.13.1.

9.3.6.6 Output-bin

This attribute specifies the output receptacle for the media on which the job is to be printed.

Outputbin ::= CHOICE (  
 bin-number [0] INTEGER (1..ub-integer),  
 bin-name [1] OBJECT IDENTIFIER )

output-bin ATTRIBUTE  
 WITH ATTRIBUTE-SYNTAX Outputbin  
 SINGLE VALUE  
 ::= id-att-output-bin

Omission of this attribute indicates that any receptacle can be used.

Standard values:

id-val-output-bin-top	id-val-output-bin-middle
id-val-output-bin-bottom	id-val-output-bin-upper
id-val-output-bin-lower	id-val-output-bin-side
id-val-output-bin-face-up	id-val-output-bin-large-capacity
id-val-output-bin-private	id-val-output-bin-secure

Available output bins can be associated with these standard values by the server. This mapping is, however, site-specific and outside the scope of this standard.

9.3.6.7 Imposition-specification

This attribute identifies the convenience imposition functions, which affect the placement of a page image on a sheet.

imposition-specification ATTRIBUTE  
 WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
 MULTI VALUE  
 ::= id-att-imposition-specification

Standard values are defined in 9.14.1.

The list of convenience imposition functions given as a value of this attribute need not be mutually exclusive; any interdependencies between these convenience imposition functions are outside the scope of this Standard.

However, the imposition will be dependent on the values for content-orientation.

### 9.3.6.8 Print-quality

This attribute specifies the desired output quality of the printed document.

print-quality ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-at-print-quality

Standard values

Identifier	Descriptor Text
id-val-print-quality-draft	Lowest quality available on this printer
id-val-print-quality-normal	Normal or intermediate quality on this printer
id-val-print-quality-high	Highest quality available on this printer

Some printers have programmatically controlled output quality. This attribute allows the user to specify the level of output quality desired from these printers.

### 9.3.6.9 Font-fidelity

This attribute specifies how closely the available font resources must adhere to the requested font resources. Font specifications may occur both in the document attributes and in the production instructions.

font-fidelity ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-at-font-fidelity

Standard values are to be defined in ISO/IEC 9541.

### 9.3.6.10 Media-fidelity

This attribute specifies how closely the available media must adhere to the requested media. Media may be specified both in the document attributes and in the production instructions.

Standard values

media-fidelity ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-at-media-fidelity

Standard values

Identifier	Descriptor Text
id-val-media-fidelity-absolute	This value indicates that only the exact media as specified in the document or in the media-substitution attribute are acceptable for the user
id-val-media-fidelity-colour	This value indicates that an appropriate medium may be substituted if this medium has the same value of the attribute 'medium-colour'

## Standard values (continued)

Identifier	Descriptor Text
id-val-media-fidelity-size	This value indicates that an appropriate medium may be substituted by the server if this medium has the same value for the attribute 'medium-size' or 'medium-dimensions'.
id-val-media-fidelity-reproduction-area	This value indicates that an appropriate medium may be substituted by the server if the value for 'medium-size', 'medium-dimension' or 'medium-reproduction-area' of the substituting medium is equal to or larger than the value for the corresponding attribute of the requested medium.
id-val-media-fidelity-any	This value indicates that an appropriate medium may be substituted by the server.

## 9.3.6.11 Colour-fidelity

This attribute identifies how closely the available named-colours must adhere to the requested named-colours. Named-colours may be specified both in the document attributes and in the production instructions.

## colour-fidelity ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-att-colour-fidelity

## Standard values

Identifier	Descriptor Text
id-val-colour-fidelity-absolute	This value indicates that only the exact named-colours as specified in the document or by the attribute named-colour-substitution are acceptable for the user.
id-val-colour-fidelity-colour	This value indicates that the server may substitute a named-colour with another available named-colour, provided that this named-colour has the same approximate colour (e.g. 'red', 'white', 'blue').
id-val-colour-fidelity-distinct	This value indicates that the server may substitute named-colours, provided all different named-colours remain distinct.
id-val-colour-fidelity-any	This value indicates that the server may substitute any named-colour with any other named-colour.

## 9.3.6.12 Sides

This attribute specifies the number of printable surfaces of the medium to be imaged.

sides ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (1..2)

SINGLE VALUE

::= id-att-sides

Depending on the capabilities of the associated (logical) printer a system administrator defined default will be used.

9.3.6.13 Page-select

This attribute specifies one or more sequences of pages to be printed.

If the beginning page is omitted, the first page of the document is assumed. If the end page is omitted, the last page of the document is assumed.

For each page range specified, if beginning-page and end-page exist, and beginning page is not after end-page, the pages from the former to the latter are printed, inclusive. Sequences shall not overlap. If these conditions are not met, an error is returned.

PageIdentifier ::= CHOICE (  
  nominal-page-number    INTEGER,  
  alphanumeric-page-number  ASString (1..ub-name-string) )

PageSelect ::= SEQUENCE (  
  beginning-page    [0]PageIdentifier OPTIONAL, -- omitted means no lower bound --  
  ending-page      [1]PageIdentifier OPTIONAL -- omitted means no upper bound -- )

page-select ATTRIBUTE

WITH ATTRIBUTE-SYNTAX PageSelect

MULTI VALUE

::= id-att-page-select

The nominal-page-number is the in-sequence number in the document in the order that the pages are presented to the server.

The alphanumeric-page-number is a page-number that can be recognized in the page identifiers of the print format. Note that this is not possible for most print formats.

9.3.6.14 Page-media-select

This attribute indicates that specified pages shall be printed on identified media. For the specified pages, this shall override any media specified in the document, and any media-substitutions, performed subsequently. Any pages not specified shall be printed on the default media in the document context if any, or defaulted by the server. Both logical and absolute media can be selected by this attribute.

PageMediaSelect ::= SEQUENCE (  
  page-range    [0]PageSelect,  
  medium-id    [1]MediumIdentifier )

page-media-select ATTRIBUTE

WITH ATTRIBUTE-SYNTAX PageMediaSelect

MULTI VALUE

::= id-att-page-media-select



**9.3.6.15 Copy-count**

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

copy-count ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
DEFAULT 1  
SINGLE VALUE  
::= id-att-copy-count

A value of '1' for "copy-count" will generate a single human perceptible copy of the electronic document.

**9.3.6.16 Start-sheet**

This attribute identifies a start sheet template to be used by the print server to generate a job start sheet.

start-sheet ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-start-sheet

Standard values:

id-val-job-sheet-none	id-val-job-sheet-brief
id-val-job-sheet-full	

The omission of this attribute indicates the default start sheet of a particular implementation is to be selected.

**9.3.6.17 Separator-sheet**

This attribute identifies a separator sheet template to be used by the print server to generate a job separator sheet.

separator-sheet ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-separator-sheet

Standard values:

id-val-job-sheet-none	id-val-job-sheet-brief
id-val-job-sheet-full	

The omission of this attribute indicates the default separator sheet of a particular implementation is to be selected.

**9.3.6.18 End-sheet**

This attribute identifies an end sheet template to be used by the print server to generate a job end sheet.

**end-sheet ATTRIBUTE**

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-att-end-sheet

Standard values:

id-val-job-sheet-none	id-val-job-sheet-brief
id-val-job-sheet-full	

The omission of this attribute indicates the default end sheet of a particular implementation is to be selected.

**9.3.6.19 Reset-printer**

This attribute specifies that the interpreter and/or printer be reset after processing this document (in a multiple document job).

**reset-printer ATTRIBUTE**

WITH ATTRIBUTE-SYNTAX BOOLEAN

DEFAULT TRUE

SINGLE VALUE

::= id-att-reset-printer

This attribute would normally be used to suppress the resetting of non-page-independent interpreters and/or printers so that the previously defined state (e.g. font resources, forms, etc.) are inherited by the next document in the job. In addition, some interpreters and/or printers require explicit resetting before printing of other documents in the job.

This attribute has no meaning or effect for print formats that are page-independent, such as SPDL.

A printer is always reset after the last document in a job, independent of whether reset-printer is TRUE or FALSE for the last document.

**9.3.7 Document-description**

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

**9.3.7.1 Transfer-method**

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

**transfer-method ATTRIBUTE**

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-att-transfer-method

Standard values are defined in 9.9.1.

**NOTE 2)**

Four paradigms are identified by the standard values, but there may be many (implementation specific) transfer methods and corresponding protocols. A standardized example is Referenced

*Data Transfer. Conforming implementations must support at least transfer-document-with-job-request.*

### 9.3.7.2 Document-format

This attribute identifies the overall print format used for the document.

```
document-format ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  SINGLE VALUE
  ::= id-att-document-format
```

Standard values for this attribute are to be defined by the applicable print format standards.

### 9.3.7.3 Document-sequence-number

This attribute specifies the number of this document in relation to the set of documents in this job. The first integer is the ordinal number of this document with 1 indicating the first document in the job. The second number indicates the total number of documents in the job. If this attribute is omitted, the meaning is a single-document job.

Documents are assumed to be submitted in order (i.e., document number 1 followed by document number 2, etc.). The semantics associated with the submission of documents in any other order are outside the scope of this standard.

When the total number of documents is unknown, the individual document number should be supplied, but the total number of documents should be set to 0. When the total number of documents is discovered, the total should then be supplied. No action will be taken by the server on the job until the total number of documents is known by the server and until all of these documents have been submitted. When the total number of documents is known at the time the last document is submitted, the document number will be the same as the total number of documents. When the total number of documents is not known until after the last document has been submitted, the print operation is invoked with document number one greater than the specified total number. The print server will only use this invocation of the print operation to indicate that all the documents have been submitted. All other attributes passed to the operation will be ignored.

```
SequenceNumber ::= SEQUENCE {
  sequence-number-in-set  INTEGER (1..ub-integer) DEFAULT 1,
  total-documents-in-set  INTEGER (1..ub-integer) DEFAULT 1 }
```

```
document-sequence-number ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SequenceNumber
  SINGLE VALUE
  ::= id-att-document-sequence-number
```

### 9.3.7.4 Document-content

This attribute specifies a transfer-method-specific reference for the document to be transferred.

```
DocumentContent ::= CHOICE {
  included-document  EXTERNAL,
  referenced-document RDT-reference }
```

The RDT-reference data type is imported from ISO/IEC 10021-2. The RDT-reference data type is also used for FTAM references.

document-content ATTRIBUTE

WITH ATTRIBUTE-SYNTAX DocumentContent

SINGLE VALUE

::= id-att-document-content

9.3.7.5 File-reference

This attribute specifies a file reference in a file store (or a transfer-method-specific reference). It is returned by the server as part of the PrintResult, if the document transfer method of the job required the client to perform the transfer (e.g. FTAM-by-client).

file-reference ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OCTET STRING

SINGLE VALUE

::= id-att-file-reference

9.3.8 Document-attributes

This group of attributes describes the characteristics of the document to be printed.

The values provided by these attributes are intended to assist the print server in validating and scheduling the print job. Providing these attributes independent of the document allows the server to validate the resources required to print the document without interpreting the contents of the document. This provides the opportunity for a server to support a broad set of print formats yet still support fast efficient validation of each job.

Note that these attributes are considered informational by the DP-Server; an error is not declared if the DP-Server finds any of these attributes to be invalid or incorrect.

9.3.8.1 Document-descriptor

Supplies a human readable string for the document. This string allows the client to indicate the source of the document in human-readable "free-form" fashion.

document-descriptor ATTRIBUTE

WITH ATTRIBUTE-SYNTAX descriptorSyntax

SINGLE VALUE

::= id-att-document-descriptor

If this attribute is not specified, no descriptor is assumed, but implementation specific defaults are allowed. It is suggested, however, that the server not supply a value for this attribute when printing the value (e.g. on a start sheet). This string only has meaning to the clients and can therefore take several forms, e.g. the name of a mail folder, name of a revisable document, a file specification, etc. It may be misleading if the server supplied a default value for this attribute.

9.3.8.2 Document-authors

Supplies a human-readable name for the authors of the document.

document-authors ATTRIBUTE

WITH ATTRIBUTE-SYNTAX textSyntax

MULTI VALUE

::= id-att-document-authors

If this attribute is not specified, no author name is assumed, but implementation specific defaults are allowed.

**9.3.8.3 Document-message**

Supplies an arbitrary human-readable message associated with the document.

document-message ATTRIBUTE

WITH ATTRIBUTE-SYNTAX messageSyntax

SINGLE VALUE

::= id-att-document-message

**9.3.8.4 Document-revision-date**

This attribute specifies the date and time the document was created or modified.

document-revision-date ATTRIBUTE

WITH ATTRIBUTE-SYNTAX UTCTime

SINGLE VALUE

::= id-att-document-revision-date

The date given in this attribute refers to the latest revision of the document.

**9.3.8.5 Content-formats-used**

This attribute identifies the content print formats that may be encountered during execution of the job. The purpose of this attribute is to permit a print client to supply supplementary information that may aid a print server in selecting an appropriate printer to process a job.

content-formats-used ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

MULTI VALUE

::= id-att-content-formats-used

The following standard values are used

Identifier	Descriptor Text
id-val-content-format-text	The document contains text
id-val-content-format-graphics	The document contains synthetic graphics, e.g. line drawings
id-val-content-format-images	The document contains raster image data, e.g. input from a scanner, fax system, etc.
id-val-content-format-unknown	The specific content formats contained by the document to be printed are not known by the client, or the base document may include references to other unknown documents

Note that specific meanings to be ascribed to the foregoing values are highly dependent upon the print server implementation and upon the print format of the base document.

**9.3.8.6 Assured-reproduction-area**

This attribute indicates the most significant area required to image the content.

AssuredReproductionArea ::= SEQUENCE

(INTEGER (1..ub-integer), INTEGER (1..ub-integer))

## assured-reproduction-area ATTRIBUTE

WITH ATTRIBUTE-SYNTAX AssuredReproductionArea

SINGLE VALUE ::= id-att-assured-reproduction-area

The first value represents the size of the shortest edge, and the second attribute the size of the longest edge of the assured reproduction area. The measure of the attributes are millimetres. The assured-reproduction-area is independent of the content-orientation as specified in the print format or the attribute content-orientation.

## 9.3.8.7 Page-size

This attribute identifies the page size of the pages the document was formatted for and is used by the imposition process. It is also used by simple formatters that format simple text in the printing system. Possible values for this attribute are the same as those possible for medium-size.

## page-size ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-att-page-size

If page size is not supplied, its value is implied by the value supplied for the medium-used value.

## 9.3.8.8 Content-orientation

This attribute identifies the most significant orientation of the content of the document.

## content-orientation ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-att-content-orientation

The following standard values are defined:

Identifier	Descriptor Text
id-val-content-orientation-portrait	The most significant orientation of the document is with the sides longer than the top when the page is held in the intended human reading orientation
id-val-content-orientation-landscape	The most significant orientation of the document is with the sides shorter than the top when the page is held in the intended human readable orientation In order to simplify binding and other finishing operations, landscape is defined to be a rotation of the page by +90 degrees (i.e. counter-clockwise) from the portrait orientation
id-val-content-orientation-mixed	The document uses both "PORTRAIT" and "LANDSCAPE" orientations. It also implies that the document content is capable of indicating the actual initial and subsequent orientations

The value of content-orientation is a scheduling and processing hint. A server may use this value to assign this job to a printer which supports the specified orientation. A server may

use this value to set the appropriate internal state of the server (including but not limited to putting a printer in the appropriate mode). Use of this attribute to set the internal state of a server is not necessary (nor advisable) for all servers and/or all document formats.

### 9.3.8.9 Page-order

This attribute indicates the order of the pages in the document, specifically, with respect to the order in which they are ordinarily viewed (read) by the user.

page-order ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

DEFAULT id-val-page-order-forward

SINGLE VALUE

::= id-att-page-order

The following standard values are defined:

Identifier	Descriptive Text
id-val-page-order-forward	from 1 to n
id-val-page-order-reverse	from n to 1
id-val-page-order-signature	a particular order on a single sheet

### 9.3.8.10 Page-independent

This attribute indicates whether or not the document is page-independent.

page-independent ATTRIBUTE

WITH ATTRIBUTE-SYNTAX BOOLEAN

DEFAULT TRUE

SINGLE VALUE

::= id-att-page-independent

This attribute should be set to 'FALSE' if the document or any portion within the document is not such that its pages are fully independently defined without relying on context established within previous pages. This attribute should be set to 'TRUE' if the document is entirely page-independent.

### 9.3.8.11 Character-sets-used

This attribute identifies the character encoding used in the document.

character-sets-used ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

MULTI VALUE

::= id-att-character-sets-used

Where the document does not indicate which character set is to be used initially, the first character set in the list provided by this attribute will be used until the document description specifies otherwise.

#### NOTE 22

Depending on the proposed work on character mapping and fonts this attribute may not be necessary.

#### 9.3.8.12 Character-mappings-used

This attribute identifies the mappings used to associate character encoding to character identifiers.

character-mappings-used ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-ati-character-mappings-used

Where the document does not indicate which character mapping is to be used initially, the first character mapping in the list provided by this attribute will be used until the document description specifies otherwise.

#### 9.3.8.13 Fonts-used

This attribute identifies the font resources that the DPA client expects are required by the document.

fonts-used ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX FontReference  
MULTI VALUE  
::= id-at-fonts-used

Where the document does not indicate which font resource is to be used initially, the first font reference in the list provided by this attribute will be used until the document description specifies otherwise. The font reference data type is defined in ISO/IEC 9541.

#### 9.3.8.14 Character-repertoires-used

This attribute identifies the character repertoires specified in the document in accordance with ISO/IEC 9541.

character-repertoires-used ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-ati-character-repertoires-used

Where the document does not indicate which character repertoire is to be used initially, the first character repertoire in the list provided by this attribute will be used until the document description specifies otherwise. A Character-Repertoire-Reference is constructed according to ISO/IEC 9541.

*NOTE 2)*

*According to ISO/IEC 9541 this is distinct from character sets as used by encoding standards. Depending on the proposed work on character mapping and fonts use of this attribute may not be necessary.*

#### 9.3.8.15 Media-used

This attribute identifies the media specified in the document.

media-used ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX MediumIdentifier  
MULTI VALUE  
::= id-at-media-used

Standard values are defined in 9.6.1.



Where the document does not indicate which medium is to be used initially, the first medium in the list provided by this attribute will be used until the document description specifies otherwise.

**9.3.8.16 Named-colours-used**

This attribute identifies the named-colours specified in the document.

named-colours-used ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-att-named-colours-used

Where the document does not indicate which named colour is to be used initially, the first named colour in the list provided by this attribute will be used until the document description specifies otherwise.

**9.3.8.17 System-hints**

These attributes provide data that may be used to estimate the complexity of the print job. If a system-hint-value is incorrect, the action taken by a server is outside the scope of this standard (e.g., when the complexity of the job being processed surpasses the value given).

**9.3.8.17.1 Octet-count**

This attribute specifies the size of the document in octets.

octet-count ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)

SINGLE VALUE

::= id-att-octet-count

**9.3.8.17.2 Page-count**

This attribute specifies the size of the document in pages.

page-count ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)

SINGLE VALUE

::= id-att-page-count

**9.3.8.17.3 Glyph-count**

This attribute specifies the size of the document in glyphs.

glyph-count ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)

SINGLE VALUE

::= id-att-glyph-count

**9.3.8.17.4 Font-count**

This attribute specifies the number of different fonts used in the document.

font-count ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)

SINGLE VALUE

::= id-att-font-count

Note that since this is only an indication of the potential effect on system performance, it is implementation-specific as to whether the value of this parameter indicates the number of different font resources or the number of different font objects involved.

**9.3.8.17.5 Font-change-count**

This attribute specifies the number of font changes in the document.

font-change-count ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
::= id-att-font-change-count

Note that since this is only an indication of the potential effect on system performance, it is implementation-specific as to whether the value of this parameter indicates the number of different font resources or the number of different font objects involved.

**9.3.8.17.6 Maximum-fonts-per-page**

This attribute specifies the maximum number of different fonts used on a page of the document.

maximum-fonts-per-page ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
::= id-att-maximum-fonts-per-page

Note that since this is only an indication of the potential effect on system performance, it is implementation-specific as to whether the value of this parameter indicates the number of different font resources or the number of different font objects involved.

**9.3.8.17.7 Percent-graphics**

This attribute indicates the percent of the document which is graphics.

percent-graphics ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..100)  
SINGLE VALUE  
::= id-att-percent-graphics

**9.3.8.17.8 Percent-images**

This attribute indicates the percent of the document which is images.

percent-images ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..100)  
SINGLE VALUE  
::= id-att-percent-images

**9.3.9 Access-and-accounting**

These attributes specify access control and accounting information used during the processing of the print job. This information is directly related to the print job and is used for authorization and accounting rather than authentication.

**9.3.9.1 User-name**

This attribute specifies the name of the user requesting access to print service operations.

**user-name ATTRIBUTE**

WITH ATTRIBUTE-SYNTAX *nameSyntax*

SINGLE VALUE

::= id-at-user-name

The usage of this attribute may be mandatory, depending upon the local security policy used.

**9.3.9.2 Accounting-Information**

This attribute specifies information required by accounting services (e.g. the account to be charged for any services rendered).

**accounting-information ATTRIBUTE**

WITH ATTRIBUTE-SYNTAX OCTET STRING

SINGLE VALUE

::= id-at-accounting-information

Accounting information is intended to be interpreted by an accounting system, and may be opaque to the print service.

**9.3.10 Job-status**

These attributes specify the job status during and after the processing of the print job.

**9.3.10.1 Current-job-state**

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

**current-job-state ATTRIBUTE**

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-at-current-job-state

The following standard values are defined (first part):

Identifier	Descriptor Text
id-val-job-state-unknown	The job state is not known, or is indeterminate, or is not returned by the operation
id-val-job-state-pre-processing	The job is being checked by the server for attributes, defaults being applied, a printer being selected, etc
id-val-job-state-document-transfer-pending	The job has been accepted by the server and is waiting for its files to start being transferred
id-val-job-state-transferring	The job files are being transferred to the server, the job has been accepted
id-val-job-state-pending	The job is waiting for a printer
id-val-job-state-processing	The job is being processed by the server; it will typically be printing on a printer, but may be undergoing start-wait, end-wait, etc., and the detailed status is represented in the printer state

Standard values (second part):

Identifier	Descriptor Text
id-val-job-state-interrupted	The job was interrupted by an intervening job on the assigned printer, and will resume processing automatically once the intervening job has completed
id-val-job-state-logfile-transfer-pending	The job has been processed and the log file is scheduled for returning to the client
id-val-job-state-logfile-transferring	The job has been completed and the log file is being transferred to the client
id-val-job-state-retained	The job has completed successfully or with warnings or errors, been aborted while printing, or cancelled before printing
id-val-job-state-hold	The job is waiting for the job-hold condition to be released
id-val-job-state-paused	The job has been paused as a result of a pause operation on the job
id-val-job-state-aborting	The job has been cancelled and is in the process of terminating

### 9.3.10.2 Previous-job-state

This attribute identifies the state of the job (pending, printing, held, etc.) before the last change of job state.

previous-job-state ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-all-previous-job-state

Standard values are defined in 9.3.10.1.

The previous-job-state should be the same as current-job-state following a list-object-attributes operation on the job. However, previous-job-state provides valuable information following cancel-job and modify-job operations.

### 9.3.10.3 Printers-assigned

This attribute identifies the printers to which this job has been assigned. In addition it provides the current state of each printer.

PrinterAssigned ::= SEQUENCE {

printer-identifier [0] OBJECT IDENTIFIER,

printer-state [1] OBJECT IDENTIFIER)

printers-assigned ATTRIBUTE

WITH ATTRIBUTE-SYNTAX PrinterAssigned

MULTI VALUE

::= id-all-printers-assigned

A set with no elements will be returned if this attribute is supported, but no printers have been assigned for this job. These printers might not be the same as the printer requested by

the user if the job has been re-assigned by an operator to ensure successful completion of the job. This attribute allows the user to find out where a job has been re-assigned (when necessary).

If the job has not been scheduled for a pre-specified printer, printer-state may take any of the printer state values except 'PRINTING'. If the job has not been scheduled, and if no printer was specifically requested in the original job request, the value of this attribute should be NULL.

#### 9.3.10.4 Estimated-completion-time

This attribute indicates the estimated time by which this job will be completed.

estimated-completion-time ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX UTCTime  
SINGLE VALUE  
::= id-att-estimated-completion-time

The time to complete one job is dependent on the time required to complete all other jobs that the scheduling algorithm determines must be printed before this job may be printed. The time to complete each individual job may depend on when on-request resources are available. The time to print each individual job is dependent on the characteristics of the print formats in which each job is encoded and the features of those print formats that are used by the document. It is recommended that document creators provide system hints to the print system (perhaps appropriately encoded in print format headers) that help the print system understand which features of the print formats the document will be exercising. The estimated-completion-time may be easy to calculate for some combinations of print format and scheduling algorithms. It may be impossible to calculate for other combinations. This attribute is highly implementation-dependent.

#### 9.3.10.5 Submission-time

This attribute indicates the time at which this job was submitted and accepted by this server.

submission-time ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX UTCTime  
SINGLE VALUE  
::= id-att-submission-time

#### 9.3.10.6 Modification-time

This attribute indicates the time at which this job was last modified.

modification-time ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX UTCTime  
SINGLE VALUE  
::= id-att-modification-time

If a job has not been modified since its submission, the value of this attribute should be NULL.

#### 9.3.10.7 Started-printing-time

This attribute indicates the time at which this job started printing.

started-printing-time ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX UTCTime  
SINGLE VALUE  
::= id-att-started-printing-time

If a job has not started printing, the value of this attribute should be zeroed.

#### 9.3.10.8 Copies-completed

This attribute indicates the number of complete copies that the printer has printed.

copies-completed ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)  
SINGLE VALUE  
::= id-att-copies-completed

Some printers print multiple copies of each individual page of a document, completing the printing of all copies at the same time, printing as many of these pages as necessary to satisfy the copy count. The interaction between interpretation of this attribute and the following pages-completed and octets-completed attributes is implementation-specific. The value of this attribute is 0 if printing has not started.

#### 9.3.10.9 Pages-completed

This attribute indicates the number of pages that the printer has completed printing.

pages-completed ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)  
SINGLE VALUE  
::= id-att-pages-completed

The accuracy of this value is implementation dependent. It is expected that the value reported is never greater than the actual value.

This attribute may not be supported for all printers and all page description languages. The value of this attribute is 0 if printing has not started.

#### 9.3.10.10 Octets-completed

This attribute indicates the number of octets that the printer has completed printing.

octets-completed ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)  
SINGLE VALUE  
::= id-att-octets-completed

The accuracy of this value is implementation-dependent. It may be approximated by the number of octets conveyed to the printer. This attribute may not be supported for all printers and all page description languages. The value of this attribute is 0 if printing has not started.

#### 9.3.10.11 Intervening-pages

This attribute indicates the number of pages of other jobs to be printed before this job may be scheduled for printing.

intervening-pages ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)

SINGLE VALUE

::= id-att-intervening-pages

#### 9.3.10.12 Intervening-octets

This attribute indicates the number of octets of other jobs to be printed before this job may be scheduled for printing.

intervening-octets ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)

SINGLE VALUE

::= id-att-intervening-octets

#### 9.3.10.13 Intervening-jobs

This attribute indicates the number of other jobs to be printed before this job may be scheduled for printing.

intervening-jobs ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)

SINGLE VALUE

::= id-att-intervening-jobs

#### 9.3.10.14 Print-checkpoint

This attribute indicates the document-number, page-number, copy-number and local context information at which the last checkpoint was taken.

print-checkpoint ATTRIBUTE

WITH ATTRIBUTE-SYNTAX SEQUENCE OF PrintCheckpoint -- see 8.3.3.2 --

SINGLE VALUE

::= id-att-print-checkpoint

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

The ability to generate the previous internal state of the job and the printer is dependent on the page-independence supported by the print format. If a print format is not page-independent it may be possible to emulate the resumption of the job at the checkpoint by processing through the entire document to the checkpoint page without printing any additional pages, then continue printing pages from that point. Some print formats may not support any form of checkpointing. If a Pause operation causes a job to pause in the middle of a document encoded in a print format that does not support any form of checkpointing, the checkpoint should be set to (1, current-copy), to force the system to resume back at the beginning of the current copy. Obviously the ability to checkpoint a job is very implementation-dependent.

#### 9.3.10.15 On-request-resources-required

This attribute identifies the resources (e.g. media and fonts) required to complete the job, but which are not currently available.

on-request-resources-required ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-on-request-resources-required

9.3.10.16 Job-message-from-administrator

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

job-message-from-administrator ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-job-message-from-administrator

9.3.10.17 Last-job-events

This attribute presents a list of the job events that have been reported. The upper limit on how many events will be reported to the user is implementation-dependent.

last-job-events ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX EventType  
MULTI VALUE  
::= id-att-last-job-events

Standard values are defined in 8.5.

This attribute allows systems that do not support asynchronous notification services for event reporting to get the job events synchronously.

9.3.10.18 Id-of-last-accessor

This attribute indicates the name of the last user or program to submit or modify the job.

id-of-last-accessor ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX nameSyntax  
SINGLE VALUE  
::= id-att-id-of-last-accessor

9.3.10.19 Interrupt-at

This attribute indicates the point at which the job was last interrupted.

At ::= ENUMERATED {  
immediately (0),  
end-of-page (1),  
end-of-copy (2),  
end-of-document (3),  
end-of-job (4),  
next-pausable-point (5)}



interrupt-at ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX A1  
SINGLE VALUE  
::= id-att-interrupt-at

9.3.10.20 Pause-at

This attribute indicates the point at which the job was last paused.

pause-at ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX A1  
SINGLE VALUE  
::= id-att-pause-at

9.3.10.21 Error-count

This attribute provides the number of errors encountered while processing the job.

error-count ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER  
SINGLE VALUE  
::= id-att-error-count

9.3.10.22 Warning-count

This attribute provides the number of warnings encountered while processing the job.

warning-count ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER  
SINGLE VALUE  
::= id-att-warning-count

9.3.10.23 Processing-time

This attribute indicates how long an individual job has been processing.

processing-time ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX DeltaTime  
SINGLE VALUE  
::= id-att-processing-time

9.3.11 Job-security-handling

This attribute specifies the security-related printing instructions associated with the job.

JobSecurityHandling ::= SET {  
copy-numbering            {0} OBJECT IDENTIFIER OPTIONAL,  
page-numbering            {1} OBJECT IDENTIFIER OPTIONAL,  
sheet-numbering            {2} OBJECT IDENTIFIER OPTIONAL,  
security-labeling          {3} OBJECT IDENTIFIER OPTIONAL }

job-security-handling ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX JobSecurityHandling  
SINGLE VALUE  
::= id-att-job-security-handling;

Security labelling identifies the security label to applied to the output of the job. The numbering components indicate how security-related numbering of sheets, pages, and copies of the job output is to be done. Assignment and definition of the specific identifiers is expected to be the responsibility of local security authorities.

#### 9.4 Printer Object attributes

These attributes provide information about a logical or physical printer.

##### 9.4.1 Printer-identifier

This attribute uniquely identifies the printer.

```
printer-identifier ATTRIBUTE  
  WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
  SINGLE VALUE  
  ::= id-att-printer-identifier
```

##### 9.4.2 Printer-name

This attribute specifies the name of a printer.

```
printer-name ATTRIBUTE  
  WITH ATTRIBUTE-SYNTAX nameSyntax  
  SINGLE VALUE  
  ::= id-att-printer-name
```

##### 9.4.3 Printer-location

This attribute identifies the location of this printer.

```
printer-location ATTRIBUTE  
  WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
  SINGLE VALUE  
  ::= id-att-printer-location
```

This attribute is not intended to define either distance to a printer or location of the printer in a standardized coordinate system. This attribute is intended to be used as a secondary "location-specific" name, and be interpreted as such. But actual implementation is expected to be vendor-specific.

##### 9.4.4 Printer-model

This attribute identifies the make and model of the printer.

```
printer-model ATTRIBUTE  
  WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
  SINGLE VALUE  
  ::= id-att-printer-model
```

##### 9.4.5 Printer-type

This attribute identifies the "type" of the printer.

```
printer-type ATTRIBUTE  
  WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
  SINGLE VALUE  
  ::= id-att-printer-type
```

Standard values:

Identifier	Descriptor Text
id-val-printer-type-line	Indicates a class of high speed printers. The quality of the output is medium, the orientation is generally landscape on continuous media
id-val-printer-type-letter-quality	Indicates a printer that is able to produce "letter-quality" output, which is a higher resolution than can be achieved with ordinary dot-matrix printers
id-val-printer-type-laser	Indicates a class of printers which produce high quality output
id-val-printer-type-colour-laser	Indicates the laser printer that is capable of rendering colour output
id-val-printer-type-microfiche-imager	Indicates that the output device is capable of presenting information on microfiche
id-val-printer-type-plotter	Indicates a class of output devices which draw images with a set of pens
id-val-printer-type-ink-jet	Indicates a printer which renders output by means of rows of dots. Ink is being ejected on the physical medium
id-val-printer-type-colour-ink-jet	Indicates an inkjet printer also capable of producing a coloured output
id-val-printer-type-dot-matrix	Indicates a printer which renders output by means of rows of dots. The output is visualized via a print ribbon or thermal paper
id-val-printer-type-colour-dot-matrix	Indicates a dot-matrix printer also capable of producing a coloured output
id-val-printer-type-daisy-wheel	Indicates a printer where characters of a daisy wheel are being hit onto the physical medium

#### 9.4.6 Printer-security

This attribute specifies the security classifications supported by the printer. The first element defines the lowest security supported by the printer, the second element defines the highest security supported by the printer.

printer-security ATTRIBUTE

WITH ATTRIBUTE-SYNTAX SecurityClassification

MULTI VALUE

::= id-all-printer-security

SecurityClassification ::= OBJECT IDENTIFIER

Standard values:

id-val-printer-security-unclassified	id-val-printer-security-classified
id-val-printer-security-confidential	id-val-printer-security-secret
id-val-printer-security-top-secret	id-val-printer-security-personal

A job which requires, e.g., 'secret' may be printed only on a printer which offers 'secret' printing. The standard values defined may be used by the local site security organization, or other values may be defined. Semantics of these values are to be defined by the appropriate site management or security organization, and are outside the scope of this standard.

#### 9.4.7 Printer-speed

This attribute specifies the maximum speed of the printer.

```
printer-speed ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)
  SINGLE VALUE
  ::= id-att-printer-speed
```

The values of this attribute are presumed to be in pages per minute (to the nearest integer value).

#### 9.4.8 Printer-resolution

This attribute specifies the maximum resolution of the printer.

```
printer-resolution ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)
  SINGLE VALUE
  ::= id-att-printer-resolution
```

The values of this attribute are presumed to be in dots per inch (dpi, to the nearest integer value). This resolution may be an average of the maximum resolutions if the printer offers different resolutions on the x and y axis.

#### 9.4.9 Printer-state

This attribute identifies the current state of the printer.

```
printer-state ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  SINGLE VALUE
  ::= id-att-printer-state
```

The following standard values are defined:

Identifier	Descriptor Text
id-val-printer-state-unknown	The printer state is not known, or is indeterminate, or is not returned by the operation
id-val-printer-state-idle	The printer is ready to accept jobs, but none have been assigned to it.
id-val-printer-state-printing	The printer is currently printing a job
id-val-printer-state-needs-attention	The printer needs human attention, such as being out of paper, has a jam, needs medium changed, needs cartridge changed, etc.
id-val-printer-state-paused	The operator has (temporarily) paused the printer, by means of the pause operation
id-val-printer-state-shutdown	The printer has been taken out of service, (for a long time), whether for repairs or others reasons. A printer-message can be used to record a reason and estimated time for return to service

**9.4.10 Printer-message**

This attribute indicates the message associated with this object.

printer-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-all-printer-message

The message field may be set by an operator or system administrator to provide useful information to the potential users of this object (e.g. "This printer is out of service and will not be available until Tuesday")

**9.4.11 Attributes-supported-by-printer**

This attribute identifies the attributes recognized and supported by this printer.

attributes-supported-by-printer ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX AttributeType  
MULTI VALUE  
::= id-all-attributes-supported-by-printer

**9.4.12 Object-classes-supported**

This attribute identifies the object classes supported by the printer.

object-classes-supported-by-printer ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-all-object-classes-supported-by-printer

Standard values:

id-oc-job	id-oc-printer
id-oc-server	id-oc-media
id-oc-font	id-oc-named-colour
id-oc-transfer-method	id-oc-delivery-method
id-oc-job-sheet	id-oc-finishing-method
id-oc-output-method	id-oc-imposition-function
id-oc-scheduler	

The object classes are defined in 6.3.

**9.4.13 Printers-supported**

This attribute identifies the logical/physical printers supported by this physical/logical printer.

printers-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-all-printers-supported

**9.4.14 Printers-active**

This attribute identifies the logical/physical printers enabled for this physical/logical printer.

printers-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-printers-active

9.4.15 Fonts-supported

This attribute identifies the font resources supported by this printer.

fonts-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX FontReference  
MULTI VALUE  
::= id-att-fonts-supported

9.4.16 Fonts-active

This attribute identifies the font resources currently active and available on this printer.

fonts-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX FontReference  
MULTI VALUE  
::= id-att-fonts-active

9.4.17 Font-default

This attribute identifies the current default font resource defined for this printer. The default font resource will be used when no specific fonts are specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the font resource to be used.

font-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX FontReference  
SINGLE VALUE  
::= id-att-font-default

9.4.18 Media-supported

This attribute identifies the media supported by this printer.

media-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX MediaIdentifier  
MULTI VALUE  
::= id-att-media-supported

9.4.19 Media-active

This attribute identifies the media currently active and available on this printer.

media-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX MediaIdentifier  
MULTI VALUE  
::= id-att-media-active

**9.4.20 Medium-default**

This attribute identifies the current default medium defined for this printer. The default medium will be used when no specific media are specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the medium to be used.

medium-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX MediaIdentifier  
SINGLE VALUE  
::= id-att-medium-default

**9.4.21 Named-colours-supported**

This attribute identifies named colours supported by this printer.

named-colours-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-named-colours-supported

**9.4.22 Named-colours-active**

This attribute identifies the named-colours currently active and available on this printer.

named-colours-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-named-colours-active

**9.4.23 Named-colour-default**

This attribute identifies the current default named colour defined for this printer. The default named colour will be used when no specific named colour is defined by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the named colour to be used.

named-colour-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-named-colour-default

**9.4.24 Finishing-methods-supported**

This attribute identifies the finishing methods supported by this printer.

finishing-methods-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-finishing-methods-supported

**9.4.25 Finishing-methods-active**

This attribute identifies the finishing methods enabled for this printer.

finishing-methods-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-finishing-methods-active

**9.4.26 Finishing-methods-default**

This attribute identifies the current default finishing methods defined for this printer. The default finishing methods will be used when no specific finishing methods are specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the finishing methods (if any) to be used.

finishing-methods-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-finishing-methods-default

**9.4.27 Output-methods-supported**

This attribute identifies the output methods supported by this printer.

output-methods-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-output-methods-supported

**9.4.28 Output-methods-active**

This attribute identifies the output methods currently active and available on this printer.

output-methods-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-output-methods-active

**9.4.29 Output-methods-default**

This attribute identifies the current default output methods defined for this printer. The default output methods will be used when no specific output methods are specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the output methods to be used.

output-methods-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-output-methods-default

**9.4.30 Output-bins-supported**

This attribute identifies the output bins supported by this printer.



output-bins-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OutputBin  
MULTI VALUE  
::= id-att-output-bins-supported

9.4.31 Output-bins-active

This attribute identifies the output bins currently active and available on this printer.

output-bins-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OutputBin  
MULTI VALUE  
::= id-att-output-bins-active

9.4.32 Output-bins-default

This attribute identifies the current default output bins defined for this printer. The default output bins will be used when no specific output bins are specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the output bins to be used.

output-bins-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OutputBin  
MULTI VALUE  
::= id-att-output-bins-default

9.4.33 Imposition-functions-supported

This attribute identifies the imposition functions supported by this printer.

imposition-functions-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-imposition-functions-supported

Standard values for this attribute are defined in 9.14.1.

9.4.34 Imposition-functions-enabled

This attribute identifies the imposition functions enabled for this printer.

imposition-functions-enabled ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-imposition-functions-enabled

Standard values for this attribute are defined in 9.14.1.

9.4.35 Imposition-functions-default

This attribute identifies the current default imposition functions defined for this printer. The default imposition functions will be used when no specific imposition function is specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the imposition functions (if any) to be used.

imposition-functions-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-imposition-functions-default

Standard values for this attribute are defined in 9.14.1.

9.4.36 Print-qualities-supported

This attribute identifies the print qualities supported by this printer.

print-qualities-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-print-qualities-supported

9.4.37 Print-qualities-active

This attribute identifies the print qualities enabled for this printer.

print-qualities-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-print-qualities-active

9.4.38 Print-quality-default

This attribute identifies the name of the current default print quality defined for this printer. The default print quality will be used when no specific print quality is specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the print quality (if any) to be used.

print-quality-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-print-quality-default

9.4.39 Job-sheets-supported

This attribute identifies the job sheets supported by this printer.

job-sheets-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-job-sheets-supported

9.4.40 Job-sheets-active

This attribute identifies the job sheets enabled for this printer.

job-sheets-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-job-sheets-active

#### 9.4.41 Job-sheet-defaults

These attributes identify, respectively, the current default job start, separator, and end sheet defined for this printer. The default job sheets will be used when no specific job sheets are specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default has been specified, other job and document attributes may be used to determine the job sheet (if any) to be used.

##### 9.4.41.1 Start-sheet-default

This attribute identifies the current default job start sheet, if any, defined for this printer.

start-sheet-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-start-sheet-default

##### 9.4.41.2 Separator-sheet-default

This attribute identifies the current default separator sheet, if any, defined for this printer.

separator-sheet-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-separator-sheet-default

##### 9.4.41.3 End-sheet-default

This attribute identifies the current default job end sheet, if any, defined for this printer.

end-sheet-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-end-sheet-default

#### 9.4.42 Character-sets-supported

This attribute identifies the character sets supported by this printer.

character-sets-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-at-character-sets-supported

#### 9.4.43 Character-sets-active

This attribute identifies the character sets enabled for this printer.

character-sets-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-at-character-sets-active

#### 9.4.44 Character-set-default

This attribute identifies the current default character set defined for this printer. The default character set will be used when no specific character set is specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default

is specified, other job and document attributes may be used to determine the character set (if any) to be used.

character-set-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-character-set-default

9.4.45 Character-mappings-supported

This attribute identifies the character mappings supported by this printer.

character-mappings-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-character-mappings-supported

*Note 24*

*Depending on the work on character mapping and fonts, this attribute may not be needed.*

9.4.46 Character-mappings-active

This attribute identifies the character mappings enabled for this printer.

character-mappings-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-character-mappings-active

9.4.47 Character-mapping-default

This attribute identifies the current default character mapping defined for this printer. The default character mapping will be used when no specific character mapping is specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the character mapping (if any) to be used.

character-mapping-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-character-mapping-default

9.4.48 Character-repertoires-supported

This attribute identifies the character repertoires supported by this printer.

character-repertoires-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-character-repertoires-supported

*Note 25*

*Depending on the work on character mapping and fonts, this attribute may not be needed.*

9.4.49 Character-repertoires-active

This attribute identifies the character repertoires enabled for this printer.

character-repertoires-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-character-repertoires-active

9.4.50 Character-repertoire-default

This attribute identifies the current default character repertoire defined for this printer. The default character repertoire will be used when no character repertoire is specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the character repertoire (if any) to be used.

character-repertoire-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-character-repertoire-default

9.4.51 Start-message-supported

This attribute indicates whether an operator start message is supported by this printer.

start-message-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX BOOLEAN  
SINGLE VALUE  
::= id-att-start-message-supported

9.4.52 End-message-supported

This attribute indicates whether an operator end message is supported by this printer.

end-message-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX BOOLEAN  
SINGLE VALUE  
::= id-att-end-message-supported

9.4.53 Sides-supported

This attribute indicates the number of surfaces of a medium that can be imaged by this printer.

SidesSupported ::= ENUMERATED {  
both (0), -- Both 1-sided and 2-sided printing is supported --  
one (1), -- Only 1-sided printing is supported --  
two (2)} -- Only 2-sided printing is supported --

sides-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX SidesSupported  
SINGLE VALUE  
::= id-att-sides-supported

9.4.54 Sides-default

This attribute indicates the current default number of medium surfaces imaged by this printer. The default number of sides will be used when no specific value for sides is specified by the job attributes or the document contents. If no value is defined by the job or document

attributes and no default is specified, other job and document attributes may be used to determine the number of sides to be imaged.

sides-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX SidesSupported  
SINGLE VALUE  
 ::= id-att-sides-default

9.4.55 Page-select-supported

This attribute indicates the types of page identifier supported by this printer, either nominal numeric or alpha-numeric string, or both. The types of page identifiers supported are defined in 9.3.6.13.

page-select-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX PageIdType  
MULTI VALUE  
 ::= id-att-page-select-supported

PageIdType ::= ENUMERATED {  
 numeric (1),  
 alphanumeric (2) }

9.4.56 Maximum-copies-supported

This attribute indicates the maximum number of copies of a document that can be imaged by this printer.

maximum-copies-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
 ::= id-att-maximum-copies-supported

9.4.57 Content-formats-supported

This attribute identifies the content formats currently supported by this printer.

content-formats-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
 ::= id-att-content-formats-supported

9.4.58 Content-format-default

This attribute identifies the current default content format defined for this printer. The default will be used when no content format is specified by the job attributes or the document content. If no value is defined by the job or document attributes and no default has been specified, other job and document attributes may be used to determine the content format to be used.

content-format-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
 ::= id-att-content-format-default

**9.4.59 Document-formats-supported**

This attribute identifies the document formats currently supported by this printer.

document-formats-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-at-document-formats-supported

**9.4.60 Document-format-default**

This attribute identifies the current default document format defined for this printer. The default document format will be used when no specific value for document format is specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the document format to be used.

document-format-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-document-format-default

**9.4.61 Area-supported**

This attribute identifies the assured reproduction areas currently supported by this printer.

area-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX AssuredReproductionArea  
MULTI VALUE  
::= id-at-area-supported

**9.4.62 Area-default**

This attribute identifies the current default assured reproduction area defined for this printer. The default will be used when no ara is specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the ara to be used.

area-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX AssuredReproductionArea  
SINGLE VALUE  
::= id-at-area-default

**9.4.63 Page-sizes-supported**

This attribute specifies the page sizes supported by the imposition processor or simple text formatter of the printing system. The default for a print system is taken from the value given for medium-default. Possible values for this attribute are the same as those possible for medium-size.

page-sizes-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-at-page-sizes-supported

**9.4.64 Orientations-supported**

This attribute identifies the content orientations currently supported by this printer.

orientations-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-orientations-supported

**9.4.65 Orientation-default**

This attribute identifies the current default content orientation defined for this printer. The default orientation will be used when no specific value for orientation is specified by the job attributes or the document contents. If no value is defined by the job or document attributes and no default is specified, other job and document attributes may be used to determine the orientation to be used.

orientation-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-orientation-default

**9.4.66 Page-order-assumed**

This attribute indicates whether the printer assumes that the document to be imaged is in 1-n order. The value TRUE is defined to mean that the pages are in 1-n order.

page-order-assumed ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX BOOLEAN  
SINGLE VALUE  
::= id-att-page-order-assumed

**9.4.67 Page-independence-assumed**

This attribute indicates whether the printer assumes that the document may be interpreted in a page-independent manner. The value TRUE is defined to mean that the document is page-independent.

page-independence-assumed ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX BOOLEAN  
SINGLE VALUE  
::= id-att-page-independence-assumed

**9.4.68 Associated-server**

This attribute indicates the name of the server with which this printer is currently associated.

*Note 26*

*In some print systems, the server and printer are one and the same. For these systems, the printer identifier and the associated-server identifier are the same. In other systems, many individual printers may be accessed through a single server. In these systems the printer identifiers might not be the same as the associated-server identifier. The exact configuration is implementation-dependent.*



associated-server ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-associated-server

9.4.69 Sheet-count

This attribute provides the number of sheets consumed during the operation of this printer.

sheet-count ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER  
SINGLE VALUE  
::= id-att-sheet-count

9.4.70 Printer-needs-attention-time

This attribute specifies how long the printer has been waiting for attention (e.g., waiting for additional media).

printer-needs-attention-time ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX DeltaTime  
SINGLE VALUE  
::= id-att-printer-needs-attention-time

9.4.71 Printer-needs-key-operator-attention-time

This attribute specifies how long a printer has been waiting for attention by a key operator (e.g., jammed). The attention by key operator may be independent of "normal" attention required (e.g., the job could be continued with lower quality, if the printer requires the key operator to perform preventative maintenance).

printer-needs-key-operator-attention-time ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX DeltaTime  
SINGLE VALUE  
::= id-att-printer-needs-key-operator-attention-time

9.5 Server Object attributes

These attributes provide information about a particular server.

9.5.1 Server-identifier

This attribute uniquely identifies the server.

server-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-server-identifier

9.5.2 Server-name

This attribute indicates the name of the server.

server-name ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX nameSyntax  
SINGLE VALUE  
::= id-att-server-name

### 9.5.3 Server-address

This attribute indicates the address of this server.

server-address ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OCTET STRING  
SINGLE VALUE  
::= id-att-server-address

### 9.5.4 Server-state

This attribute indicates the state of this server.

server-state ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX stateSyntax  
SINGLE VALUE  
::= id-att-server-state

The standard values are defined in 9.2.5

### 9.5.5 Server-message

This attribute indicates the message associated with this server.

server-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-server-message

### 9.5.6 Object-classes-supported

This attribute identifies the object-classes (e.g. printer, media, form, etc.) supported by this server.

object-classes-supported-by-server ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-object-classes-supported-by-server

Standard values:

id-cc-job	id-cc-printer
id-cc-server	id-cc-media
id-cc-form	id-cc-named-colour
id-cc-transfer-method	id-cc-delivery-method
id-cc-job-sheet	id-cc-finishing-method
id-cc-output-method	id-cc-imposition-function
id-cc-scheduler	

### 9.5.7 Attributes-supported-by-server

This attribute identifies the attributes supported by this server.

attributes-supported-by-server ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX AttributeType  
MULTI VALUE  
::= id-att-attributes-supported-by-server

9.5.8 Physical-printers-supported

This attribute identifies the physical printers supported by this server.

physical-printers-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-physical-printers-supported

9.5.9 Physical-printers-active

This attribute identifies the physical printers active and available through this server.

physical-printers-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-physical-printers-active

9.5.10 Logical-printers-supported

This attribute identifies the logical printers supported by this server.

logical-printers-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-logical-printers-supported

9.5.11 Logical-printers-active

This attribute identifies the logical printers active and available through this server.

logical-printers-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-logical-printers-active

9.5.12 Translators-supported

This attribute identifies the translators (or imposition functions) supported by this server.

translators-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-translators-supported

9.5.13 Translators-active

This attribute identifies the translators (or imposition functions) active and available through this server.

translators-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-at-translators-active

9.5.14 Delivery-methods-supported

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

delivery-methods-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-at-delivery-methods-supported

9.5.15 Delivery-methods-active

This attribute identifies the delivery-methods active and available through this server.

delivery-methods-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-at-delivery-methods-active

9.5.16 Scheduling-algorithms-supported

This attribute identifies the scheduling-algorithms supported by this server.

scheduling-algorithms-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-at-scheduling-algorithms-supported

9.5.17 Scheduling-algorithm-active

This attribute identifies the scheduling-algorithm active and available through this server.

scheduling-algorithm-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-scheduling-algorithm-active

9.5.18 System-hints-supported

This attribute identifies the system hints supported by this printer.

system-hints-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-at-system-hints-supported

Standard values:

id-att-octet-count	id-att-page-count
id-att-glyph-count	id-att-font-count
id-att-font-change-count	id-att-maximum-fonts-per-page
id-att-percent-graphics	id-att-percent-images
id-att-percent-adjustment	

Standard values for this attribute are defined in 9.14.1.

#### 9.4 Media Object attributes

This object class defines media for printing. There are two kinds of media objects, logical and absolute media.

With a logical medium, the desired approximate effect is indicated (eg A4-white or Our-Company-Letterhead). A logical medium has a generic set of characteristics. A logical medium will be realized by the server by using one of the absolute media that is associated with this logical medium.

An absolute medium is a unique realisation of a specified desired effect. It prescribes the physical medium that must be used, and in addition, it may identify an image that may be overlaid on this physical medium to realize the absolute medium.

Examples of logical and associated absolute media are:

ISO-A4-white	A4-white 90 grams normal grain A4-creme 90 grams A4-white 120 grams
Our-Company-Letterhead	NA-letter-white; form 120 (logo) Company-Stationery 90 grams
ISO-B5-coloured	ISO-B5-yellow ISO-B5-green ISO-B5-red 120 grams

The mapping of logical media on absolute media is the responsibility of the server and is outside the scope of this standard.

In the normal day-to-day situation a user would choose from the logical media that are offered at his installation. If the user wants absolute control over the result he can inquire about the available media via the ListObjectAttributes operation, and select the exact absolute medium of his choice.

The following standard values for logical media are defined:

Identifier	Descriptor Text
ISO A4 white	Specifies the ISO A4 white medium with size: 210 mm by 297 mm as defined in ISO 216
ISO A4 coloured	Specifies the ISO A4 coloured medium with size: 210 mm by 297 mm as defined in ISO 216
ISO A4 transparent	Specifies the ISO A4 transparent medium with size: 210 mm by 297 mm as defined in ISO 216
ISO A3 white	Specifies the ISO A3 white medium with size: 297 mm by 420 mm as defined in ISO 216
ISO A3 coloured	Specifies the ISO A3 coloured medium with size: 297 mm by 420 mm as defined in ISO 216
ISO A5 white	Specifies the ISO A5 white medium with size: 148 mm by 210 mm as defined in ISO 216
ISO A5 coloured	Specifies the ISO A5 coloured medium with size: 148 mm by 210 mm as defined in ISO 216
ISO B4 white	Specifies the ISO B4 white medium with size: 176 mm by 250 mm as defined in ISO 216
ISO B4 coloured	Specifies the ISO B4 coloured medium with size: 176 mm by 250 mm as defined in ISO 216
ISO B5 white	Specifies the ISO B5 white medium with size: 176 mm by 250 mm as defined in ISO 216
ISO B5 coloured	Specifies the ISO B5 coloured medium with size: 176 mm by 250 mm as defined in ISO 216
North American letter white	Specifies the North American letter white medium with size: 8.5 inches by 11 inches
North American letter coloured	Specifies the North American letter coloured medium with size: 8.5 inches by 11 inches
North American letter transparent	Specifies the North American letter transparent medium with size: 8.5 inches by 11 inches
North American legal white	Specifies the North American legal white medium with size: 8.5 inches by 14 inches
North American legal coloured	Specifies the North American legal coloured medium with size: 8.5 inches by 14 inches
ISO B5 envelope	Specifies the ISO B5 envelope medium with size: 176 mm by 250 mm as defined in ISO 216 and ISO 269
ISO B4 envelope	Specifies the ISO B4 envelope medium with size: 250 mm by 353 mm as defined in ISO 216
ISO C4 envelope	Specifies the ISO C4 envelope medium with size: 229 mm by 324 mm as defined in ISO 216 and ISO 269
ISO C5 envelope	Specifies the ISO C5 envelope medium with size: 162 mm by 229 mm as defined in ISO 269
ISO Designated Long envelope	Specifies the ISO Designated Long envelope medium with size: 110 mm by 220 mm as defined in ISO 269
North American 10x13 envelope	Specifies the North American 10x13 envelope medium with size: 10 inches by 13 inches
North American 9x12 envelope	Specifies the North American 9x12 envelope medium with size: 9 inches by 12 inches
North American business envelope	Specifies the North American business envelope medium with size: 4.125 inches by 9.5 inches

These attributes provide information about a particular medium.

All of the following attributes are applicable for both logical media and absolute media unless explicitly stated otherwise. However, logical media will generally have fewer attributes defined than absolute media.

#### 9.6.1 Medium-identifier

This attribute identifies either a logical or an absolute medium.

**AbsoluteMediumIdentifier** ::= SEQUENCE {  
     physical-medium-identifier OBJECT IDENTIFIER,  
     overlay-image-identifier OBJECT IDENTIFIER OPTIONAL }

-- an overlay image is a means that can be employed to render a specific desired effect in --  
 -- the printed output, eg an electronic or other overlay form --

**MediumIdentifier** ::= CHOICE {  
     logical-medium-identifier OBJECT IDENTIFIER,  
     absolute-medium-identifier AbsoluteMediumIdentifier }

medium-identifier ATTRIBUTE

WITH ATTRIBUTE-SYNTAX MediumIdentifier

SINGLE VALUE

::= id-all-medium-identifier

Standard values:

id-val-medium-iso-a4-white	id-val-medium-iso-a4-coloured
id-val-medium-iso-a4-transparent	id-val-medium-iso-a3-white
id-val-medium-iso-a3-coloured	id-val-medium-iso-a5-white
id-val-medium-iso-a5-coloured	id-val-medium-iso-b4-white
id-val-medium-iso-b4-coloured	id-val-medium-iso-b5-white
id-val-medium-iso-b5-coloured	id-val-medium-north-american-letter-white
id-val-medium-north-american-letter-coloured	
id-val-medium-north-american-letter-transparent	
id-val-medium-north-american-legal-white	
id-val-medium-north-american-legal-coloured	
id-val-medium-iso-b5-envelope	id-val-medium-iso-b4-envelope
id-val-medium-c4-envelope	id-val-medium-c5-envelope
id-val-medium-iso-designated-long-envelope	
id-val-medium-north-american-10x13-envelope	
id-val-medium-north-american-3x12-envelope	
id-val-medium-north-american-business-envelope	

#### 9.6.2 Medium-state

This attribute identifies the state of this medium (e.g. 'AVAILABLE', 'ON-REQUEST', 'UNAVAILABLE')

medium-state ATTRIBUTE

WITH ATTRIBUTE-SYNTAX stateSyntax

SINGLE VALUE

::= id-all-medium-state

The standard values are defined in 9.2.5

**9.6.3 Medium-message**

This attribute indicates the message associated with this medium.

medium-message ATTRIBUTE  
 WITH ATTRIBUTE-SYNTAX messageSyntax  
 SINGLE VALUE  
 ::= id-at-medium-message

**9.6.4 Medium-type**

This attribute identifies the type of this medium. (e.g. stationery, envelope, transparency, etc.)

medium-type ATTRIBUTE  
 WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
 SINGLE VALUE  
 ::= id-at-medium-type

Standard values:

id-val-medium-type-stationery	id-val-medium-type-transparency
id-val-medium-type-envelope	id-val-medium-type-form
id-val-medium-type-continuous	

**9.6.5 Medium-size**

This attribute identifies the size of this medium by specifying a standardized value.

medium-size ATTRIBUTE  
 WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
 SINGLE VALUE  
 ::= id-at-medium-size

**9.6.6 Medium-Dimensions**

This attribute indicates the size of this medium in millimetres.

medium-dimensions ATTRIBUTE  
 WITH ATTRIBUTE-SYNTAX SEQUENCE  
 {INTEGER(1..ub-integer), INTEGER(1..ub-integer)}  
 SINGLE VALUE  
 ::= id-at-medium-dimensions

Note 27

*This attribute may not be applicable to all logical media.*

**9.6.7 Medium-colour**

This attribute indicates the colour of this medium.

medium-colour ATTRIBUTE  
 WITH ATTRIBUTE-SYNTAX nameSyntax  
 SINGLE VALUE  
 ::= id-at-medium-colour



Note 28

For logical media, this attribute could, for example, have the value "coloured", while for associated absolute media this could have the value "blue", "red", etc.

**9.6.8 Medium-tooth**

This attribute indicates the tooth (or roughness) of this medium. 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 must match the tooth of the medium for best resolution.

medium-tooth ATTRIBUTE

WITH ATTRIBUTE-SYNTAX nameSyntax

SINGLE VALUE

::= id-att-medium-tooth

**9.6.9 Medium-grain**

This attribute indicates the grain of this medium. Grain affects the curl of medium. Some marking engines are sensitive to the resulting curl.

medium-grain ATTRIBUTE

WITH ATTRIBUTE-SYNTAX nameSyntax

SINGLE VALUE

::= id-att-medium-grain

**9.6.10 Medium-sides**

This attribute indicates the number of sides (or imageable surfaces) of this medium. (e.g. 1 sided media such as transparency mediums, or 2 sided medium such as stationery)

medium-sides ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (1..2)

SINGLE VALUE

::= id-att-medium-sides

**9.6.11 Medium-weight**

This attribute indicates the weight of this medium in grams per m<sup>2</sup>.

medium-weight ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)

SINGLE VALUE

::= id-att-medium-weight

**9.6.12 Medium-assured-reproduction-area**

This attribute indicates the assured reproduction area of this medium.

medium-assured-reproduction-area ATTRIBUTE

WITH ATTRIBUTE-SYNTAX AssuredReproductionArea

SINGLE VALUE

::= id-att-medium-assured-reproduction-area

**9.6.13 Medium-associated-media**

This attribute identifies which absolute mediums, or set of absolute media is associated with this logical medium. This attribute does not exist for absolute media.

MediaList ::= SEQUENCE OF SEQUENCE OF AbsoluteMediumIdentifier

medium-associated-media ATTRIBUTE

WITH ATTRIBUTE-SYNTAX MediaList

SINGLE VALUE

::= id-at-medium-associated-media

The first SEQUENCE OF lists the different realisations of the logical medium. The order in which the absolute media (or sequence of absolute media) are given in this first SEQUENCE OF indicates the order in which these absolute media are likely to be used to realize the logical medium. The first item of the sequence is the realization currently used.

The second SEQUENCE OF has normally only one element. The only case when several elements occur is when the logical medium represents a multi-part form.

#### 9.6.14 Medium-form-parts

This attribute indicates the number of parts supported by the form associated with this medium.

medium-form-parts ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)

SINGLE VALUE

::= id-at-medium-form-parts

*New 29*

*This attribute is only valid for logical media.*

### 9.7 Font Resource Object attributes

These attributes provide information about a particular font resource.

#### 9.7.1 Font-identifier

This attribute uniquely identifies the font.

font-identifier ATTRIBUTE

WITH ATTRIBUTE-SYNTAX FontReference

SINGLE VALUE

::= id-at-font-identifier

#### 9.7.2 Font-type

This attribute identifies the type of this font resource.

font-type ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-at-font-type

#### 9.7.3 Font-state

This attribute identifies the state of the font resource. (e.g. 'AVAILABLE', 'ON-REQUEST', 'UNAVAILABLE').

font-state ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX stateSyntax  
SINGLE VALUE  
::= id-att-font-state

The standard values are defined in 9.2.5

#### 9.7.4 Font-message

This attribute indicates the message associated with this object. The message field may be set by an operator or system administrator to provide useful information to the potential users of this object (e.g. "This font is on order and will not be available until June 30").

font-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-font-message

### 9.8 Named-colour Object attributes

These attributes provide information about a particular named-colour.

#### 9.8.1 Named-colour-identifier

This attribute uniquely identifies the named-colour.

named-colour-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-named-colour-identifier

#### 9.8.2 Named-colour-state

This attribute indicates the state of the named colour. (e.g. 'AVAILABLE', 'BUSY', 'ON-REQUEST', 'UNAVAILABLE').

named-colour-state ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX stateSyntax  
SINGLE VALUE  
::= id-att-named-colour-state

The standard values are defined in 9.2.5.

#### 9.8.3 Named-colour-message

This attribute provides an informative message associated with this named-colour. The message field may be set by an operator or system administrator to provide useful information to the potential users of this object (e.g. "This resource is out of stock and will not be available until June 31").

colour-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-colour-message

### 9.9 Transfer-method Object attributes

This object class defines standard transfer methods.

Identifier	Descriptor Text
Transfer document with job request	<p>This value identifies the transfer of the document by value in the attribute "document-content"</p> <p>This value identifies the referenced data transfer as defined in ECMA-131. The reference is provided by the value of the attribute "document-reference"</p> <p>This value specifies the use of FTAM by the client to transfer the contents of the document or file to be printed to the print server. The client requests a reference to an FTAM filestore to which the document(s) is to be transferred</p> <p>This value specifies the use of FTAM by the server to transfer the contents of the document or file to be printed to the print server. The client provides a reference to an FTAM filestore with the PrintArgument</p>
Referenced data transfer	
FTAM initiated by DPA Client	
FTAM initiated by DPA Server	

These attributes provide information about a particular transfer-method.

#### 9.9.1 Transfer-method-identifier

This attribute uniquely identifies the transfer-method.

transfer-method-identifier ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

= id-at-transfer-method-identifier

Standard values:

id-val-transfer-method-transfer-document-with-job-request id-val-transfer-method-referenced-data-transfer id-val-transfer-method-ftam-by-client id-val-transfer-method-ftam-by-server
--

#### 9.9.2 Document-reference-type

This attribute identifies the type of the document reference required by this transfer-method. (e.g. RDT-reference).

document-reference-type ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

= id-at-document-reference-type

Standard values: id-val-document-ref-type-immediate, id-val-document-ref-type-rdt.

#### 9.9.3 Transfer-method-message

This attribute indicates the message associated with this object.

transfer-method-message ATTRIBUTE  
 WITH ATTRIBUTE-SYNTAX messageSyntax  
 SINGLE VALUE  
 ::= id-at-transfer-method-message

The message field may be set by an operator or system administrator to provide useful information to the potential users of this object.

### 9.10 Delivery Method Object attributes

Delivery method objects define the existence and characteristics of methods for delivering job results, logs and notifications. A few standard instances of delivery methods are defined in this standard to ensure interoperability.

Identifier	Descriptor Text
Pick-up	This value specifies that a hardcopy output of a print job is to be picked up at a place defined by the system administration. This value may be used for the delivery of hardcopy results and printed logs. It shall not be used for the delivery of event notifications.
Secure pick-up	This value specifies that a hardcopy output of a print job is to be picked up at a place defined by the system administration. This place must be protected against unauthorized access. This value may be used for the delivery of hardcopy results and printed logs. It shall not be used for the delivery of event notifications.
Inter office mail	This value specifies that a hardcopy output of a print job is to be mailed to the recipient as specified in delivery-identifier and delivery-address. For this purpose the inter office mail service should be used. This value may be used for the delivery of hardcopy results and printed logs. It shall not be used for the delivery of event notifications. The use of this value requires the presence of values for the attributes "...-delivery-identifier" and "...-delivery-address". The format of the recipient's address should follow the local administrative convention for inter office mail.
Postal mail	This value may be used for the delivery of hardcopy results and printed logs. It shall not be used for the delivery of event notifications. This value specifies that a hardcopy output of a print job is to be mailed to the recipient as specified in delivery-identifier and delivery-address. For this purpose the postal mail service should be used. The use of this value requires the presence of values for the attributes "...-delivery-identifier" and "...-delivery-address". The format of the recipient's address should follow the international conventions for postal mail.

Identifier	Descriptor Text
Notification service	This value specifies that an electronic output of a print job is to be forwarded to the recipient in an implementation dependent manner. This value may be used for the delivery of event notifications. It shall not be used for the delivery of hardcopy results, printed logs or electronic images of logs. The use of this value may require the presence of values for the attributes "...-delivery-identifier" and "...-delivery-address"
Electronic mail	This value specifies that an electronic output of a print job is to be forwarded to the recipient using electronic mail. This value may be used for the delivery of event notifications and electronic images of logs. It shall not be used for the delivery of hardcopy results and printed logs. The use of this value may require the presence of values for the attributes "...-delivery-identifier" and "...-delivery-address"
File	This value specifies that an electronic output of a print job is to be copied to the file specified by the "method-information" attribute in an implementation dependent manner. This value may be used for the delivery of event notifications, electronic images of results and electronic images of logs. It shall not be used for the delivery of hardcopy results and printed logs. The use of this value may require the presence of values for the attributes "...-delivery-identifier" and "...-delivery-address"
None	This value specifies that no delivery method is requested. This method would normally be used in an argument to the ModifyJob operation to cancel a previously requested delivery method. In this case, the delivery method to be used will revert to the system default

These attributes provide information about a particular delivery-method.

#### 9.10.1 Delivery-method-identifier

This attribute uniquely identifies the delivery-method.

delivery-method-identifier ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-att-delivery-method-identifier

Standard values:

id-val-delivery-method-pick-up	id-val-delivery-method-secure-pickup
id-val-delivery-method-interoffice-mail	id-val-delivery-method-postal-mail
id-val-delivery-method-notification-service	
id-val-delivery-method-electronic-mail	id-val-delivery-method-file
id-val-delivery-method-none	

### 9.10.2 Delivery-method-message

This attribute indicates the message associated with this object.

delivery-method-message ATTRIBUTE

WITH ATTRIBUTE-SYNTAX messageSyntax

SINGLE VALUE

::= id-att-delivery-method-message

The message field may be set by an operator or system administrator to provide useful information to the potential users of this object.

### 9.11 Job Sheet Object attributes

Job Sheet objects define the information to be printed on a start sheet, separator sheet or end sheet. It is expected that many implementation and installation specific job sheets may be defined. To ensure basic interoperability of print systems, this standard specifies the existence of a few standard instances of job sheets.

Identifier	Descriptor Text
None	This value indicates that no start, separator or end sheet is to be provided, respectively
Brief	This value references a short form of a start separator or end sheet. This sheet is expected to contain information sufficient to identify the print job (e.g., job-name, job-originator, etc.
Full	This value references a long form of a start, separator or end sheet. This sheet is expected to contain more extensive information identifying the print job and the user that submitted the job

This standard does not define standard values for the attributes of these standard instances (other than job sheet identifier). Attributes of standard and non-standard job sheets are expected to be implementation and site-specific.

These attributes provide information about a particular job sheet.

#### 9.11.1 Job-sheet-identifier

This attribute uniquely identifies the job sheet.

job-sheet-identifier ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-att-job-sheet-identifier

## Standard Values

id-val-job-sheet-none	id-val-job-sheet-brief
id-val-job-sheet-full	

## 9.11.2 Job-sheet-message

This attribute indicates the message associated with this object.

job-sheet-message ATTRIBUTE

WITH ATTRIBUTE-SYNTAX messageSyntax

SINGLE VALUE

::= id-att-job-sheet-message

The message field may be set by an operator or system administrator to provide useful information to the potential users of this object.

## 9.12 Finishing Method Object attributes

This object class is to be used to define standard finishing methods.

Identifier	Descriptor Text
Staples	This indicates that staples are to be used to bind the document. The exact placement of the staples is site-defined
Staples Top	This indicates that a site-defined number of staples should be placed at the top of the document
Staples Bottom	This indicates that a site-defined number of staples should be placed at the bottom of the document
Staples Left	This indicates that a site-defined number of staples should be placed on the left edge of the document
Staples Right	This indicates that a site-defined number of staples should be placed on the right edge of the document
Staples Top-Left	This indicates that one or more staples should be placed on the top left corner of the document
Staples Bottom-Left	This indicates that one or more staples should be placed on the bottom left corner of the document
Staples Top-Right	This indicates that one or more staples should be placed on the top right corner of the document
Staples Bottom-Right	This indicates that one or more staples should be placed on the bottom right corner of the document
Staples None	This indicates that no additional staples are to be added to the document
Holes	This indicates that holes are to be drilled or punched in the document. The exact placement of the holes is site-defined
Holes Top	This indicates that a site-defined number of holes should be placed at the top of the document
Holes 2-Top	This indicates that two holes should be placed at the top of the document
Holes 3-Top	This indicates that three holes should be placed at the top of the document
Holes 4-Top	This indicates that four holes should be placed at the top of the document
Holes Left	This indicates that a site-defined number of holes



Identifier	Descriptor Text
Holes 2-Left	should be placed on the left edge of the document This indicates that two holes should be placed on the left edge of the document
Holes 3-Left	This indicates that three holes should be placed on the left edge of the document
Holes 4-Left	This indicates that four holes should be placed on the left edge of the document
Holes Right	This indicates that a site-defined number of holes should be placed on the right edge of the document
Holes 2-Right	This indicates that two holes should be placed on the right edge of the document
Holes 3-Right	This indicates that three holes should be placed on the right edge of the document
Holes 4-Right	This indicates that four holes should be placed on the right edge of the document
Holes None	This indicates that no additional holes are to be added to the document
Cover	This object is specified when it is desired to select non-printed (or pre-printed) cover for the document. This does not supplant the specification of a print cover (on cover stock medium) by the document print format
Cover None	This indicates that no additional cover is to be applied to the document
Bind Top	This indicates that a binding is to be placed on the top edge of the document
Bind Left	This indicates that a binding is to be placed on the left edge of the document
Bind Right	This indicates that a binding is to be placed on the right edge of the document
Bind Bottom	This indicates that a binding is to be placed on the bottom edge of the document
Bind None	This indicates that no binding is to be done

These attributes provide information about a particular finishing method.

#### 9.12.1 Finishing-method-identifier

This attribute uniquely identifies the finishing-method.

finishing-method-identifier ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-att-finishing-method-identifier

## Standard values:

id-val-finishing-staples	id-val-finishing-staples-top
id-val-finishing-staples-bottom	id-val-finishing-staples-left
id-val-finishing-staples-right	id-val-finishing-staples-top-left
id-val-finishing-staples-bottom-left	id-val-finishing-staples-top-right
id-val-finishing-staples-bottom-right	id-val-finishing-staples-none
id-val-finishing-holes	id-val-finishing-holes-top
id-val-finishing-holes-2-top	id-val-finishing-holes-3-top
id-val-finishing-holes-4-top	id-val-finishing-holes-left
id-val-finishing-holes-2-left	id-val-finishing-holes-3-left
id-val-finishing-holes-4-left	id-val-finishing-holes-right
id-val-finishing-holes-2-right	id-val-finishing-holes-3-right
id-val-finishing-holes-4-right	id-val-finishing-holes-none
id-val-finishing-cover	id-val-finishing-nocover
id-val-finishing-bind	id-val-finishing-bind-top
id-val-finishing-bind-left	id-val-finishing-bind-right
id-val-finishing-bind-bottom	id-val-finishing-bind-none

## 9.12.2 Finishing-method-message

This attribute indicates the message associated with this object. The message field may be set by an operator or system administrator to provide useful information to the potential users of this object (e.g. "This facility will not be available until Feb 29").

finishing-method-message ATTRIBUTE

WITH ATTRIBUTE-SYNTAX messageSyntax

SINGLE VALUE

:= id-att-finishing-method-message

## 9.13 Output Method Object attributes

This object class is to be used to define standard output specification methods.

Identifier	Descriptor Text
Page-Collate	This value specifies that the pages of a job are to be in sequence, when multiple copies of the job are specified by the "copy count" attribute (possibly up to some implementation-defined maximum number of copies). Whether this effect is achieved by placing copies of the job in multiple output bins or in the same output bin with implementation-defined job separation is implementation-dependent. Also whether it is achieved by making multiple passes over the job or by using an output sorter is implementation-dependent.

Identifier	Descriptor Text
No Page-Collate	This value specifies that the copies of the pages of a job are to follow one another when multiple copies are specified by the "copy-count" attribute. That is, if 3 copies are specified, three copies of page 1 are printed then three copies of page 2, etc., and no collating of these pages is desired. This may be useful in some implementations where multiple copies requires re-interpretation of the print format of the job and the job contains some compute intensive pages (such as images)
Document-Collate	This value specifies that the documents of a job are to be printed in sequence within the job, when multiple copies of the documents are specified by the "copy-count" attributes of the individual jobs. Whether this effect is achieved by placing copies of the documents in multiple output bins or in the same output bin with implementation-defined document separation is implementation-dependent. Also whether it is achieved by making multiple passes over the documents in the job or by using an output sorter is implementation-dependent
No Document-Collate	This value specifies that copies of a document are to follow one another when multiple copies are specified by the "copy-count" attribute. That is, if 3 copies are specified and there is more than one document in the job, then the output stack is expected to have three copies of document 1, three copies of document 2, etc. (no collating of these documents is required)
Decollate	The parts of a multi-part form are to be separated and sorted into stacks for each part
No Decollate	The parts of a multi-part form are to remain intact
Bursting	Continuous media is to be separated into individual sheets, generally by bursting along perforations
No Bursting	Continuous media is to remain continuous, no bursting is desired

These attributes provide information about a particular output-method.

#### 9.13.1 Output-method-identifier

This attribute uniquely identifies the output-method.

output-method-identifier ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= .i.id-all-output-method-identifier

## Standard Values:

id-val-output-page-collate	id-val-output-nopage-collate
id-val-output-document-collate	id-val-output-nodocument-collate
id-val-output-decollate	id-val-output-nodecollate
id-val-output-bursting	id-val-output-nobursting

## 9.13.2 Output-method-message

This attribute indicates the message associated with this object.

output-method-message ATTRIBUTE

WITH ATTRIBUTE-SYNTAX messageSyntax

SINGLE VALUE

= id-att-output-method-message

The message field may be set by an operator or system administrator to provide useful information to the potential users of this output method.

## 9.14 Imposition-Function Object attributes

This object class is used to define standard convenience imposition functions. These imposition functions may be requested using the imposition-specification attribute.

Identifier	Descriptor Text
Simplex	This value indicates that the document has been formatted for one-sided printing, independent of the value of the "sides" attribute
Duplex	This value indicates that the document has been formatted for two-sided, non-tumbled printing, independent of the value of the "sides" attribute. No rotation of the content pages is applied by this convenience imposition function (although such rotation may be provided by the document print format). A value of 'one' for the "sides" attribute in combination with a value of 'duplex' for the "imposition-specification" attribute could be used to produce a one-sided master that would be used to produce two-sided copies using some other form of printing, such as offset printing
Tumble	This value indicates that the document has been formatted for two-sided printing in which alternate pages in the principal orientation are rotated 180 degrees, independent of the value of the "sides" attribute. Pages in the subordinate orientation are not rotated. If the content orientation is specified as 'mixed', this convenience imposition function assumes that the predominant content orientation is 'portrait'. A value of 'one' for the "sides" attribute in combination with a value of 'tumble' for the "imposition-specification" attribute could be used to produce a one-sided master that would be used to produce two-sided copies using some other form of printing, such as offset printing

Identifier	Descriptor Text
2-up	Reduces the sizes of two adjacent pages by approximately the quotient of the length of the short edge divided by the length of the long edge and places them on a single side of the medium, in order to simplify binding: - When the value of "content-orientation" is "portrait", each page is rotated +90 degrees (i.e., counter-clockwise) - When the value of "content-orientation" is "landscape", each page is rotated -90 degrees (i.e., clockwise)
4-up	Reduces the sizes of four adjacent pages by approximately half and places them on a single side of the medium. The pages are not rotated. The order of the four pages is left to right, then top to bottom for both portrait and landscape pages
Left bind shift	This value specifies that a shift of the page content is to be made in order to accommodate binding on the "left" of the physical medium independent of the actual "content-orientation" of the page, i.e., on the long edge of the medium corresponding to the left edge if the page were to be a portrait page
Right bind shift	This value specifies that a shift of the page content is to be made in order to accommodate binding on the "right" of the physical medium independent of the actual "content-orientation" of the page, i.e., on the long edge of the medium corresponding to the right edge if the page were to be a portrait page
Top bind shift	This value specifies that a shift of the page content is to be made in order to accommodate binding on the "top" of the physical medium independent of the actual "content-orientation" of the page, i.e., on the short edge of the medium corresponding to the top edge if the page were to be a portrait page
Bottom bind shift	This value specifies that a shift of the page content is to be made in order to accommodate binding on the "bottom" of the physical medium independent of the actual "content-orientation" of the page, i.e., on the short edge of the medium corresponding to the bottom edge if the page were to be a portrait page
None	Specifies that no convenience imposition functions are requested. This instance would normally used in the argument of a ModifyJob operation to cancel a previously specified imposition

These attributes provide information about a particular imposition-function.

#### 9.14.1 Imposition-function-identifier

This attribute uniquely identifies the imposition-function.

imposition-function-identifier ATTRIBUTE  
 WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
 SINGLE VALUE  
 ::= id-att-imposition-function-identifier

Standard values:

id-val-imposition-simplex	id-val-imposition-duplex
id-val-imposition-tumble	id-val-imposition-2-up
id-val-imposition-4-up	id-val-imposition-left-bind-shift
id-val-imposition-right-bind-shift	id-val-imposition-top-bind-shift
id-val-imposition-bottom-bind-shift	id-val-imposition-none

The imposition specification value in combination with the current orientation determines the layout used.

9.14.2 Imposition-function-message

This attribute indicates the message associated with this object.

imposition-function-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-imposition-function-message

The message field may be set by an operator or system administrator to provide useful information to the potential users of this function.

9.15 Scheduler Object attributes

These attributes provide information about a particular scheduling algorithm.

9.15.1 Scheduler-identifier

This attribute uniquely identifies the scheduling algorithm.

scheduler-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-scheduler-identifier

9.15.2 Scheduler-message

This attribute indicates the message associated with this object.

scheduler-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-scheduler-message

The message field may be set by an operator or system administrator to provide useful information to the potential users of this object.

SECTION FOUR - DPA REALIZATION





## 10. OVERVIEW

Section four describes the procedures for the DPA and the port realization. It contains a description of the supply of the DP-user abstract-service in clause 11, and the supply of the DP-Administrator abstract-service is described in clause 12. The consumption sides of the DP-user and DP-Administrator ports are trivial and are not described in this Standard. The port realization in the form of service elements is described in clause 13.

The performance of the abstract-operations described in clause 11 and 12 shall be subject to the requirements of the security-policy (if one is in force), that applies to the DP abstract-services.

## 11. SUPPLY OF THE DP-USER ABSTRACT-SERVICE

This clause specifies how a DP-server supplies the DP-user abstract-service, covering the supply of the DP-user port.

### 11.1 Supply of the DP-user Port Abstract-services

This clause covers the supply of the Print, ModifyJob, CancelJob and ListObjectAttributes abstract-operations. The DP-user abstract-service supply of the DP-user port abstract-services assumes that an abstract-association exists between the DP-user port supplier (the DP-server) and the DP-user port consumer (the DP-user). The performance of the abstract-operations is in sequential order, no parallel processing takes place. Not all error cases are described.

#### 11.1.1 Performance of the Print Abstract-operation

When the DP-Server receives a Print request from the DP-User or the DP-Administrator, it performs the following steps:

- 1) Checks that the attributes in the supplied argument are valid for the Print abstract-operation. For details see 8.2.1.
- 2) If a security-policy is in force, ensure that such a policy is not violated by the performance of the Print abstract-operation.
- 3) Checks if the requested critical resources needed to perform the print-job are available, and if suitable alternatives can be offered for non-critical resources (e.g. printer type, media, fonts).
- 4) Creates a new print-job in the DP-server including the allocation of a print-job-id.
- 5) Checks whether the object to be printed was explicitly included in the print request, or whether some other form of transfer method is involved. If FTAM by client is to be used, DP-server obtains a file reference for an associated filestore, for return to the client.
- 6) Returns the Print result to the DP-user or DP-Administrator.
- 7) If the object(s) to be printed is to be retrieved by DP-Server, this function is performed. When all required file transfers have arrived at DP-Server, the requested print operation is completed.
- 8) Records any problems encountered after the print-job was accepted.

#### 11.1.2 Performance of the ModifyJob Abstract-operation

When the DP-Server receives a ModifyJob request from the DP-User or DP-Administrator, it performs the following steps:

- 1) Checks that the attributes in the supplied argument are valid for the ModifyJob abstract-operation including the print-job-id. This also includes checking that at-

tempts are not made to modify job attributes that are not modifiable. For details see 8.2.2.

- 2) If a security-policy is in force, ensure that such a policy is not violated by the proposed changes.
- 3) Checks that the proposed modification does not alter the requested critical or un-critical resources needed to perform the print-job in such a way that the print-request is invalid.
- 4) Checks that the print-job has not advanced so far in the processing cycle, that changes cannot be made.
- 5) Modifies the print-job attributes according to the request.
- 6) Returns the ModifyJob result to the DP-User or DP-Administrator.

#### 11.1.3 Performance of the CancelJob Abstract-operation

When the DP-Server receives a CancelJob request from the DP-User or DP-Administrator, it performs the following steps:

- 1) Checks that the parameters and attributes in the supplied argument are valid for the CancelJob abstract-operation. For details see 8.2.3.
- 2) Checks that a print-job with the specified print-job-id exists.
- 3) Checks that the request does not violate the security-policy, if a security-policy is in force.
- 4) Cancels the print-job. If the print-job is currently being printed, it will be interrupted and cancelled.
- 5) Returns the CancelJob result to the DP-User or DP-Administrator.

#### 11.1.4 Performance of the ListObjectAttributes Abstract-operation

When the DP-Server receives a ListObjectAttributes request from the DP-User or DP-Administrator, it performs the following steps:

- 1) Checks that the parameters and attributes in the supplied argument are valid for the ListObjectAttributes abstract-operation. For details see 8.2.4.
- 2) Checks if the specified object-class and object-instances (if specified) exist.
- 3) Checks that the requested attributes are appropriate for the object class.
- 4) Checks that the request does not violate the security-policy, if a security-policy is in force.
- 5) Retrieves the requested attribute-information from one or several objects.
- 6) Returns the ListObjectAttributes result to the DP-User or DP-Administrator.

### 12. SUPPLY OF THE DP-ADMINISTRATOR ABSTRACT-SERVICE

This clause specifies how a DP-server supplies the DP-administrator abstract service, covering the supply of the DP-administration port.

#### 12.1 Supply of the DP-administration Port Abstract-services

This clause covers the supply of the PromoteJob, Interrupt, Pause and Resume abstract operations. The DP-administrator abstract-service supply of the DP-administration port abstract-services assumes that an abstract-association exists between the DP-administration port supplier (the

DP-server) and the DP-administration port consumer (the DP-user). The performance of the abstract-operations is in sequential order, no parallel processing takes place. Not all error cases are described.

#### 12.1.1 Performance of the PromoteJob Abstract-operation

When the DP-Server receives a PromoteJob request from the DP-Administrator, it performs the following steps:

- 1) Checks that the parameters and attributes in the supplied argument are valid for the PromoteJob abstract-operation. For details see 8.3.1.
- 2) Checks if the job specified by the job-identifier exists.
- 3) Checks that the request does not violate the security-policy, if a security-policy is in force.
- 4) Schedules job for printing on the next available matching printer
- 5) Returns the PromoteJob result to the DP-Administrator.

#### 12.1.2 Performance of the Interrupt Abstract-operation

When the DP-Server receives an Interrupt request from the DP-Administrator, it performs the following steps:

- 1) Checks that the parameters and attributes in the supplied argument are valid for the Interrupt abstract-operation. For details see 8.3.2
- 2) Checks if the specified print-job or printer exists.
- 3) Checks that the request does not violate the security-policy, if a security-policy is in force.
- 4) Interrupts the specified printer or job, allows the specified interrupting job to print, and allows the interrupted job to resume printing when the interrupting job completes
- 5) Returns the Interrupt result to the DP-Administrator.

#### 12.1.3 Performance of the Pause Abstract-operation

When the DP-Server receives a Pause request from the DP-Administrator, it performs the following steps:

- 1) Checks that the parameters and attributes in the supplied argument are valid for the Pause abstract-operation. For details see 8.3.3
- 2) Checks if the specified object exists.
- 3) Checks that the request does not violate the security-policy, if a security-policy is in force.
- 4) Pauses the specified object.
- 5) Returns the Pause result to the DP-Administrator.

#### 12.1.4 Performance of the Resume Abstract-operation

When the DP-Server receives a Resume request from the DP-Administrator, it performs the following steps:

- 1) Checks that the parameters and attributes in the supplied argument are valid for the Resume abstract-operation. For details see 8.3.4.

- 2) Checks if the specified object exists.
- 3) Checks that the request does not violate the security-policy, if a security-policy is in force.
- 4) Resumes the specified object from being paused.
- 5) Returns the Resume result to the DP-Administrator.

### 13. PORT REALIZATION

The Document Printing abstract service is realized on a one-to-one basis between abstract operations defined in this part of the Standard and the real operations in the Document Printing Service Element specified in Part 2 of this Standard.

APPENDICES PART I



## APPENDIX A

## FORMAL ASSIGNMENT OF DPA STANDARD OBJECT IDENTIFIERS

All Object Identifiers assigned in this Part of the Standard are formally assigned in the present appendix using ASN.1. The specified values are cited in the ASN.1 modules of subsequent appendices.

This appendix is definitive for all values except those for ASN.1 modules of this Part of this Standard. The definitive assignments for those occur in the modules themselves.

```
DPAStandardObjectIdentifiers { iso-identified-organization(3) idc-ecma(0012) standard(0) dpa(140)
  modules (0) object-identifiers(0) }
```

```
DEFINITIONS IMPLICIT TAGS ::=
```

```
BEGIN
```

```
-- PROLOGUE --
```

```
EXPORTS -- everything --;
```

```
IMPORTS -- nothing --;
```

```
ID ::= OBJECT IDENTIFIER
```

```
id-dpa ID ::= { iso-identified-organization(3) idc-ecma(0012) standard(0) dpa(140) modules (0)
  object-identifiers(0) }
```

```
-- Categories --
```

```
id-mod ID ::= { id-dpa 0 } -- modules --
id-ot ID ::= { id-dpa 1 } -- objects --
id-pt ID ::= { id-dpa 2 } -- ports --
id-oc ID ::= { id-dpa 3 } -- object classes = attribute type classes --
id-syn ID ::= { id-dpa 5 } -- attribute syntaxes --
id-vc ID ::= { id-dpa 6 } -- attribute value classes --
```

```
-- Modules --
```

```
id-mod-object-identifiers ID ::= { id-mod 0 }
id-mod-abstract-service ID ::= { id-mod 1 }
id-mod-standard-attributes ID ::= { id-mod 2 }
id-mod-upper-bound ID ::= { id-mod 3 2 }
```

```
-- Objects --
```

```
id-dpa-server ID ::= { id-ot 0 }
id-dpa-user ID ::= { id-ot 1 }
```

```
-- Ports --
```

```
id-pa-dpa ID ::= { id-pt 0 }
```

## -- DPA Object Classes --

id-oc-job	ID ::= { id-oc 0 }
id-oc-printer	ID ::= { id-oc 1 }
id-oc-server	ID ::= { id-oc 2 }
id-oc-media	ID ::= { id-oc 3 }
id-oc-font	ID ::= { id-oc 4 }
id-oc-named-colour	ID ::= { id-oc 5 }
id-oc-transfer-method	ID ::= { id-oc 6 }
id-oc-delivery-method	ID ::= { id-oc 7 }
id-oc-job-sheet	ID ::= { id-oc 8 }
id-oc-finishing-method	ID ::= { id-oc 9 }
id-oc-output-method	ID ::= { id-oc 10 }
id-oc-imposition-function	ID ::= { id-oc 11 }
id-oc-scheduler	ID ::= { id-oc 12 }

## -- Attribute Syntaxes --

## -- Generic Object attributes --

id-syn-identifier	ID ::= { id-syn 0 }
id-syn-descriptor	ID ::= { id-syn 1 }
id-syn-name	ID ::= { id-syn 2 }
id-syn-aliases	ID ::= { id-syn 3 }
id-syn-state	ID ::= { id-syn 4 }
id-syn-message	ID ::= { id-syn 5 }
id-syn-availability	ID ::= { id-syn 6 }
id-syn-list-of-managers	ID ::= { id-syn 7 }
id-syn-text	ID ::= { id-syn 8 }

## -- Attribute Types --

## -- Job Object attributes --

## -- job-information --

id-att-job-identifier	ID ::= { id-oc-job 0 }
id-att-job-name	ID ::= { id-oc-job 1 }
id-att-job-originator	ID ::= { id-oc-job 2 }
id-att-job-owner	ID ::= { id-oc-job 3 }
id-att-job-comment	ID ::= { id-oc-job 4 }
id-att-job-start-message	ID ::= { id-oc-job 5 }
id-att-job-end-message	ID ::= { id-oc-job 6 }

## -- job-results-handling --

id-att-results-profile	ID ::= { id-oc-job 15 }
------------------------	-------------------------



-- job-event-handling --	
id-att-notification-profile	ID ::= { id-oc-job 20 }
id-att-logging-profile	ID ::= { id-oc-job 21 }
id-att-job-abort-criteria	ID ::= { id-oc-job 22 }
id-att-job-warning-criteria	ID ::= { id-oc-job 23 }
-- job-scheduling-instructions --	
id-att-job-start-wait	ID ::= { id-oc-job 30 }
id-att-job-end-wait	ID ::= { id-oc-job 31 }
id-att-job-hold	ID ::= { id-oc-job 32 }
id-att-job-password	ID ::= { id-oc-job 33 }
id-att-job-print-after	ID ::= { id-oc-job 34 }
id-att-job-priority	ID ::= { id-oc-job 35 }
id-att-job-deadline-time	ID ::= { id-oc-job 36 }
id-att-job-retention-time	ID ::= { id-oc-job 37 }
id-att-job-retention-period	ID ::= { id-oc-job 38 }
id-att-job-message-to-operator	ID ::= { id-oc-job 39 }
-- production-instructions --	
id-att-font-substitution	ID ::= { id-oc-job 50 }
id-att-media-substitution	ID ::= { id-oc-job 51 }
id-att-named-colour-substitution	ID ::= { id-oc-job 52 }
id-att-finishing-specification	ID ::= { id-oc-job 53 }
id-att-output-specification	ID ::= { id-oc-job 54 }
id-att-output-bin	ID ::= { id-oc-job 55 }
id-att-imposition-specification	ID ::= { id-oc-job 56 }
id-att-print-quality	ID ::= { id-oc-job 57 }
id-att-font-fidelity	ID ::= { id-oc-job 58 }
id-att-media-fidelity	ID ::= { id-oc-job 59 }
id-att-colour-fidelity	ID ::= { id-oc-job 60 }
id-att-sides	ID ::= { id-oc-job 61 }
id-att-page-select	ID ::= { id-oc-job 62 }
id-att-page-media-select	ID ::= { id-oc-job 63 }
id-att-copy-count	ID ::= { id-oc-job 64 }
id-att-start-sheet	ID ::= { id-oc-job 65 }
id-att-separator-sheet	ID ::= { id-oc-job 66 }
id-att-end-sheet	ID ::= { id-oc-job 67 }
id-att-reset-printer	ID ::= { id-oc-job 68 }
-- document-description --	
id-att-transfer-method	ID ::= { id-oc-job 80 }
id-att-document-format	ID ::= { id-oc-job 81 }
id-att-document-sequence-number	ID ::= { id-oc-job 82 }
id-att-document-content	ID ::= { id-oc-job 83 }
id-att-file-reference	ID ::= { id-oc-job 84 }

-- document attributes --

id-att-document-descriptor	ID ::= { id-oc-job 90 }
id-att-document-authors	ID ::= { id-oc-job 91 }
id-att-document-message	ID ::= { id-oc-job 92 }
id-att-document-revision-date	ID ::= { id-oc-job 93 }
id-att-content-formats-used	ID ::= { id-oc-job 94 }
id-att-page-order	ID ::= { id-oc-job 95 }
id-att-assured-reproduction-area	ID ::= { id-oc-job 96 }
id-att-page-size	ID ::= { id-oc-job 97 }
id-att-content-orientation	ID ::= { id-oc-job 98 }
id-att-page-independent	ID ::= { id-oc-job 99 }
id-att-character-sets-used	ID ::= { id-oc-job 100 }
id-att-character-mappings-used	ID ::= { id-oc-job 101 }
id-att-fonts-used	ID ::= { id-oc-job 102 }
id-att-character-repertoires-used	ID ::= { id-oc-job 103 }
id-att-media-used	ID ::= { id-oc-job 104 }
id-att-named-colours-used	ID ::= { id-oc-job 105 }
id-att-octet-count	ID ::= { id-oc-job 106 }
id-att-page-count	ID ::= { id-oc-job 107 }
id-att-glyph-count	ID ::= { id-oc-job 108 }
id-att-font-count	ID ::= { id-oc-job 109 }
id-att-font-change-count	ID ::= { id-oc-job 110 }
id-att-maximum-fonts-per-page	ID ::= { id-oc-job 111 }
id-att-percent-graphics	ID ::= { id-oc-job 112 }
id-att-percent-images	ID ::= { id-oc-job 113 }

-- security-access-accounting --

id-att-user-name	ID ::= { id-oc-job 120 }
id-att-accounting-information	ID ::= { id-oc-job 122 }

## -- job status --

id-att-current-job-state	ID ::= ( id-oc-job 200 )
id-att-previous-job-state	ID ::= ( id-oc-job 201 )
id-att-printers-assigned	ID ::= ( id-oc-job 202 )
id-att-estimated-completion-time	ID ::= ( id-oc-job 203 )
id-att-submission-time	ID ::= ( id-oc-job 204 )
id-att-modification-time	ID ::= ( id-oc-job 205 )
id-att-started-printing-time	ID ::= ( id-oc-job 206 )
id-att-copies-completed	ID ::= ( id-oc-job 207 )
id-att-pages-completed	ID ::= ( id-oc-job 208 )
id-att-octets-completed	ID ::= ( id-oc-job 209 )
id-att-intervening-pages	ID ::= ( id-oc-job 210 )
id-att-intervening-octets	ID ::= ( id-oc-job 211 )
id-att-intervening-jobs	ID ::= ( id-oc-job 212 )
id-att-print-checkpoint	ID ::= ( id-oc-job 213 )
id-att-on-request-resources-required	ID ::= ( id-oc-job 215 )
id-att-job-message-from-administrator	ID ::= ( id-oc-job 216 )
id-att-last-job-events	ID ::= ( id-oc-job 217 )
id-att-id-of-last-accessor	ID ::= ( id-oc-job 218 )
id-att-interrupt-at	ID ::= ( id-oc-job 219 )
id-att-pause-at	ID ::= ( id-oc-job 220 )
id-att-error-count	ID ::= ( id-oc-job 221 )
id-att-warning-count	ID ::= ( id-oc-job 222 )
id-att-processing-time	ID ::= ( id-oc-job 223 )

## -- Security handling --

id-att-job-security-handling	ID ::= ( id-oc-job 224 )
------------------------------	--------------------------

-- Printer Object attributes --

id-att-printer-identifier	ID ::= { id-oc-printer 0 }
id-att-printer-name	ID ::= { id-oc-printer 1 }
id-att-printer-location	ID ::= { id-oc-printer 2 }
id-att-printer-model	ID ::= { id-oc-printer 3 }
id-att-printer-type	ID ::= { id-oc-printer 4 }
id-att-printer-security	ID ::= { id-oc-printer 5 }
id-att-printer-speed	ID ::= { id-oc-printer 6 }
id-att-printer-resolution	ID ::= { id-oc-printer 7 }
id-att-printer-state	ID ::= { id-oc-printer 8 }
id-att-printer-message	ID ::= { id-oc-printer 9 }
id-att-attributes-supported-by-printer	ID ::= { id-oc-printer 10 }
id-att-object-classes-supported-by-printer	ID ::= { id-oc-printer 11 }
id-att-printers-supported	ID ::= { id-oc-printer 12 }
id-att-printers-active	ID ::= { id-oc-printer 13 }
id-att-fonts-supported	ID ::= { id-oc-printer 15 }
id-att-fonts-active	ID ::= { id-oc-printer 16 }
id-att-font-default	ID ::= { id-oc-printer 17 }
id-att-media-supported	ID ::= { id-oc-printer 21 }
id-att-media-active	ID ::= { id-oc-printer 22 }
id-att-medium-default	ID ::= { id-oc-printer 23 }
id-att-named-colours-supported	ID ::= { id-oc-printer 24 }
id-att-named-colours-active	ID ::= { id-oc-printer 25 }
id-att-named-colour-default	ID ::= { id-oc-printer 26 }
id-att-finishing-methods-supported	ID ::= { id-oc-printer 27 }
id-att-finishing-methods-active	ID ::= { id-oc-printer 28 }
id-att-finishing-methods-default	ID ::= { id-oc-printer 29 }
id-att-output-methods-supported	ID ::= { id-oc-printer 30 }
id-att-output-methods-active	ID ::= { id-oc-printer 31 }
id-att-output-methods-default	ID ::= { id-oc-printer 32 }
id-att-output-bins-supported	ID ::= { id-oc-printer 33 }
id-att-output-bins-active	ID ::= { id-oc-printer 34 }
id-att-output-bins-default	ID ::= { id-oc-printer 35 }
id-att-imposition-functions-supported	ID ::= { id-oc-printer 36 }
id-att-imposition-functions-enabled	ID ::= { id-oc-printer 37 }
id-att-imposition-functions-default	ID ::= { id-oc-printer 38 }
id-att-print-qualities-supported	ID ::= { id-oc-printer 39 }
id-att-print-qualities-active	ID ::= { id-oc-printer 40 }
id-att-print-quality-default	ID ::= { id-oc-printer 41 }
id-att-job-sheets-supported	ID ::= { id-oc-printer 42 }
id-att-job-sheets-active	ID ::= { id-oc-printer 43 }
id-att-start-sheet-default	ID ::= { id-oc-printer 44 }
id-att-separator-sheet-default	ID ::= { id-oc-printer 45 }

id-att-end-sheet-default	ID ::= ( id-oc-printer 46 )
id-att-character-sets-supported	ID ::= ( id-oc-printer 50 )
id-att-character-sets-active	ID ::= ( id-oc-printer 51 )
id-att-character-set-default	ID ::= ( id-oc-printer 52 )
id-att-character-mappings-supported	ID ::= ( id-oc-printer 53 )
id-att-character-mappings-active	ID ::= ( id-oc-printer 54 )
id-att-character-mapping-default	ID ::= ( id-oc-printer 55 )
id-att-character-repertoires-supported	ID ::= ( id-oc-printer 56 )
id-att-character-repertoires-active	ID ::= ( id-oc-printer 57 )
id-att-character-repertoire-default	ID ::= ( id-oc-printer 58 )
id-att-start-message-supported	ID ::= ( id-oc-printer 59 )
id-att-end-message-supported	ID ::= ( id-oc-printer 60 )
id-att-sides-supported	ID ::= ( id-oc-printer 61 )
id-att-sides-default	ID ::= ( id-oc-printer 62 )
id-att-page-select-supported	ID ::= ( id-oc-printer 63 )
id-att-maximum-copies-supported	ID ::= ( id-oc-printer 65 )
id-att-content-formats-supported	ID ::= ( id-oc-printer 66 )
id-att-content-format-default	ID ::= ( id-oc-printer 67 )
id-att-document-formats-supported	ID ::= ( id-oc-printer 68 )
id-att-document-format-default	ID ::= ( id-oc-printer 69 )
id-att-arcs-supported	ID ::= ( id-oc-printer 70 )
id-att-arc-default	ID ::= ( id-oc-printer 71 )
id-att-page-sizes-supported	ID ::= ( id-oc-printer 72 )
id-att-orientations-supported	ID ::= ( id-oc-printer 73 )
id-att-orientation-default	ID ::= ( id-oc-printer 74 )
id-att-page-order-assumed	ID ::= ( id-oc-printer 75 )
id-att-page-independence-assumed	ID ::= ( id-oc-printer 76 )
id-att-associated-server	ID ::= ( id-oc-printer 77 )
id-att-sheet-count	ID ::= ( id-oc-printer 78 )
id-att-printer-needs-attention-time	ID ::= ( id-oc-printer 79 )
id-att-printer-needs-key-operator-attention-time	ID ::= ( id-oc-printer 80 )

## -- Server Object attributes --

id-att-server-identifier	ID ::= { id-oc-server 0 }
id-att-server-name	ID ::= { id-oc-server 1 }
id-att-server-address	ID ::= { id-oc-server 2 }
id-att-server-state	ID ::= { id-oc-server 3 }
id-att-server-message	ID ::= { id-oc-server 4 }
id-att-object-classes-supported-by-server	ID ::= { id-oc-server 5 }
id-att-attributes-supported-by-server	ID ::= { id-oc-server 6 }
id-att-physical-printers-supported	ID ::= { id-oc-server 7 }
id-att-physical-printers-active	ID ::= { id-oc-server 8 }
id-att-logical-printers-supported	ID ::= { id-oc-server 9 }
id-att-logical-printers-active	ID ::= { id-oc-server 10 }
id-att-translators-supported	ID ::= { id-oc-server 11 }
id-att-translators-active	ID ::= { id-oc-server 12 }
id-att-delivery-methods-supported	ID ::= { id-oc-server 13 }
id-att-delivery-methods-active	ID ::= { id-oc-server 14 }
id-att-scheduling-algorithms-supported	ID ::= { id-oc-server 15 }
id-att-scheduling-algorithm-active	ID ::= { id-oc-server 16 }
id-att-system-hint-supported	ID ::= { id-oc-server 17 }

## -- Medium Object attributes --

id-att-medium-identifier	ID ::= { id-oc-media 0 }
id-att-medium-state	ID ::= { id-oc-media 1 }
id-att-medium-message	ID ::= { id-oc-media 2 }
id-att-medium-type	ID ::= { id-oc-media 3 }
id-att-medium-size	ID ::= { id-oc-media 4 }
id-att-medium-dimensions	ID ::= { id-oc-media 5 }
id-att-medium-colour	ID ::= { id-oc-media 6 }
id-att-medium-tooth	ID ::= { id-oc-media 7 }
id-att-medium-grain	ID ::= { id-oc-media 8 }
id-att-medium-sides	ID ::= { id-oc-media 9 }
id-att-medium-weight	ID ::= { id-oc-media 10 }
id-att-medium-assured-reproduction-area	ID ::= { id-oc-media 11 }
id-att-medium-associated-media	ID ::= { id-oc-media 12 }
id-att-medium-form-parts	ID ::= { id-oc-media 13 }

## -- Font Object attributes --

id-att-font-identifier	ID ::= { id-oc-font 0 }
id-att-font-type	ID ::= { id-oc-font 1 }
id-att-font-state	ID ::= { id-oc-font 2 }
id-att-font-message	ID ::= { id-oc-font 3 }

-- <i>Named Colour Object attributes</i> --	
id-att-named-colour-identifier	ID ::= { id-oc-named-colour 0 }
id-att-named-colour-state	ID ::= { id-oc-named-colour 1 }
id-att-named-colour-message	ID ::= { id-oc-named-colour 2 }
-- <i>Transfer Method Object attributes</i> --	
id-att-transfer-method-identifier	ID ::= { id-oc-transfer-method 0 }
id-att-document-reference-type	ID ::= { id-oc-transfer-method 2 }
id-att-transfer-method-message	ID ::= { id-oc-transfer-method 3 }
-- <i>Delivery Method Object attributes</i> --	
id-att-delivery-method-identifier	ID ::= { id-oc-delivery-method 0 }
id-att-delivery-method-message	ID ::= { id-oc-delivery-method 1 }
-- <i>Job Sheet Object attributes</i> --	
id-att-job-sheet-identifier	ID ::= { id-oc-job-sheet 0 }
id-att-job-sheet-message	ID ::= { id-oc-job-sheet 1 }
-- <i>Finishing Method Object attributes</i> --	
id-att-finishing-method-identifier	ID ::= { id-oc-finishing-method 0 }
id-att-finishing-method-message	ID ::= { id-oc-finishing-method 1 }
-- <i>Output Method Object attributes</i> --	
id-att-output-method-identifier	ID ::= { id-oc-output-method 0 }
id-att-output-method-message	ID ::= { id-oc-output-method 1 }
-- <i>Imposition Function Object attributes</i> --	
id-att-imposition-function-identifier	ID ::= { id-oc-imposition-function 0 }
id-att-imposition-function-message	ID ::= { id-oc-imposition-function 1 }
-- <i>Scheduler Object attributes</i> --	
id-att-scheduler-identifier	ID ::= { id-oc-scheduler 0 }
id-att-scheduler-message	ID ::= { id-oc-scheduler 1 }
-- <i>DPA Attribute Value Classes</i> --	
-- <i>Generic Object</i> --	
id-vc-state	ID ::= { id-vc 0 }
id-vc-availability	ID ::= { id-vc 1 }
-- <i>Job Object</i> --	
id-vc-print-quality	ID ::= { id-vc 10 }
id-vc-fore-fidelity	ID ::= { id-vc 11 }
id-vc-media-fidelity	ID ::= { id-vc 12 }
id-vc-colour-fidelity	ID ::= { id-vc 13 }
id-vc-document-format	ID ::= { id-vc 14 }
id-vc-content-format	ID ::= { id-vc 15 }
id-vc-content-orientation	ID ::= { id-vc 16 }
id-vc-page-order	ID ::= { id-vc 17 }
id-vc-job-state	ID ::= { id-vc 18 }

```

-- Printer Object --
id-vc-printer-state          ID ::= { id-vc 30 }
id-vc-printer-type          ID ::= { id-vc 31 }
id-vc-printer-security      ID ::= { id-vc 32 }
id-vc-printer-output-bin    ID ::= { id-vc 33 }

-- Server Object --
-- server-state (uses only generic object state values) --

-- Medium Object --
id-vc-medium                 ID ::= { id-vc 50 }

-- Medium-state (uses only generic object state values) --
id-vc-medium-type           ID ::= { id-vc 52 }
id-vc-medium-size          ID ::= { id-vc 53 }
id-vc-medium-colour        ID ::= { id-vc 54 }
id-vc-medium-form-type     ID ::= { id-vc 55 }

-- Font Object --
-- font-state (uses only generic object state values) --

-- Named Colour Object --
id-vc-named-colour          ID ::= { id-vc 65 }
id-vc-named-colour-type     ID ::= { id-vc 66 }
id-vc-named-colour-transparency ID ::= { id-vc 67 }

-- Transfer Method Object --
id-vc-transfer-method       ID ::= { id-vc 75 }
id-vc-transfer-method-type  ID ::= { id-vc 76 }
id-vc-document-rel-type     ID ::= { id-vc 77 }

-- Delivery Method Object --
id-vc-delivery-method       ID ::= { id-vc 80 }

-- Job Sheet Object --
id-vc-job-sheet             ID ::= { id-vc 85 }

-- Finishing Method Object --
id-vc-finishing-method      ID ::= { id-vc 90 }

-- Output Method Object --
id-vc-output-method         ID ::= { id-vc 95 }

-- Imposition Function Object --
id-vc-imposition-method     ID ::= { id-vc 100 }

-- Scheduler Object --
id-vc-scheduler             ID ::= { id-vc 105 }

```



-- Events --

id-vc-event-id	ID ::= { id-vc 120 }
id-vc-abort-event	ID ::= { id-vc 121 }
id-vc-error-event	ID ::= { id-vc 122 }
id-vc-warning-event	ID ::= { id-vc 123 }
id-vc-report-event	ID ::= { id-vc 124 }
id-vc-other-event	ID ::= { id-vc 125 }

-- Generic Object attribute values --

-- state --

id-val-state-available	ID ::= { id-vc-state 0 }
id-val-state-busy	ID ::= { id-vc-state 1 }
id-val-state-on-request	ID ::= { id-vc-state 2 }
id-val-state-unavailable	ID ::= { id-vc-state 3 }
id-val-state-unknown	ID ::= { id-vc-state 4 }

-- availability --

id-val-availability-immediate	ID ::= { id-vc-availability 0 }
id-val-availability-high	ID ::= { id-vc-availability 1 }
id-val-availability-normal	ID ::= { id-vc-availability 2 }
id-val-availability-low	ID ::= { id-vc-availability 3 }
id-val-availability-restricted	ID ::= { id-vc-availability 4 }
id-val-availability-none	ID ::= { id-vc-availability 5 }
id-val-availability-unknown	ID ::= { id-vc-availability 6 }

-- Job Object attribute values --

-- print-quality --

id-val-print-quality-draft	ID ::= { id-vc-print-quality 0 }
id-val-print-quality-normal	ID ::= { id-vc-print-quality 1 }
id-val-print-quality-high	ID ::= { id-vc-print-quality 2 }

-- media-fidelity --

id-val-media-fidelity-absolute	ID ::= { id-vc-media-fidelity 0 }
id-val-media-fidelity-colour	ID ::= { id-vc-media-fidelity 1 }
id-val-media-fidelity-size	ID ::= { id-vc-media-fidelity 2 }
id-val-media-fidelity-reproduction-area	ID ::= { id-vc-media-fidelity 3 }
id-val-media-fidelity-any	ID ::= { id-vc-media-fidelity 4 }

-- colour-fidelity --

id-val-colour-fidelity-absolute	ID ::= { id-vc-colour-fidelity 0 }
id-val-colour-fidelity-colour	ID ::= { id-vc-colour-fidelity 1 }
id-val-colour-fidelity-distinct	ID ::= { id-vc-colour-fidelity 2 }
id-val-colour-fidelity-any	ID ::= { id-vc-colour-fidelity 3 }

-- document format --

-- content-orientation --

id-val-content-orientation-portrait	ID ::= { id-vc-content-orientation 0 }
id-val-content-orientation-landscape	ID ::= { id-vc-content-orientation 1 }
id-val-content-orientation-mixed	ID ::= { id-vc-content-orientation 2 }

-- page-size --

-- page-order --

id-val-page-order-forward	ID ::= { id-vc-page-order 0 }
id-val-page-order-reverse	ID ::= { id-vc-page-order 1 }
id-val-page-order-signature	ID ::= { id-vc-page-order 2 }

-- job-state --

id-val-job-state-unknown	ID ::= { id-vc-job-state 0 }
id-val-job-state-pre-processing	ID ::= { id-vc-job-state 1 }
id-val-job-state-document-transfer-pending	ID ::= { id-vc-job-state 2 }
id-val-job-state-transferring	ID ::= { id-vc-job-state 3 }
id-val-job-state-pending	ID ::= { id-vc-job-state 4 }
id-val-job-state-processing	ID ::= { id-vc-job-state 5 }
id-val-job-state-interrupted	ID ::= { id-vc-job-state 6 }
id-val-job-state-logfile-transfer-pending	ID ::= { id-vc-job-state 7 }
id-val-job-state-logfile-transferring	ID ::= { id-vc-job-state 8 }
id-val-job-state-retained	ID ::= { id-vc-job-state 9 }
id-val-job-state-held	ID ::= { id-vc-job-state 10 }
id-val-job-state-paused	ID ::= { id-vc-job-state 11 }
id-val-job-state-aborting	ID ::= { id-vc-job-state 12 }

-- Printer Object attribute valact --

-- printer-type --

id-val-printer-type-line	ID ::= { id-vc-printer-type 0 }
id-val-printer-type-letter-quality	ID ::= { id-vc-printer-type 1 }
id-val-printer-type-laser	ID ::= { id-vc-printer-type 2 }
id-val-printer-type-colour-laser	ID ::= { id-vc-printer-type 3 }
id-val-printer-type-microfiche-imager	ID ::= { id-vc-printer-type 4 }
id-val-printer-type-plotter	ID ::= { id-vc-printer-type 5 }
id-val-printer-type-ink-jet	ID ::= { id-vc-printer-type 6 }
id-val-printer-type-colour-ink-jet	ID ::= { id-vc-printer-type 7 }
id-val-printer-type-dot-matrix	ID ::= { id-vc-printer-type 8 }
id-val-printer-type-colour-dot-matrix	ID ::= { id-vc-printer-type 9 }
id-val-printer-type-daisy-wheel	ID ::= { id-vc-printer-type 10 }

-- printer-security --

id-val-printer-security-unclassified	ID ::= (id-vc-printer-security 0 )
id-val-printer-security-classified	ID ::= (id-vc-printer-security 1 )
id-val-printer-security-secret	ID ::= (id-vc-printer-security 2 )
id-val-printer-security-top-secret	ID ::= (id-vc-printer-security 3 )
id-val-printer-security-confidential	ID ::= (id-vc-printer-security 4 )
id-val-printer-security-personal	ID ::= (id-vc-printer-security 5 )

-- printer-state --

id-val-printer-state-unknown	ID ::= (id-vc-printer-state 0 )
id-val-printer-state-idle	ID ::= (id-vc-printer-state 1 )
id-val-printer-state-printing	ID ::= (id-vc-printer-state 2 )
id-val-printer-state-needs-attention	ID ::= (id-vc-printer-state 3 )
id-val-printer-state-paused	ID ::= (id-vc-printer-state 4 )
id-val-printer-state-shutdown	ID ::= (id-vc-printer-state 5 )

-- printer-output-bin --

id-val-output-bin-top	ID ::= (id-vc-printer-output-bin 0 )
id-val-output-bin-middle	ID ::= (id-vc-printer-output-bin 1 )
id-val-output-bin-bottom	ID ::= (id-vc-printer-output-bin 2 )
id-val-output-bin-side	ID ::= (id-vc-printer-output-bin 3 )
id-val-output-bin-face-up	ID ::= (id-vc-printer-output-bin 4 )
id-val-output-bin-upper	ID ::= (id-vc-printer-output-bin 5 )
id-val-output-bin-lower	ID ::= (id-vc-printer-output-bin 6 )
id-val-output-bin-large-capacity	ID ::= (id-vc-printer-output-bin 7 )
id-val-output-bin-private	ID ::= (id-vc-printer-output-bin 8 )
id-val-output-bin-secure	ID ::= (id-vc-printer-output-bin 9 )

-- Server Object attributes values --

-- server-state (uses only generic object values) --

-- Media Object attribute values --

-- medium --

id-val-medium-iso-a4-white	ID ::= { id-vc-medium 0 }
id-val-medium-iso-a4-coloured	ID ::= { id-vc-medium 1 }
id-val-medium-iso-a4-transparent	ID ::= { id-vc-medium 2 }
id-val-medium-iso-a3-white	ID ::= { id-vc-medium 10 }
id-val-medium-iso-a3-coloured	ID ::= { id-vc-medium 11 }
id-val-medium-iso-a5-white	ID ::= { id-vc-medium 20 }
id-val-medium-iso-a5-coloured	ID ::= { id-vc-medium 21 }
id-val-medium-iso-b4-white	ID ::= { id-vc-medium 30 }
id-val-medium-iso-b4-coloured	ID ::= { id-vc-medium 31 }
id-val-medium-iso-b5-white	ID ::= { id-vc-medium 40 }
id-val-medium-iso-b5-coloured	ID ::= { id-vc-medium 41 }
id-val-medium-north-american-letter-white	ID ::= { id-vc-medium 50 }
id-val-medium-north-american-letter-coloured	ID ::= { id-vc-medium 51 }
id-val-medium-north-american-letter-transparent	ID ::= { id-vc-medium 52 }
id-val-medium-north-american-legal-white	ID ::= { id-vc-medium 60 }
id-val-medium-north-american-legal-coloured	ID ::= { id-vc-medium 61 }
id-val-medium-iso-b5-envelope	ID ::= { id-vc-medium 73 }
id-val-medium-iso-b4-envelope	ID ::= { id-vc-medium 83 }
id-val-medium-c4-envelope	ID ::= { id-vc-medium 93 }
id-val-medium-c5-envelope	ID ::= { id-vc-medium 103 }
id-val-medium-iso-designated-long-envelope	ID ::= { id-vc-medium 113 }
id-val-medium-north-american-10x13-envelope	ID ::= { id-vc-medium 123 }
id-val-medium-north-american-9x12-envelope	ID ::= { id-vc-medium 133 }
id-val-medium-north-american-business-envelope	ID ::= { id-vc-medium 143 }

-- medium-state (uses only generic object state values) --

-- medium-type --

id-val-medium-type-stationery	ID ::= { id-vc-medium-type 0 }
id-val-medium-type-transparency	ID ::= { id-vc-medium-type 1 }
id-val-medium-type-envelope	ID ::= { id-vc-medium-type 2 }
id-val-medium-type-form	ID ::= { id-vc-medium-type 2 }
id-val-medium-type-continuous	ID ::= { id-vc-medium-type 2 }

-- medium-colour --

-- medium-size --

-- medium-form-type --

id-val-medium-form-type-electronic	ID ::= { id-vc-medium-form-type 0 }
id-val-medium-form-type-preprinted	ID ::= { id-vc-medium-form-type 1 }

-- Font Object attribute values --

-- font-state (uses only generic object values) --

-- Named Colour Object attribute values --

-- colour-state (uses only generic object state values) --

-- Transfer Method object attribute values --

-- transfer-method --

id-val-transfer-method-transfer-document-with-job-request ID ::= { id-vc-transfer-method 0 }

id-val-transfer-method-referenced-data-transfer ID ::= { id-vc-transfer-method 1 }

id-val-transfer-method-ftp-by-client ID ::= { id-vc-transfer-method 2 }

id-val-transfer-method-ftp-by-server ID ::= { id-vc-transfer-method 3 }

-- Delivery Method object attribute values --

-- delivery-method --

id-val-delivery-method-pick-up ID ::= { id-vc-delivery-method 0 }

id-val-delivery-method-secure-pickup ID ::= { id-vc-delivery-method 1 }

id-val-delivery-method-interoffice-mail ID ::= { id-vc-delivery-method 2 }

id-val-delivery-method-postal-mail ID ::= { id-vc-delivery-method 3 }

id-val-delivery-method-notification-service ID ::= { id-vc-delivery-method 4 }

id-val-delivery-method-electronic-mail ID ::= { id-vc-delivery-method 5 }

id-val-delivery-method-file ID ::= { id-vc-delivery-method 6 }

id-val-delivery-method-none ID ::= { id-vc-delivery-method 7 }

-- Job Sheet object attribute values --

-- job-sheet --

id-val-job-sheet-none ID ::= { id-vc-job-sheet 0 }

id-val-job-sheet-brief ID ::= { id-vc-job-sheet 1 }

id-val-job-sheet-full ID ::= { id-vc-job-sheet 2 }

-- Finishing Method object attribute values --

-- Finishing method --

id-val-finishing-staples	ID ::= (id-vc-finishing-method 0 )
id-val-finishing-staples-top	ID ::= (id-vc-finishing-method 1 )
id-val-finishing-staples-bottom	ID ::= (id-vc-finishing-method 2 )
id-val-finishing-staples-left	ID ::= (id-vc-finishing-method 3 )
id-val-finishing-staples-right	ID ::= (id-vc-finishing-method 4 )
id-val-finishing-staples-top-left	ID ::= (id-vc-finishing-method 5 )
id-val-finishing-staples-bottom-left	ID ::= (id-vc-finishing-method 6 )
id-val-finishing-staples-top-right	ID ::= (id-vc-finishing-method 7 )
id-val-finishing-staples-bottom-right	ID ::= (id-vc-finishing-method 9 )
id-val-finishing-staples-none	ID ::= (id-vc-finishing-method 10 )
id-val-finishing-holes	ID ::= (id-vc-finishing-method 20 )
id-val-finishing-holes-top	ID ::= (id-vc-finishing-method 21 )
id-val-finishing-holes-2-top	ID ::= (id-vc-finishing-method 22 )
id-val-finishing-holes-3-top	ID ::= (id-vc-finishing-method 23 )
id-val-finishing-holes-4-top	ID ::= (id-vc-finishing-method 24 )
id-val-finishing-holes-left	ID ::= (id-vc-finishing-method 25 )
id-val-finishing-holes-2-left	ID ::= (id-vc-finishing-method 26 )
id-val-finishing-holes-3-left	ID ::= (id-vc-finishing-method 27 )
id-val-finishing-holes-4-left	ID ::= (id-vc-finishing-method 28 )
id-val-finishing-holes-right	ID ::= (id-vc-finishing-method 29 )
id-val-finishing-holes-2-right	ID ::= (id-vc-finishing-method 30 )
id-val-finishing-holes-3-right	ID ::= (id-vc-finishing-method 31 )
id-val-finishing-holes-4-right	ID ::= (id-vc-finishing-method 32 )
id-val-finishing-holes-none	ID ::= (id-vc-finishing-method 33 )
id-val-finishing-cover	ID ::= (id-vc-finishing-method 40 )
id-val-finishing-nocover	ID ::= (id-vc-finishing-method 41 )
id-val-finishing-bind	ID ::= (id-vc-finishing-method 50 )
id-val-finishing-bind-top	ID ::= (id-vc-finishing-method 51 )
id-val-finishing-bind-left	ID ::= (id-vc-finishing-method 52 )
id-val-finishing-bind-right	ID ::= (id-vc-finishing-method 53 )
id-val-finishing-bind-bottom	ID ::= (id-vc-finishing-method 54 )
id-val-finishing-bind-none	ID ::= (id-vc-finishing-method 55 )

## -- Output Method object attribute values --

## -- Output-method --

id-val-output-page-collate	ID ::= (id-vc-output-method 0)
id-val-output-nopage-collate	ID ::= (id-vc-output-method 1)
id-val-output-document-collate	ID ::= (id-vc-output-method 2)
id-val-output-nodocument-collate	ID ::= (id-vc-output-method 3)
id-val-output-decollate	ID ::= (id-vc-output-method 4)
id-val-output-nodecollate	ID ::= (id-vc-output-method 5)
id-val-output-bursting	ID ::= (id-vc-output-method 6)
id-val-output-nobursting	ID ::= (id-vc-output-method 7)

## -- Imposition Function object attribute values --

## -- Imposition-methods --

id-val-imposition-simplex	ID ::= (id-vc-imposition-method 0)
id-val-imposition-duplex	ID ::= (id-vc-imposition-method 1)
id-val-imposition-tumble	ID ::= (id-vc-imposition-method 2)
id-val-imposition-2-up	ID ::= (id-vc-imposition-method 3)
id-val-imposition-4-up	ID ::= (id-vc-imposition-method 4)
id-val-imposition-left-bind-shift	ID ::= (id-vc-imposition-method 5)
id-val-imposition-right-bind-shift	ID ::= (id-vc-imposition-method 6)
id-val-imposition-top-bind-shift	ID ::= (id-vc-imposition-method 7)
id-val-imposition-bottom-bind-shift	ID ::= (id-vc-imposition-method 8)
id-val-imposition-none	ID ::= (id-vc-imposition-method 9)

## -- Scheduler object attribute values --

## -- EVENTS --

## -- Event Identifiers --

id-val-abort-event-id	ID ::= (id-vc-event-id 0)
id-val-error-event-id	ID ::= (id-vc-event-id 1)
id-val-warning-event-id	ID ::= (id-vc-event-id 2)
id-val-report-event-id	ID ::= (id-vc-event-id 3)
id-val-other-event-id	ID ::= (id-vc-event-id 4)

## -- Abort identifiers --

id-val-abort-event-by-server	ID ::= (id-vc-abort-event 0)
id-val-abort-event-by-operator	ID ::= (id-vc-abort-event 1)
id-val-abort-event-by-user	ID ::= (id-vc-abort-event 2)

## -- Error identifiers --

id-val-error-event-deadline	ID ::= (id-vc-error-event 3)
id-val-error-event-retention	ID ::= (id-vc-error-event 4)
id-val-error-event-shutdown	ID ::= (id-vc-error-event 5)
id-val-error-event-noresource	ID ::= (id-vc-error-event 6)
id-val-error-event-name-unrecognized	ID ::= (id-vc-error-event 7)

```
-- Warning events --
id-val-warning-event-res-attn          ID ::= { id-vc-warning-event 0 }
id-val-warning-event-res-operator      ID ::= { id-vc-warning-event 1 }
id-val-warning-event-pr-shutdown       ID ::= { id-vc-warning-event 2 }
id-val-warning-event-modified          ID ::= { id-vc-warning-event 3 }
id-val-warning-event-close-to-deadline ID ::= { id-vc-warning-event 4 }
id-val-warning-event-close-to-retentiontime ID ::= { id-vc-warning-event 5 }
id-val-warning-event-criterion-exceeded ID ::= { id-vc-warning-event 6 }

-- Report events --
id-val-report-event-completed          ID ::= { id-vc-report-event 0 }
id-val-report-event-checkpoint         ID ::= { id-vc-report-event 1 }
id-val-report-event-resource           ID ::= { id-vc-report-event 2 }
id-val-report-event-discarded          ID ::= { id-vc-report-event 3 }
id-val-report-event-file-translated    ID ::= { id-vc-report-event 4 }

-- Other Events --
id-val-other-event-unknown             ID ::= { id-vc-other-event 0 }

END -- of DPA Objectclassifiers --
```



## APPENDIX B

### FORMAL DEFINITION OF THE DPA ABSTRACT SERVICE

This Appendix, a supplement to Section 2, formally defines the Document Printing abstract service. It employs ASN.1 and the OBJECT, PORT, ABSTRACT-BIND, ABSTRACT-UNBIND, ABSTRACT-OPERATION, and ABSTRACT-ERROR macros of ISO/IEC 10021-3.

---

DPAAbstractService { iso-identified-organization(3) idc-ecma(0012) standard(0) dpa(140) modules(0) abstract-service(1) }

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

– PROLOGUE –

EXPORTS – everything – ;

IMPORTS

– Abstract service macros –

ABSTRACT-BIND, ABSTRACT-ERROR, ABSTRACT-OPERATION,  
ABSTRACT-UNBIND, OBJECT, PORT

FROM AbstractServiceNotation

{ joint-iso-ccitt mhs-mots(6) asdc(2) modules(0) notation(1) }

– OST Directory –

Attribute, AttributeType, AttributeValue,  
AttributeValueAssertion, DistinguishedName

FROM InformationFramework

{ joint-iso-ccitt ds(5) modules(1) informationFramework(1) }

Filter, FilterItem

FROM DirectoryAbstractService

{ joint-iso-ccitt ds(5) modules(1) directoryAbstractService(2) }

– DPA object identifiers –

id-at-accounting-information, id-at-assured-reproduction-area,  
id-at-character-mappings-used, id-at-character-repertoires-used,  
id-at-character-sets-used, id-at-colour-fidelity, id-at-content-orientation,  
id-at-copies-completed, id-at-copy-count, id-at-current-job-state,  
id-at-delivery-method-identifier, id-at-document-authors, id-at-document-content,  
id-at-document-descriptor, id-at-document-format, id-at-document-message,  
id-at-document-revision-date, id-at-document-sequence-number, id-at-end-sheet,  
id-at-estimated-completion-time, id-at-file-reference, id-at-finishing-method-identifier,  
id-at-finishing-specification, id-at-fonts-used, id-at-foot-change-count, id-at-foot-count,  
id-at-font-fidelity, id-at-font-identifier, id-at-font-substitution, id-at-glyph-count,  
id-at-id-of-last-accessor, id-at-imposition-function-identifier,

id-att-imposition-specification, id-att-interrupt-at, id-att-intervening-jobs,  
id-att-intervening-octets, id-att-intervening-pages, id-att-job-abort-criteria,  
id-att-job-comment, id-att-job-deadline-time, id-att-job-end-message, id-att-job-end-wait,  
id-att-job-hold, id-att-job-identifier, id-att-job-message-from-administrator,  
id-att-job-message-to-operator, id-att-job-name, id-att-job-originator, id-att-job-owner,  
id-att-job-password, id-att-job-print-after, id-att-job-priority, id-att-job-retention-period,  
id-att-job-retention-time, id-att-job-security-handling, id-att-job-sheet-identifier,  
id-att-job-start-message, id-att-job-start-wait, id-att-job-warming-criteria,  
id-att-last-job-events, id-att-logging-profile, id-att-maximum-forts-per-page,  
id-att-media-fidelity, id-att-media-substitution, id-att-media-used, id-att-medium-identifier,  
id-att-modification-time, id-att-named-colours-used, id-att-named-colour-identifier,  
id-att-named-colour-substitution, id-att-new-server-name, id-att-notification-profile,  
id-att-octets-completed, id-att-octet-count, id-att-on-request-resources-required,  
id-att-output-bin, id-att-output-method-identifier, id-att-output-specification,  
id-att-pages-completed, id-att-page-count, id-att-page-independent,  
id-att-page-media-select, id-att-page-order, id-att-page-select, id-att-page-size,  
id-att-pause-at, id-att-percent-graphics, id-att-percent-images, id-att-previous-job-state,  
id-att-printers-assigned, id-att-printer-identifier, id-att-printer-location,  
id-att-printer-message, id-att-printer-model, id-att-printer-resolution, id-att-printer-security,  
id-att-printer-speed, id-att-printer-type, id-att-print-checkpoint, id-att-print-quality,  
id-att-reset-printer, id-att-results-profile, id-att-scheduler-identifier, id-att-separator-sheet,  
id-att-server-identifier, id-att-server-message, id-att-server-state, id-att-sides,  
id-att-started-printing-time, id-att-start-sheet, id-att-submission-time,  
id-att-transfer-method, id-att-transfer-method-identifier, id-att-user-name,  
id-dp-administrator, id-dp-server, id-dp-user, id-pt-dp-administration, id-pt-dp-user,

FROM DPASStandardObjectIdentifiers

( iso identified-organization(3) idc-ecma(0012) standard(0) dpa(140) modules(0)  
object-identifiers(0) );

FROM DPASStandardObjectIdentifiers

( iso identified-organization(3) idc-ecma(0012) standard(0) dpa(140) modules(0)  
object-identifiers(0) );

-- DPA upper bound --

ub-descriptor-string, ub-integer, ub-message-string, ub-text-string, ub-name-string

FROM DPAUpperBound

( iso identified-organization(3) idc-ecma(0012) standard(0) dpa(140) modules(0)  
upper-bound(2) );

-- DPA ABSTRACT OBJECTS --

dp-serverOBJECT

PORTS { dp-user [S],  
dp-administration [S] }

::= id-dp-server

```
dp-user OBJECT
  PORTS { dp-user [C] }
  ::= id-dp-user

dp-administrator OBJECT
  PORTS { dp-user [C],
          dp-administration [C] }
  ::= id-dp-administrator

-- DPA ABSTRACT PORTS --
dp-user PORT
  CONSUMER INVOKES {
    Print,
    ModifyJob,
    CancelJob,
    ListObjectAttributes }
  ::= id-pt-dp-user

dp-administration PORT
  CONSUMER INVOKES {
    PromoteJob,
    Interrupt,
    Pause,
    Resume }
  ::= id-pt-dp-administration

-- DPA session data types --
JobIdentifier ::= SEQUENCE {
  server-identifier [0] OBJECT IDENTIFIER,
  local-identifier [1] OCTET STRING -- implementation dependent -- }

QualifiedAttribute ::= SEQUENCE {
  attribute Attribute,
  qualifier Compulsory DEFAULT FALSE}

Compulsory ::= BOOLEAN

DpAccessListElement ::= SEQUENCE {
  access-id AccessId,
  access-rights DpAccessRights }

AccessId ::= DistinguishedName

DpAccessRights ::= ENUMERATED {
  dp-user [0],
  dp-administrator [1] }
```

-- ABSTRACT BIND UNBIND --

DpBind ::= ABSTRACT-BIND  
TO ( dp-user[S], dp-administration[S] )  
BIND  
ARGUMENT DpBindArgument  
RESULT DpBindResult  
BIND-ERROR DpBindError

DpBindArgument ::= SEQUENCE {  
credentials [0] Credentials,  
retrieve-restrictions [1] Restrictions OPTIONAL  
--default is none--  
bind-security [2] BindSecurity OPTIONAL }

Credentials ::= CHOICE {  
simple [0] Creds,  
-- used for initial authentication --  
certified [1] PrivilegeAttributeCertificate  
-- used when initial authentication has already taken place external to the DP-Server -- }

PrivilegeAttributeCertificate ::= EXTERNAL

Restrictions ::= SET {  
maximum-result-length [1] ResultLength OPTIONAL  
--default is no restriction-- }

ResultLength ::= INTEGER

BindSecurity ::= EXTERNAL

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

AuthenticationAttribute ::= EXTERNAL

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

DpUnbind ::= ABSTRACT-UNBIND  
FROM ( dp-user[S], dp-administration[S] )

-- ABSTRACT OPERATIONS --

-- Common arguments --

CommonArguments ::= SET {  
    extensions [25] SET OF QualifiedAttribute OPTIONAL,  
    error-handling [27] ErrorHandlingMode DEFAULT all-or-nothing,  
    priority [28] Priority DEFAULT medium,  
    privileges [30] Privileges DEFAULT ( ) }

ErrorHandlingMode ::= ENUMERATED {  
    all-or-nothing (0),  
    until-first-error (1),  
    do-maximum (2) }

Priority ::= ENUMERATED {  
    low (0),  
    medium (1),  
    high (2) }

Privileges ::= SEQUENCE {  
    operation-pac [0] PrivilegeAttributeCertificate OPTIONAL,  
    proxy-pac [1] PrivilegeAttributeCertificate OPTIONAL }

PrivilegeAttributeCertificate ::= EXTERNAL

-- Common results --

CommonResults ::= SEQUENCE {  
    extensions [25] SET OF QualifiedAttribute OPTIONAL }

-- ABSTRACT OPERATIONS --

-- Print --

Print ::= ABSTRACT-OPERATION  
    ARGUMENT PrintArgument  
    RESULT PrintResult  
    ERRORS { AttributeError,  
            DocumentAccessError,  
            PrinterError,  
            SecurityError,  
            SelectionError,  
            ServiceError }

PrintArgument ::= SEQUENCE {  
    job-attributes [0] SET OF JobAttribute OPTIONAL,  
    document-description [1] SET OF DocumentDescription,  
    -- This set must include all possible components of DocumentDescription  
    COMPONENTS OF CommonArguments }

```
JobAttribute ::= QualifiedAttribute
(WITH COMPONENTS {
  attribute
  (WITH COMPONENTS {
    type ( id-at-job-identifier
          | id-at-document-sequence-number
          | INCLUDES JobInformationTypes
          | INCLUDES JobResultsHandlingTypes
          | INCLUDES JobEventHandlingTypes
          | INCLUDES JobSchedulingInstructionsTypes
          | INCLUDES PrinterIdentificationTypes
          | INCLUDES ProductionInstructionsTypes
          | INCLUDES DocumentAttributeTypes
          | INCLUDES AccessAndAccountingTypes
          | INCLUDES JobSecurityTypes ),
    values
    qualifier
  })
}
```

```
DocumentDescription ::= Attribute
(WITH COMPONENTS {
  type ( id-at-transfer-method
        | id-at-document-format
        | id-at-document-content ),
  values
}
```

DocumentAttributeTypes ::= AttributeType

- ( id-att-content-formats-used
- | id-att-assured-reproduction-area
- | id-att-content-orientation
- | id-att-page-order
- | id-att-page-size
- | id-att-page-independent
- | id-att-character-sets-used
- | id-att-character-mappings-used
- | id-att-fonts-used
- | id-att-character-repertoires-used
- | id-att-media-used
- | id-att-named-colours-used
- | id-att-document-descriptor
- | id-att-document-authors
- | id-att-document-message
- | id-att-document-revision-data
- | id-att-octet-count
- | id-att-page-count
- | id-att-glyph-count
- | id-att-foet-count
- | id-att-foet-change-count
- | id-att-maximum-fonts-per-page
- | id-att-percent-graphics
- | id-att-percent-images)

JobEventHandlingTypes ::= AttributeType

- ( id-att-notification-profile
- | id-att-logging-profile
- | id-att-job-abort-criteria
- | id-att-job-warning-criteria )

JobInformationTypes ::= AttributeType

- ( id-att-job-name
- | id-att-job-originator
- | id-att-job-owner
- | id-att-job-comment
- | id-att-job-start-message
- | id-att-job-end-message )

JobResultsHandlingTypes ::= AttributeType

- ( id-att-results-profile )

```
JobSchedulingInstructionsTypes ::= AttributeType
(
  | id-at-job-message-to-operator
  | id-at-job-start-wait
  | id-at-job-end-wait
  | id-at-job-hold
  | id-at-job-password
  | id-at-job-print-after
  | id-at-job-priority
  | id-at-job-deadline-time
  | id-at-job-retention-time
  | id-at-job-retention-period )

PrinterIdentificationTypes ::= AttributeType
(
  | id-at-printer-identifier
  | id-at-printer-location
  | id-at-printer-model
  | id-at-printer-type
  | id-at-printer-security
  | id-at-printer-speed
  | id-at-printer-resolution )

ProductionInstructionsTypes ::= AttributeType
(
  | id-at-font-substitution
  | id-at-media-substitution
  | id-at-named-colour-substitution
  | id-at-finishing-specification
  | id-at-output-specification
  | id-at-output-bin
  | id-at-imposition-specification
  | id-at-print-quality
  | id-at-font-fidelity
  | id-at-media-fidelity
  | id-at-colour-fidelity
  | id-at-sides
  | id-at-page-select
  | id-at-page-media-select
  | id-at-copy-count
  | id-at-start-sheet
  | id-at-separator-sheet
  | id-at-end-sheet
  | id-at-reset-printer)

AccessAndAccountingTypes ::= AttributeType
(
  | id-at-user-name
  | id-at-accounting-information )
```



```
JobSecurityTypes ::= AttributeType
    ( id-att-job-security-handling )

PrintResult ::= SEQUENCE {
    job-identification [0] JobIdentification,
    returned-information [1] SET OF ReturnedInformation OPTIONAL,
    COMPONENTS OF CommonResults }

JobIdentification ::= Attribute
    ( WITH COMPONENTS {
        type ( id-att-job-identifier ),
        values } )

ReturnedInformation ::= Attribute
    ( WITH COMPONENTS {
        type ( id-att-server-state
            | id-att-server-message
            | id-att-file-reference
            | INCLUDES JobStatusTypes),
        values } )

JobStatusTypes ::= AttributeType
    ( id-att-current-job-state
    | id-att-previous-job-state
    | id-att-printers-assigned
    | id-att-estimated-completion-time
    | id-att-submission-time
    | id-att-modification-time
    | id-att-started-printing-time
    | id-att-copies-completed
    | id-att-pages-completed
    | id-att-octets-completed
    | id-att-intervening-pages
    | id-att-intervening-octets
    | id-att-intervening-jobs
    | id-att-print-checkpoint
    | id-att-now-server-name
    | id-att-on-request-resources-required
    | id-att-job-message-from-administrator
    | id-att-last-job-events
    | id-att-id-of-last-accessor
    | id-att-interrupt-at
    | id-att-pause-at )
```

-- Modify-job --

ModifyJob ::= ABSTRACT-OPERATION

ARGUMENT      ModifyJobArgument  
RESULT        ModifyJobResult  
ERRORS        { AccessError,  
               AttributeError,  
               DocumentAccessError,  
               PrinterError,  
               SecurityError,  
               SelectorError,  
               ServiceError,  
               UpdateError }

ModifyJobArgument ::= SEQUENCE {

  job-identification      [0] JobIdentification,  
  document-identification [1] DocumentIdentification OPTIONAL,  
  -- required for addressing individual documents in a multiple document print job --  
  job-attr-modification    [2] SET OF JobAttrModification OPTIONAL,  
  COMPONENTS OF CommonArguments }

DocumentIdentification ::= Attribute

  (WITH COMPONENTS {  
    type { id-at-document-sequence-number },  
    values                    })

JobAttrModification ::= QualifiedAttribute

  (WITH COMPONENTS {  
    attribute  
    (WITH COMPONENTS {  
      type    (    id-at-job-identifier  
                  | id-at-document-sequence-number  
                  | INCLUDES JobInformationTypes  
                  | INCLUDES JobResultsHandlingTypes  
                  | INCLUDES JobEventHandlingTypes  
                  | INCLUDES JobSchedulingInstructionsTypes  
                  | INCLUDES PrinterIdentificationTypes  
                  | INCLUDES ProductionInstructionsTypes ),  
      values                    ),  
    qualifier                    })

ModifyJobResult ::= SEQUENCE {

  returned-information [0] SET OF ReturnedInformation OPTIONAL,  
  COMPONENTS OF CommonResults }

-- CancelJob --

CancelJob ::= ABSTRACT-OPERATION

ARGUMENT CancelJobArgument

RESULT CancelJobResult

ERRORS { AccessError,  
NameError,  
SecurityError,  
SelectionError,  
ServiceError,  
UpdateError }

CancelJobArgument ::= SEQUENCE {

job-identification [0] JobIdentification,

document-identification [1] DocumentIdentification OPTIONAL,

-- required for addressing individual documents in a multiple document print job --

COMPONENTS OF CommonArguments }

CancelJobResult ::= SEQUENCE {

returned-information [0] SET OF ReturnedInformation OPTIONAL,

COMPONENTS OF CommonResults }

-- List-object-attributes --

ListObjectAttributes ::= ABSTRACT-OPERATION

ARGUMENT ListObjectAttributesArgument

RESULT ListObjectAttributesResult

ERRORS ( AccessError,  
AttributeError,  
NameError,  
SecurityError,  
ServiceError )

ListObjectAttributesArgument ::= SEQUENCE {

object-class [0] OBJECT IDENTIFIER,

continuation [1] ContinuationContext OPTIONAL,

selector [2] Selector OPTIONAL,

requested-attributes [3] SEQUENCE OF AttributeType OPTIONAL,

COMPONENTS OF CommonArguments }

ContinuationContext ::= OCTET STRING

-- implementation-specific information --

```

Selector ::= SET {
    object-identification [0] SET OF ObjectIdentification OPTIONAL
    -- There is only one member in this set, except for the case --
    -- when a particular document of a multiple document print job is selected --
    object-filter [1] Filter,
    time-limit [2] DeltaTime OPTIONAL,
    count-limit [3] INTEGER (1..ub-integer) OPTIONAL }

ObjectIdentification ::= Attribute
    ( WITH COMPONENTS {
        type ( id-at-job-identifier
            | id-at-document-sequence-number
            | id-at-printer-identifier
            | id-at-server-identifier
            | id-at-medium-identifier
            | id-at-font-identifier
            | id-at-named-colour-identifier
            | id-at-transfer-method-identifier
            | id-at-delivery-method-identifier
            | id-at-job-sheet-identifier
            | id-at-finishing-method-identifier
            | id-at-output-method-identifier
            | id-at-imposition-function-identifier
            | id-at-scheduler-identifier ),
        values ) )

DeltaTime ::= INTEGER -- time in seconds --

ListObjectAttributesResult ::= SEQUENCE {
    answer-time [1] UTCTime,
    continuation [2] ContinuationContext OPTIONAL,
    limit-encountered [3] LimitEncountered OPTIONAL,
    attributes [4] SEQUENCE OF ObjectResult }

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

ObjectResult ::= SEQUENCE {
    object-identification [0] SET OF ObjectIdentification
    -- There is only one member in this set, except for the case --
    -- when a particular document of a multiple document print job is selected --
    attributes [1] SET OF Attribute }

```

-- Promote-job --

PromoteJob ::= ABSTRACT-OPERATION

ARGUMENT PromoteJobArgument

RESULT PromoteJobResult

ERRORS { AccessError,  
AttributeError,  
NameError,  
SecurityError,  
SelectionError,  
ServiceError }

PromoteJobArgument ::= SEQUENCE {

job-identification [0] JobIdentification,

COMPONENTS OF CommonArguments }

PromoteJobResult ::= SEQUENCE {

returned-information [0] SET OF ReturnedInformation OPTIONAL,

COMPONENTS OF CommonResults }

-- Interrupt --

Interrupt ::= ABSTRACT-OPERATION

ARGUMENT InterruptArgument

RESULT InterruptResult

ERRORS { AccessError,  
AttributeError,  
NameError,  
SecurityError,  
SelectionError,  
ServiceError }

InterruptArgument ::= SEQUENCE {

interrupted-object-identification [0] Attribute

{ WITH COMPONENTS {

type ( id-att-job-identifier  
[ id-att-printer-identifier ],  
values ) },

interrupting-job-identification [1] Attribute

{ WITH COMPONENTS {

type ( id-att-job-identifier ),  
values ) }

COMPONENTS OF CommonArguments }

```
InterruptResult ::= SEQUENCE {
    interrupted-object-identification [0] Attribute
        ( WITH COMPONENTS {
            type ( id-att-job-identifier
                | id-att-printer-identifier ),
            values ) },
    returned-information [1] SET OF ReturnedInformation OPTIONAL,
    COMPONENTS OF CommonResults )
```

-- Pause --

```
Pause ::= ABSTRACT-OPERATION
    ARGUMENT PauseArgument
    RESULT PauseResult
    ERRORS ( AccessError,
        AttributeError,
        NameError,
        SecurityError,
        SelectorError,
        ServiceError )
```

```
PauseArgument ::= SEQUENCE {
    paused-object-identification [0] Attribute
        ( WITH COMPONENTS {
            type ( id-att-job-identifier
                | id-att-printer-identifier ),
            values ) },
    pause-message[1] Attribute
        ( WITH COMPONENTS {
            type ( id-att-job-message-from-administrator
                | id-att-printer-message ),
            values ) ) OPTIONAL,
    COMPONENTS OF CommonArguments )
```

```
PauseResult ::= SEQUENCE {
    paused-object-identification [0] Attribute
        ( WITH COMPONENTS {
            type ( id-att-job-identifier
                | id-att-printer-identifier ),
            values ) },
    returned-information [1] SET OF ReturnedInformation OPTIONAL,
    checkpoint [2] PrintCheckpoint,
    COMPONENTS OF CommonResults )
```

```
PrintCheckpoint ::= SEQUENCE {
    document-number [0] INTEGER (1 .. ub-integer) DEFAULT 1,
    page-number      [1] INTEGER (1 .. ub-integer) DEFAULT 1,
    copy-number      [2] INTEGER (1 .. ub-integer) DEFAULT 1,
    context-info     [3] OCTET STRING OPTIONAL -- implementation dependent --
}
```

-- Resume --

```
Resume ::= ABSTRACT-OPERATION
    ARGUMENT ResumeArgument
    RESULT ResumeResult
    ERRORS { AccessError,
             AttributeError,
             NameError,
             SecurityError,
             SelectionError,
             ServiceError }
```

```
ResumeArgument ::= SEQUENCE {
    resumed-object-identification [0] Attribute
        ( WITH COMPONENTS {
            type ( id-att-job-identifier
                  [ id-att-printer-identifier ] ),
            values } ),
    checkpoint [1] PrintCheckpoint,
    COMPONENTS OF CommonArguments }
```

```
ResumeResult ::= SEQUENCE {
    returned-information [1] SET OF ReturnedInformation OPTIONAL,
    COMPONENTS OF CommonResults }
```

-- ABSTRACT ERRORS --

-- Access-error --

```
AccessError ::= ABSTRACT-ERROR
    PARAMETER SEQUENCE OF SEQUENCE {
        job-identifier [0] JobIdentifier,
        problem [1] AccessProblem }
```

```
AccessProblem ::= ENUMERATED {
    inappropriate-object-class (1),
    insufficient-access-rights (2) }
```

- *Attribute-error* -

AttributeError ::= ABSTRACT-ERROR

PARAMETER SEQUENCE {  
  job-identifier [0] JobIdentifier OPTIONAL,  
  problems [1] SEQUENCE OF SEQUENCE {  
    problem [0] AttributeProblem,  
    type [1] AttributeType,  
    value [2] AttributeValue OPTIONAL } }

AttributeProblem ::= CHOICE {

  standard-problem ENUMERATED {  
    no-such-attribute (1),  
    invalid-attribute-syntax (2),  
    undefined-attribute-type (3),  
    inappropriate-matching (4),  
    constraint-violation (5),  
    attribute-or-value-already-exists (6),  
    illegal-modification (7),  
    inconsistent-with-other-attributes (8),  
    undefined-attribute-value (9),  
    unsupported-features (10) },  
  extended-problem OBJECT IDENTIFIER }

- *Document-access-error* -

DocumentAccessError ::= ABSTRACT-ERROR

PARAMETER SEQUENCE {  
  problem [0] DocumentAccessProblem }

DocumentAccessProblem ::= CHOICE {

  standard-problem ENUMERATED {  
    value-not-available (1),  
    referent-modified (2),  
    access-denied (3),  
    unknown-document (4) },  
  extended-problem OBJECT IDENTIFIER }

- *Name-error* -

NameError ::= ABSTRACT-ERROR

PARAMETER SEQUENCE OF SEQUENCE {  
  job-identifier [0] JobIdentifier,  
  problem [1] NameProblem }

NameProblem ::= CHOICE {

  standard-problem ENUMERATED {  
    invalid-job-identifier (1) },  
  extended-problem OBJECT IDENTIFIER }



-- Printer-error --

PrinterError ::= ABSTRACT-ERROR  
PARAMETER SEQUENCE {  
    problem [0] PrinterProblem }

PrinterProblem ::= CHOICE {  
    standard-problem ENUMERATED {  
        printer-error (1),  
        printer-attention (2),  
        printer-key-attention (3) },  
    extended-problem OBJECT IDENTIFIER }

-- Security-error --

SecurityError ::= ABSTRACT-ERROR  
PARAMETER SEQUENCE {  
    problem [0] SecurityProblem }

SecurityProblem ::= CHOICE {  
    standard-problem ENUMERATED {  
        inappropriate-authentication (1),  
        invalid-credentials (2),  
        insufficient-access-rights (3),  
        invalid-pac (4) },  
    extended-problem OBJECT IDENTIFIER }

-- Selection-error --

SelectionError ::= ABSTRACT-ERROR  
PARAMETER SEQUENCE {  
    problem [0] SelectionProblem }

SelectionProblem ::= CHOICE {  
    standard-problem ENUMERATED {  
        wrong-job-identifier (1),  
        wrong-printer-identifier (2),  
        wrong-document-sequence-number (3) },  
    extended-problem OBJECT IDENTIFIER }

-- Service-error --

ServiceError ::= ABSTRACT-ERROR  
PARAMETER SEQUENCE {  
    problem [0] ServiceProblem }

```
ServiceProblem ::= CHOICE {
    standard-problem    ENUMERATED {
        server-busy      (1),
        server-unavailable (2),
        operation-too-complex (3),
        resource-limit-exceeded (4),
        unclassified-server-error (5) },
    extended-problem    OBJECT IDENTIFIER }

-- Update-error
UpdateError ::= ABSTRACT-ERROR
    PARAMETER SEQUENCE {
        problem [0] UpdateProblem }

UpdateProblem ::= CHOICE {
    standard-problem    ENUMERATED {
        no-modifications-allowed (1),
        insufficient-access-rights (2),
        quality-of-service-violation (3),
        illegal-content-modification (4),
        previous-operation-incomplete (5),
        cancellation-not-possible (6) },
    extended-problem    OBJECT IDENTIFIER }

-- EVENTS --
EventType ::= OBJECT IDENTIFIER

EventIdentifier ::= OBJECT IDENTIFIER

END -- of DPA-AbstractService --
```

## APPENDIX C

### FORMAL DEFINITION OF DPA STANDARD ATTRIBUTES

This Appendix, a supplement to Section 3, formally defines the standard attribute-types applicable to DPA. It employs the ASN.1 and the ATTRIBUTE macro.

---

```
DPAStandardAttributes { iso-identified-organization(3) idc-ecma(0012) standard(3) dpa(140)
modules(0) attributes(2) }
```

```
DEFINITIONS IMPLICIT TAGS ::=
```

```
BEGIN
```

```
-- PROLOGUE --
```

```
EXPORTS -- everything -- ;
```

```
IMPORTS
```

```
-- OSI Directory --
```

```
ATTRIBUTE, ATTRIBUTE-SYNTAX, DistinguishedName
FROM InformationFramework
[ joint-iso-ccitt ds(5) modules(1) informationFramework(1) ]
```

```
-- MOTIS --
```

```
ORAAAddressAndOrDirectoryName
FROM MTSAbstractService
[ joint-iso-ccitt mhs-motis(6) mts(3) modules(0) mts-abstract-service(1) ]
```

```
-- RDT --
```

```
RDT-Reference
FROM RDT-Reference-definition
[ iso-standard doam(10031) part-2(2) reference-definition(0) ]
```

```
-- DPA Object Identifiers --
```

```
id-syn-identifier, id-syn-descriptor, id-syn-name, id-syn-aliases, id-syn-state,
id-syn-message, id-syn-availability, id-syn-list-of-managers, id-syn-text, id-att-job-identifier,
id-att-job-name, id-att-job-originator, id-att-job-owner, id-att-job-comment,
id-att-job-start-message, id-att-job-end-message, id-att-results-profile,
id-att-notification-profile, id-att-logging-profile, id-att-job-abort-criteria,
id-att-job-warning-criteria, id-att-job-start-wait, id-att-job-end-wait, id-att-job-hold,
id-att-job-password, id-att-job-print-after, id-att-job-priority, id-att-job-deadline-time,
id-att-job-retention-time, id-att-job-retention-period, id-att-job-message-to-operator,
id-att-font-substitution, id-att-media-substitution, id-att-named-colour-substitution,
id-att-freshing-specification, id-att-output-specification, id-att-output-bin,
id-att-imposition-specification, id-att-print-quality, id-att-font-fidelity, id-att-media-fidelity,
id-att-colour-fidelity, id-att-sides, id-att-page-select, id-att-page-media-select,
id-att-copy-count, id-att-start-sheet, id-att-separator-sheet, id-att-end-sheet,
```

id-att-reset-printer, id-att-transfer-method, id-att-document-format,  
id-att-document-sequence-number, id-att-document-content, id-att-file-reference,  
id-att-document-descriptor, id-att-document-authors, id-att-document-message,  
id-att-document-revision-date, id-att-content-formats-used,  
id-att-assured-reproduction-area, id-att-page-size, id-att-content-orientation,  
id-att-page-order, id-att-page-independent, id-att-character-sets-used,  
id-att-character-mappings-used, id-att-fonts-used, id-att-character-repertoires-used,  
id-att-media-used, id-att-named-colours-used, id-att-octet-count, id-att-page-count,  
id-att-glyph-count, id-att-font-count, id-att-font-change-count,  
id-att-maximum-fonts-per-page, id-att-percent-graphics, id-att-percent-images,  
id-att-user-name, id-att-accounting-information, id-att-current-job-state,  
id-att-previous-job-state, id-att-printers-assigned, id-att-estimated-completion-time,  
id-att-submission-time, id-att-modification-time, id-att-started-printing-time,  
id-att-copies-completed, id-att-pages-completed, id-att-octets-completed,  
id-att-intervening-pages, id-att-intervening-octets, id-att-intervening-jobs,  
id-att-print-checkpoint, id-att-on-request-resources-required,  
id-att-job-message-from-administrator, id-att-last-job-events, id-att-id-of-last-accessor,  
id-att-interrupt-at, id-att-pause-at, id-att-error-count, id-att-warning-count,  
id-att-processing-time, id-att-job-security-handling, id-att-printer-identifier,  
id-att-printer-name, id-att-printer-location, id-att-printer-model, id-att-printer-type,  
id-att-printer-security, id-att-printer-speed, id-att-printer-resolution, id-att-printer-state,  
id-att-printer-message, id-att-attributes-supported-by-printer,  
id-att-object-classes-supported-by-printer, id-att-printers-supported, id-att-printers-active,  
id-att-fonts-supported, id-att-fonts-active, id-att-font-default, id-att-media-supported,  
id-att-media-active, id-att-medium-default, id-att-named-colours-supported,  
id-att-named-colours-active, id-att-named-colour-default, id-att-finishing-methods-supported,  
id-att-finishing-methods-active, id-att-finishing-methods-default,  
id-att-output-methods-supported, id-att-output-methods-active,  
id-att-output-methods-default, id-att-output-bins-supported, id-att-output-bins-active,  
id-att-output-bins-default, id-att-imposition-functions-supported,  
id-att-imposition-functions-enabled, id-att-imposition-functions-default,  
id-att-print-qualities-supported, id-att-print-qualities-active, id-att-print-quality-default,  
id-att-job-sheets-supported, id-att-job-sheets-active, id-att-start-sheet-default,  
id-att-separator-sheet-default, id-att-end-sheet-default, id-att-character-sets-supported,  
id-att-character-sets-active, id-att-character-set-default,  
id-att-character-mappings-supported, id-att-character-mappings-active,  
id-att-character-mapping-default, id-att-character-repertoires-supported,  
id-att-character-repertoires-active, id-att-character-repertoire-default,  
id-att-start-message-supported, id-att-end-message-supported, id-att-sides-supported,  
id-att-sides-default, id-att-page-select-supported, id-att-maximum-copies-supported,  
id-att-content-formats-supported, id-att-content-format-default,  
id-att-document-formats-supported, id-att-document-format-default, id-att-ares-supported,  
id-att-ara-default, id-att-page-sizes-supported, id-att-orientations-supported,  
id-att-orientation-default, id-att-page-order-assumed, id-att-page-independence-assumed,

id-att-associated-server, id-att-sheet-count, id-att-printer-needs-attention-time,  
id-att-printer-needs-key-operator-attention-time, id-att-server-identifier, id-att-server-name,  
id-att-server-address, id-att-server-state, id-att-server-message,  
id-att-object-classes-supported-by-server, id-att-attributes-supported-by-server,  
id-att-physical-printers-supported, id-att-physical-printers-active,  
id-att-logical-printers-supported, id-att-logical-printers-active, id-att-translators-supported,  
id-att-translators-active, id-att-delivery-methods-supported, id-att-delivery-methods-active,  
id-att-scheduling-algorithms-supported, id-att-scheduling-algorithm-active,  
id-att-system-hints-supported, id-att-medium-identifier, id-att-medium-state,  
id-att-medium-message, id-att-medium-type, id-att-medium-size, id-att-medium-dimensions,  
id-att-medium-colour, id-att-medium-looth, id-att-medium-grain, id-att-medium-sides,  
id-att-medium-weight, id-att-medium-assured-reproduction-area,  
id-att-medium-associated-media, id-att-medium-realisation, id-att-medium-form-type,  
id-att-medium-form-parts, id-att-font-identifier, id-att-font-type, id-att-font-state,  
id-att-font-message, id-att-named-colour-identifier, id-att-named-colour-type,  
id-att-named-colour-state, id-att-named-colour-message, id-att-transfer-method-identifier,  
id-att-document-reference-type, id-att-transfer-method-message,  
id-att-delivery-method-identifier, id-att-delivery-method-message, id-att-job-sheet-identifier,  
id-att-job-sheet-message, id-att-finishing-method-identifier, id-att-finishing-method-message,  
id-att-output-method-identifier, id-att-output-method-message,  
id-att-imposition-function-identifier, id-att-imposition-function-message,  
id-att-scheduler-identifier, id-att-scheduler-message

FROM DPASandardObjectIdentifiers

{ iso-identified-organization(3) idc-ecma(0012) standard(0) dpa(140) modules(0) object-  
identifiers(0) }

- Data types from DPA abstract service -

JobIdentifier

FROM DPAAbstractService

{ iso-identified-organization(3) idc-ecma(0012) standard(0) dpa(140) modules(0)  
abstract-service(1) };

- DPA upper bound -

ub-descriptor-string, ub-integer, ub-message-string, ub-text-string

FROM DPAUpperbound

{ iso-identified-organization(3) idc-ecma(0012) standard(0) dpa(140) modules(0)  
upper-bound(2) };

- Attribute Syntaxes -

IdentifierSyntax ATTRIBUTE-SYNTAX

OBJECT IDENTIFIER

::= id-syn-identifier

descriptorSyntax ATTRIBUTE-SYNTAX

IA5String (1..ub-descriptor-string)

::= id-syn-descriptor

nameSyntax ATTRIBUTE-SYNTAX

IA5String (1..ub-name-string)

::= id-syn-name

aliasesSyntax ATTRIBUTE-SYNTAX

OBJECT IDENTIFIER

::= id-syn-aliases

stateSyntax ATTRIBUTE-SYNTAX

OBJECT IDENTIFIER

::= id-syn-state

messageSyntax ATTRIBUTE-SYNTAX

IA5String (1..ub-message-string)

::= id-syn-message

availabilitySyntax ATTRIBUTE-SYNTAX

OBJECT IDENTIFIER

::= id-syn-availability

listOfManagersSyntax ATTRIBUTE-SYNTAX

IA5String (1..ub-name-string)

::= id-syn-list-of-managers

textSyntax ATTRIBUTE-SYNTAX

IA5String (1..ub-text-string)

::= id-syn-text

-- Attributes --

job-identifier ATTRIBUTE

WITH ATTRIBUTE-SYNTAX jobIdentifier

SINGLE VALUE

::= id-at-job-identifier

job-name ATTRIBUTE

WITH ATTRIBUTE-SYNTAX nameSyntax

SINGLE VALUE

::= id-at-job-name

job-originator ATTRIBUTE

WITH ATTRIBUTE-SYNTAX textSyntax

SINGLE VALUE

::= id-at-job-originator

job-owner ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX textSyntax  
SINGLE VALUE  
::= id-att-job-owner

job-comment ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-job-comment

job-start-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-job-start-message

job-end-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-job-end-message

ResultsProfile ::= SEQUENCE {  
  delivery-method OBJECT IDENTIFIER,  
  delivery-descriptor IASString (1..ub-text-string),  
  delivery-address DeliveryAddress}

results-profile ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX ResultsProfile  
SINGLE VALUE  
::= id-att-results-profile

DeliveryAddress ::= CHOICE {  
  ORAddressAndOrDirectoryName,  
  DistinguishedName,  
  IASString (1..ub-text-string)}

NotificationProfile ::= SEQUENCE {  
  event-identifiers SET OF EventIdentifier,  
  delivery-method OBJECT IDENTIFIER,  
  delivery-descriptor IASString (1..ub-text-string),  
  delivery-address DeliveryAddress}

notification-profile ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX NotificationProfile  
SINGLE VALUE  
::= id-att-notification-profile

LoggingProfile ::= SEQUENCE {  
    event-identifiers SET OF EventIdentifier,  
    delivery-method OBJECT IDENTIFIER,  
    delivery-descriptor IA5String (1..ub-text-string),  
    delivery-address DeliveryAddress}

logging-profile ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX LoggingProfile  
    SINGLE VALUE  
    ::= id-att-logging-profile

Criteria ::= SEQUENCE {  
    counter-identifier AttributeType,  
    counter-threshold Threshold}

Threshold ::= CHOICE (INTEGER (1..ub-integer), UTCTime)

job-abort-criteria ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX Criteria  
    MULTI VALUE  
    ::= id-att-job-abort-criteria

job-warning-criteria ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX Criteria  
    MULTI VALUE  
    ::= id-att-job-warning-criteria

job-start-wait ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX Boolean  
    SINGLE VALUE  
    ::= id-att-job-start-wait

job-end-wait ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX Boolean  
    SINGLE VALUE  
    ::= id-att-job-end-wait

job-hold ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX Boolean  
    SINGLE VALUE  
    ::= id-att-job-hold

job-password ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX OCTET STRING  
    SINGLE VALUE  
    ::= id-att-job-password



- job-print-after ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX UTCTime  
SINGLE VALUE  
::= id-att-job-print-after
- job-priority ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..100)  
SINGLE VALUE  
::= id-att-job-priority
- job-deadline-time ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX UTCTime  
SINGLE VALUE  
::= id-att-job-deadline-time
- job-retention-time ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX UTCTime  
SINGLE VALUE  
::= id-att-job-retention-time
- job-retention-period ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX DeltaTime  
SINGLE VALUE  
::= id-att-job-retention-period
- job-message-to-operator ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-job-message-to-operator
- FontSubstitution ::= SEQUENCE {  
original-font FontReference,  
substitution-font FontReference }
- font-substitution ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX FontSubstitution  
MULTI VALUE  
::= id-att-font-substitution
- MediaSubstitution ::= SEQUENCE {  
original-medium MediumIdentifier,  
substitution-mediumMediumIdentifier }
- media-substitution ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX MediaSubstitution  
MULTI VALUE  
::= id-att-media-substitution

NamedColourSubstitution ::= SEQUENCE (  
    original-colour       OBJECT IDENTIFIER,  
    substitution-colour   OBJECT IDENTIFIER )

named-colour-substitution ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX NamedColourSubstitution  
    MULTI VALUE  
    ::= id-att-named-colour-substitution

finishing-specification ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
    MULTI VALUE  
    ::= id-att-finishing-specification

output-specification ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
    MULTI VALUE  
    ::= id-att-output-specification

OutputBin ::= CHOICE (  
    bin-number   [0]INTEGER (1..ub-integer),  
    bin-name     [1]OBJECT IDENTIFIER )

output-bin ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX OutputBin  
    SINGLE VALUE  
    ::= id-att-output-bin

imposition-specification ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
    MULTI VALUE  
    ::= id-att-imposition-specification

print-quality ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
    SINGLE VALUE  
    ::= id-att-print-quality

font-fidelity ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
    SINGLE VALUE  
    ::= id-att-font-fidelity

media-fidelity ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
    SINGLE VALUE  
    ::= id-att-media-fidelity

colour-fidelity ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-att-colour-fidelity

sides ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (1..2)

SINGLE VALUE

::= id-att-sides

PageIdentifier ::= CHOICE {

nominal-page-number INTEGER,

alphanumeric-page-number IAString (1..ub-name-string) }

PageSelect ::= SEQUENCE {

beginning-page [0]PageIdentifier OPTIONAL, -- omitted means no lower bound --

ending-page [1]PageIdentifier OPTIONAL -- omitted means no upper bound -- }

PageIdentifier ::= CHOICE {

nominal-page-number INTEGER,

alphanumeric-page-number IAString (1..ub-name-string) }

page-select ATTRIBUTE

WITH ATTRIBUTE-SYNTAX PageSelect

MULTI VALUE

::= id-att-page-select

PageMediaSelect ::= SEQUENCE {

page-range [0]PageSelect,

medium-id [1]MediaIdentifier }

page-media-select ATTRIBUTE

WITH ATTRIBUTE-SYNTAX PageMediaSelect

MULTI VALUE

::= id-att-page-media-select

copy-count ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)

DEFAULT 1

SINGLE VALUE

::= id-att-copy-count

start-sheet ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

SINGLE VALUE

::= id-att-start-sheet

- separator-sheet ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-separator-sheet
- end-sheet ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-end-sheet
- reset-printer ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX BOOLEAN  
DEFAULT TRUE  
SINGLE VALUE  
::= id-att-reset-printer
- transfer-method ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-transfer-method
- document-format ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-document-format
- SequenceNumber ::= SEQUENCE {  
sequence-number-in-set INTEGER (1..ub-integer) DEFAULT 1,  
total-documents-in-set INTEGER (1..ub-integer) DEFAULT 1 }
- document-sequence-number ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX SequenceNumber  
SINGLE VALUE  
::= id-att-document-sequence-number
- DocumentContent ::= CHOICE {  
included-document EXTERNAL,  
referenced-document RDT-reference}
- document-content ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX DocumentContent  
SINGLE VALUE  
::= id-att-document-content
- file-reference ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OCTET STRING  
SINGLE VALUE  
::= id-att-file-reference

document-descriptor ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX descriptorSyntax  
SINGLE VALUE  
::= id-att-document-descriptor

document-authors ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX textSyntax  
MULTI VALUE  
::= id-att-document-authors

document-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-document-message

document-revision-date ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX UTCTime  
SINGLE VALUE  
::= id-att-document-revision-date

content-formats-used ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-content-formats-used

AssuredReproductionArea ::= SEQUENCE  
(INTEGER (1..ub-integer),  
INTEGER (1..ub-integer))

assured-reproduction-area ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX AssuredReproductionArea  
SINGLE VALUE  
::= id-att-assured-reproduction-area

page-size ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-page-size

content-orientation ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-content-orientation

page-order ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
DEFAULT id-val-page-order-forward  
SINGLE VALUE  
::= id-att-page-order

- page-independent ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX BOOLEAN
  - DEFAULT TRUE
  - SINGLE VALUE
  - ::= id-att-page-independent
- character-sets-used ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-character-sets-used
- character-mappings-used ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-character-mappings-used
- fonts-used ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX FontReference
  - MULTI VALUE
  - ::= id-att-fonts-used
- character-repertoires-used ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-character-repertoires-used
- media-used ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-media-used
- named-colours-used ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-named-colours-used
- octet-count ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)
  - SINGLE VALUE
  - ::= id-att-octet-count
- page-count ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)
  - SINGLE VALUE
  - ::= id-att-page-count

glyph-count ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
::= id-at-glyph-count

font-count ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
::= id-at-font-count

font-change-count ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
::= id-at-font-change-count

maximum-fonts-per-page ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
::= id-at-maximum-fonts-per-page

percent-graphics ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..100)  
SINGLE VALUE  
::= id-at-percent-graphics

percent-images ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..100)  
SINGLE VALUE  
::= id-at-percent-images

user-name ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX nameSyntax  
SINGLE VALUE  
::= id-at-user-name

accounting-information ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OCTET STRING  
SINGLE VALUE  
::= id-at-accounting-information

current-job-state ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-current-job-state

previous-job-state ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-previous-job-state

PrinterAssigned ::= SEQUENCE {  
    printer-identifier [0] OBJECT IDENTIFIER,  
    printer-state [1] OBJECT IDENTIFIER}

printers-assigned ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX PrinterAssigned  
    MULTI VALUE  
    ::= id-att-printers-assigned

estimated-completion-time ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX UTCTime  
    SINGLE VALUE  
    ::= id-att-estimated-completion-time

submission-time ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX UTCTime  
    SINGLE VALUE  
    ::= id-att-submission-time

modification-time ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX UTCTime  
    SINGLE VALUE  
    ::= id-att-modification-time

started-printing-time ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX UTCTime  
    SINGLE VALUE  
    ::= id-att-started-printing-time

copies-completed ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)  
    SINGLE VALUE  
    ::= id-att-copies-completed

pages-completed ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)  
    SINGLE VALUE  
    ::= id-att-pages-completed

octets-completed ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)  
    SINGLE VALUE  
    ::= id-att-octets-completed

intervening-pages ATTRIBUTE  
    WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)  
    SINGLE VALUE  
    ::= id-att-intervening-pages



intervening-octets ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)  
SINGLE VALUE  
::= id-att-intervening-octets

intervening-jobs ATTRIBUTE

WITH ATTRIBUTE-SYNTAX INTEGER (0..ub-integer)  
SINGLE VALUE  
::= id-att-intervening-jobs

print-checkpoint ATTRIBUTE

WITH ATTRIBUTE-SYNTAX SEQUENCE OF PrintCheckpoint  
SINGLE VALUE  
::= id-att-print-checkpoint

on-request-resources-required ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-on-request-resources-required

job-message-from-administrator ATTRIBUTE

WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-job-message-from-administrator

last-job-events ATTRIBUTE

WITH ATTRIBUTE-SYNTAX EventType  
MULTI VALUE  
::= id-att-last-job-events

id-of-last-accessor ATTRIBUTE

WITH ATTRIBUTE-SYNTAX nameSyntax  
SINGLE VALUE  
::= id-att-id-of-last-accessor

At ::= ENUMERATED {  
    immediately (0),  
    end-of-page (1),  
    end-of-copy (2),  
    end-of-document (3),  
    end-of-job (4),  
    next-pausable-point (5)}

interrupt-at ATTRIBUTE

WITH ATTRIBUTE-SYNTAX At  
SINGLE VALUE  
::= id-att-interrupt-at

pause-at ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX At  
SINGLE VALUE  
::= id-att-pause-at

error-count ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER  
SINGLE VALUE  
::= id-att-error-count

warning-count ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER  
SINGLE VALUE  
::= id-att-warning-count

processing-time ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX DeltaTime  
SINGLE VALUE  
::= id-att-processing-time

JobSecurityHandling ::= SET (  
copy-numbering [0] OBJECT IDENTIFIER OPTIONAL,  
page-numbering [1] OBJECT IDENTIFIER OPTIONAL,  
sheet-numbering [2] OBJECT IDENTIFIER OPTIONAL,  
security-labelling [3] OBJECT IDENTIFIER OPTIONAL )

job-security-handling ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX JobSecurityHandling  
SINGLE VALUE  
::= id-att-job-security-handling

printer-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-printer-identifier

printer-name ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX nameSyntax  
SINGLE VALUE  
::= id-att-printer-name

printer-location ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-printer-location

printer-model ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-printer-model

printer-type ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-printer-type

printer-security ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX SecurityClassification  
SINGLE VALUE  
::= id-att-printer-security

SecurityClassification ::= OBJECT IDENTIFIER

printer-speed ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
::= id-att-printer-speed

printer-resolution ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
::= id-att-printer-resolution

printer-state ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-printer-state

printer-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-printer-message

attributes-supported-by-printer ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX AttributeType  
MULTI VALUE  
::= id-att-attributes-supported-by-printer

object-classes-supported-by-printer ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-object-classes-supported-by-printer

printers-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-printers-supported

printers-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-printers-active

fonts-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX FontReference  
MULTI VALUE  
::= id-att-fonts-supported

fonts-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX FontReference  
MULTI VALUE  
::= id-att-fonts-active

font-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX FontReference  
SINGLE VALUE  
::= id-att-font-default

media-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX MediaIdentifier  
MULTI VALUE  
::= id-att-media-supported

media-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX MediaIdentifier  
MULTI VALUE  
::= id-att-media-active

medium-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX MediaIdentifier  
SINGLE VALUE  
::= id-att-medium-default

named-colours-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-named-colours-supported

named-colours-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-named-colours-active

named-colour-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-named-colour-default

finishing-methods-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-finishing-methods-supported

finishing-methods-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-finishing-methods-active

finishing-methods-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-finishing-methods-default

output-methods-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-output-methods-supported

output-methods-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-output-methods-active

output-methods-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-output-methods-default

output-bins-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OutputBin  
MULTI VALUE  
::= id-att-output-bins-supported

output-bins-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OutputBin  
MULTI VALUE  
::= id-att-output-bins-active

output-bins-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OutputBin  
MULTI VALUE  
::= id-att-output-bins-default

imposition-functions-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE

□ = id-at-imposition-functions-supported

imposition-functions-enabled ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE

□ = id-at-imposition-functions-enabled

imposition-functions-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE

□ = id-at-imposition-functions-default

print-qualities-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE

□ = id-at-print-qualities-supported

print-qualities-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE

□ = id-at-print-qualities-active

print-quality-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE

□ = id-at-print-quality-default

job-sheets-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE

□ = id-at-job-sheets-supported

job-sheets-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE

□ = id-at-job-sheets-active

start-sheet-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE

□ = id-at-start-sheet-default

separator-sheet-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE

□ = id-at-separator-sheet-default

- end-sheet-default ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - SINGLE VALUE
  - ::= id-att-end-sheet-default
- character-sets-supported ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-character-sets-supported
- character-sets-active ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-character-sets-active
- character-set-default ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - SINGLE VALUE
  - ::= id-att-character-set-default
- character-mappings-supported ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-character-mappings-supported
- character-mappings-active ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-character-mappings-active
- character-mapping-default ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - SINGLE VALUE
  - ::= id-att-character-mapping-default
- character-repertoires-supported ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-character-repertoires-supported
- character-repertoires-active ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-character-repertoires-active
- character-repertoire-default ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - SINGLE VALUE
  - ::= id-att-character-repertoire-default

start-message-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX BOOLEAN  
SINGLE VALUE  
::= id-att-start-message-supported

end-message-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX BOOLEAN  
SINGLE VALUE  
::= id-att-end-message-supported

SidesSupported ::= ENUMERATED {  
both (0), -- Both 1-sided and 2-sided printing is supported --  
one (1), -- Only 1-sided printing is supported --  
two (2) } -- Only 2-sided printing is supported --

sides-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX SidesSupported  
SINGLE VALUE  
::= id-att-sides-supported

sides-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX SidesSupported  
SINGLE VALUE  
::= id-att-sides-default

page-select-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX PageIdType  
MULTI VALUE  
::= id-att-page-select-supported

PageIdType ::= ENUMERATED {  
numeric (1),  
alphanumeric (2) }

maximum-copies-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
::= id-att-maximum-copies-supported

content-formats-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-content-formats-supported

content-format-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-content-format-default



- document-formats-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-document-formats-supported
- document-format-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-document-format-default
- aras-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX AssuredReproductionArea  
MULTI VALUE  
::= id-att-aras-supported
- ara-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX AssuredReproductionArea  
SINGLE VALUE  
::= id-att-ara-default
- page-sizes-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-page-sizes-supported
- orientations-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-orientations-supported
- orientation-default ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-orientation-default
- page-order-assumed ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX BOOLEAN  
SINGLE VALUE  
::= id-att-page-order-assumed
- page-independence-assumed ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX BOOLEAN  
SINGLE VALUE  
::= id-att-page-independence-assumed
- associated-server ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-associated-server

- sheet-count ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX INTEGER
  - SINGLE VALUE
  - ::= id-att-sheet-count
  
- printer-needs-attention-time ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX DeltaTime
  - SINGLE VALUE
  - ::= id-att-printer-needs-attention-time
  
- printer-needs-key-operator-attention-time ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX DeltaTime
  - SINGLE VALUE
  - ::= id-att-printer-needs-key-operator-attention-time
  
- server-identifier ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - SINGLE VALUE
  - ::= id-att-server-identifier
  
- server-name ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX nameSyntax
  - SINGLE VALUE
  - ::= id-att-server-name
  
- server-address ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OCTET STRING
  - SINGLE VALUE
  - ::= id-att-server-address
  
- server-state ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX stateSyntax
  - SINGLE VALUE
  - ::= id-att-server-state
  
- server-message ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX messageSyntax
  - SINGLE VALUE
  - ::= id-att-server-message
  
- object-classes-supported-by-server ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  - MULTI VALUE
  - ::= id-att-object-classes-supported-by-server
  
- attributes-supported-by-server ATTRIBUTE
  - WITH ATTRIBUTE-SYNTAX AttributeType
  - MULTI VALUE
  - ::= id-att-attributes-supported-by-server

physical-printers-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-physical-printers-supported

physical-printers-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-physical-printers-active

logical-printers-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-logical-printers-supported

logical-printers-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-logical-printers-active

translators-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-translators-supported

translators-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-translators-active

delivery-methods-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-delivery-methods-supported

delivery-methods-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-delivery-methods-active

scheduling-algorithms-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-scheduling-algorithms-supported

scheduling-algorithm-active ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-scheduling-algorithm-active

system-hints-supported ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-system-hints-supported

medium-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-medium-identifier

medium-state ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX stateSyntax  
SINGLE VALUE  
::= id-att-medium-state

medium-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-medium-message

medium-type ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-medium-type

medium-size ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-medium-size

medium-dimensions ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX SEQUENCE  
{ INTEGER(1..ub-integer),  
INTEGER(1..ub-integer)}  
SINGLE VALUE  
::= id-att-medium-dimensions

medium-colour ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX nameSyntax  
SINGLE VALUE  
::= id-att-medium-colour

medium-tooth ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX nameSyntax  
SINGLE VALUE  
::= id-att-medium-tooth

medium-grain ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX nameSyntax  
SINGLE VALUE  
::= id-att-medium-grain

medium-sides ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..2)  
SINGLE VALUE  
::= id-att-medium-sides

medium-weight ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
::= id-att-medium-weight

medium-assured-reproduction-area ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX AssuredReproductionArea  
SINGLE VALUE  
::= id-att-medium-assured-reproduction-area

medium-associated-media ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
MULTI VALUE  
::= id-att-medium-associated-media

MediumRealisation ::= ENUMERATED (  
logical (1),  
physical (2),  
logical-and-physical(3) )

medium-realisation ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX MediumRealisation  
SINGLE VALUE  
::= id-att-medium-realisation

medium-form-type ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-medium-form-type

medium-form-parts ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX INTEGER (1..ub-integer)  
SINGLE VALUE  
::= id-att-medium-form-parts

font-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-font-identifier

font-type ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-font-type

font-state ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX stateSyntax  
SINGLE VALUE  
::= id-att-font-state

font-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-font-message

named-colour-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-named-colour-identifier

named-colour-type ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-named-colour-type

named-colour-state ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX stateSyntax  
SINGLE VALUE  
::= id-att-named-colour-state

named-colour-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-named-colour-message

transfer-method-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-transfer-method-identifier

document-reference-type ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-att-document-reference-type

transfer-method-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-att-transfer-method-message

delivery-method-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-delivery-method-identifier

delivery-method-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-at-delivery-method-message

job-sheet-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-job-sheet-identifier

job-sheet-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-at-job-sheet-message

finishing-method-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-finishing-method-identifier

finishing-method-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-at-finishing-method-message

output-method-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-output-method-identifier

output-method-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-at-output-method-message

imposition-function-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-at-imposition-function-identifier

imposition-function-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-at-imposition-function-message

scheduler-identifier ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER  
SINGLE VALUE  
::= id-ab-scheduler-identifier

scheduler-message ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX messageSyntax  
SINGLE VALUE  
::= id-ab-scheduler-message

END -- of DPASstandardAttributes --



## APPENDIX D

### FORMAL DEFINITION OF DPA STANDARD PARAMETER UPPER BOUND

This Appendix, a supplement to Sections 2 and 3, formally defines the standard parameter upper bounds applicable to DPA. It employs ASN.1.

---

DPAUpperBounds { iso-identified-organization(3) idc-ecma(2012) standard(0) dpa(140) modules(0)  
upper-bounds(3) }

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- PROLOGUE --

EXPORTS -- everything -- :

IMPORTS -- nothing -- :

-- Upper Bounds --

ub-descriptor-string INTEGER ::= 256

ub-integer INTEGER ::= 2147483647 -- 2<sup>31</sup> --

ub-message-string INTEGER ::= 512

ub-name-string INTEGER ::= 32

ub-text-string INTEGER ::= 127

END -- of DPA-UpperBounds --



---

**Part 2**  
**Protocol specification**

---

1065

1065

SECTION ONE - INTRODUCTION



## 1. SCOPE AND FIELD OF APPLICATION

### 1.1 Scope

This Standard ECMA-140 consists of two parts:

Part 1: Document Printing Application - Abstract Service Definition and Procedures

Part 2: Document Printing Application - Protocol Specification

The Document Printing Application is one component of a coordinated set of facilities and standards needed to satisfy the printing requirements of the modern distributed office. Together, the capabilities provided can enable users to create and produce high-quality office documents in a consistent and unambiguous manner within a distributed open system environment.

Specifically, this Standard addresses those aspects of document processing that enable users in a distributed open system environment to send electronic documents to shared, possibly geographically-dispersed printers, and to cause the documents to be printed in accordance with their desires. For the purposes of this Standard, it is assumed that such documents have been composed in a form that is compatible with the destination printing system prior to their introduction to the Document Printing Application.

Other Standards deal with related aspects of document processing, such as the creation and formatting of electronic documents, and the underlying protocols used to transport electronic documents to a printing system. This Standard is aligned with these related Standards as appropriate, and shares some information in common with them. Clause 2 identifies those standards that are directly applicable to this one.

The Document Printing Application defined in this Standard is consistent with the model, architectural framework and design principles of the Distributed Office Applications Model (ISO/IEC 10031-1). This Document Printing Application Standard defines services and specifies access protocols available within the application layer of the Reference Model (ISO 7488).

### 1.2 Field of Application

The document printing application constitutes the final phase of the document processing cycle, i.e., the queuing, preparation, rendering and finishing of the fully composed form of the document on marking engines and other image generation devices. This cycle includes other processes such as document creation and interchange through public and private networks.

This Standard is oriented toward satisfying the following subset of the overall document processing functional requirements:

- an ability for multiple users to share access to distributed printers;
- an ability for users to convey information to a printing system to influence the scheduling and processing requirements of a print job;
- a capability for users to monitor and manage the progress of their print jobs;

- a capability for printing systems, and associated facilities, to protect against unauthorized printing of documents.

Many different document print formats have been developed, and are in wide use. For this reason, the Document Printing Application has been developed with a view toward supporting arbitrary print formats in a transparent manner. That is, the specific content or format of an electronic document is independent of the access protocol defined by the Document Printing Application Standard. The only requirement is that the destination printing system be capable of dealing with the format of the transmitted document, and possess the features and functionality needed to successfully render the document.

However, in spite of this generality of focus, this Standard is particularly oriented toward providing the features needed to assist in the transport and faithful rendering of documents formatted in the Standard Page Description Language (SPDL - ISO/IEC 10180).

The access protocol defined by this standard enables a user to convey document files to a document print server, along with the parameters needed to express the user's desires regarding the scheduling and production of the ensuing print job. In addition, the protocol permits a user to inquire about the status, capabilities and characteristics of a document print server in order to choose from a variety of printing devices, depending on capabilities, formats, logistic convenience, cost, ownership and availability.

The protocol also allows users to inquire about jobs, modify the characteristics and progress of jobs, and obtain feedback about a job.

## 2. REFERENCES

The references to corresponding standards are noted in Part I, as well as the definition of the terminology and the abbreviations used.



SECTION TWO - DPA ACCESS PROTOCOL SPECIFICATION



### 3. OVERVIEW OF THE PROTOCOL.

#### 3.1 DPA Access Protocol Model

Part I of this ECMA Standard describes an abstract model of the Document Printing Application, and the DPA Abstract Service which is provided to the DPA-User.

This clause describes how the DPA Abstract Service is supported by instances of OSI communication when an abstract-service user and an abstract-service provider are realized as application-processes located in different open systems.

In the OSI environment, communication between application-processes is represented in terms of communication between a pair of application-entities (AEs) using the presentation-service. The functionality of an application-entity is factored into a set of one or more application-service-elements (ASEs). The interaction between AEs is described in terms of their use of the services provided by the ASEs.

Access to the DPA Abstract Service is supported by two application-service-elements, each supporting a port paired between a DPA-User and the DPA-Server in the abstract model. The Document Printing Service Element (DPSE) supports the services of the printing-port; and the Document Printing Administration Service Element (DPASE) supports the services of the administration port. The DPSE and DPASE are asymmetric ASEs, that is, the DPA-User acts as the consumer, and the DPA-Server acts as the supplier, of the DPA Abstract Service.

The application-service-elements are in turn supported by other application-service-elements.

The Remote Operations Service Element (ROSE) supports the request/reply paradigm of the abstract operations that occur at the DPA-port in the abstract model. The DPSE and DPASE provide the mapping function of the abstract-syntax notation of this abstract-service onto the services provided by the ROSE.

The Association Control Service Element (ACSE) supports the establishment and release of an application-association between a pair of AEs. Associations between a DPA-User and the DPA-Server may be established only by the DPA-User, and only the initiator of an established association can release it.

The combination of one or more of the DPSE and DPASE, together with the supporting ASEs, defines the application-context of an application-association. Note that a single application-association may be used to support one or more port types paired between two objects in the abstract model.

Table 1 identifies the application-contexts defined in this Standard for the DPA access protocol. Definition of other application-contexts which include DPA service elements is not precluded but is outside the scope of this Standard.

application context	DPSE	DPASE	ROSE	RTSE	ACSE
dpa-access	C	-	X	-	X
dpa-access-and-management	C	C	X	-	X
dpa-reliable-access	C	-	X	X	X
dpa-reliable-access-and-management	C	C	X	X	X

X : present    C : present with initiator the consumer    - : absent

Table 1 - Application-contexts for the DPA access protocol

In the case of the DPA-Server, support for the dpa-access application-context is mandatory, and support for the other application-contexts is optional.

In the case of the DPA-User, support for each application-contexts is optional.

Figure 1 models an application-context between a DPA-User and a DPA-Server. The consumer role of the DPA-User ASE and the supplier role of the DPA-Server ASE, is indicated by the subscript "c" or "s".

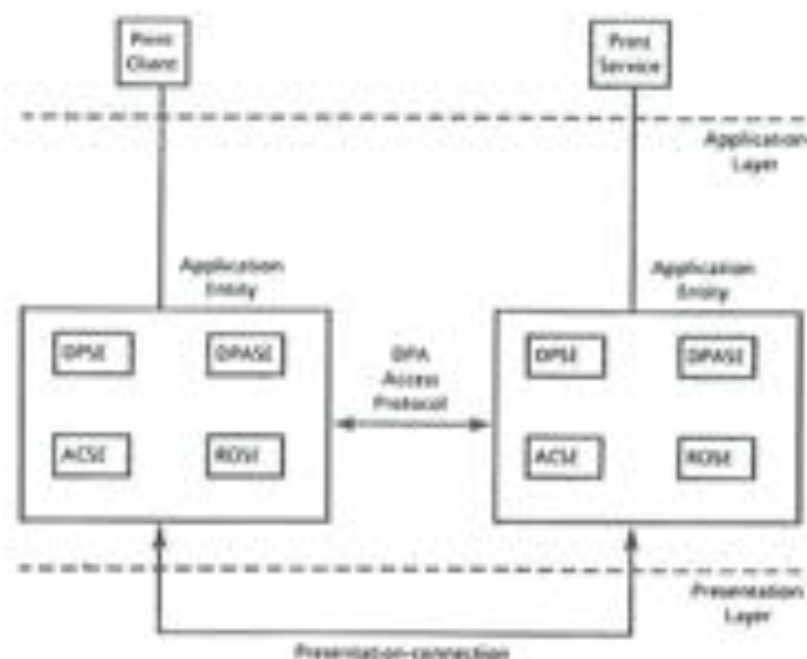


Fig. 1. - Application-context between a DPA-User and a DPA-Server

### 3.2 Services Provided by the DPA Access Protocol

The DPA Access Protocol comprises the following operations which provide the services defined in Part 1 of this Standard.

- DP-bind and DP-unbind
  - DP-bind
  - DP-unbind
- Document Print Service Element (DPSE)
  - Print
  - Modify-job
  - Cancel-job
  - List-object-attributes
- Document Printing Administration Service Element (DPASE)
  - Promote-job
  - Interrupt
  - Pause
  - Resume

### 3.3 Use of Underlying Services

The DPA Access Protocol makes use of underlying services as defined in ISO/IEC 9072-1 and the following additional complements.

#### 3.3.1 Use of ROSE Services

The remote operations of the DPA Access Protocol are class 2 (asynchronous) operations.

### 3.3.2 Use of RTSE Services

When included in an application-context, the RTSE is used in the normal mode, this implies the use of the normal mode of the ACSE and of the presentation.

### 3.3.3 Use of ACSE Services

If the RTSE is not included in an application-context, the DPA-bind and DPA-unbind are the sole users of the A-ASSOCIATE and A-RELEASE services of the ACSE in the normal mode. The ROSE is the sole user of the A-ABORT and A-P-ABORT services of the ACSE.

If the RTSE is included in an application-context, the RTSE is the sole user of the ACSE services.

## 4. DPA ACCESS PROTOCOL ABSTRACT SYNTAX DEFINITION

The abstract-syntax of the DPA Access Protocol is defined using the abstract syntax notation (ASN.1) defined in ISO/IEC 8824, and the remote operations notation defined in ISO/IEC 9072-1.

```
DPAAccessProtocol { iso-identified-organization(3) idc-ecma(0012) standard(0) number(140)
  modules(0) access-protocol(4)
  DEFINITIONS IMPLICIT TAGS ::=
  BEGIN
  -- PROLOGUE --
  EXPORTS
    -- DPA Application Service Elements --
    dPSE, dPASE;
  IMPORTS
    -- Application Service Elements and Application Context --
    APPLICATION-SERVICE-ELEMENT, APPLICATION-CONTEXT, aCSE
    FROM Remote-Operations-Notation-extension { joint-iso-ccitt
    remote-operations(4) notation-extension(2) };
  rTSE
    FROM Reliable-Transfer-APDUs { joint-iso-ccitt reliable-transfer(3) apdus(0) };
  -- DPA Abstract Service Parameters --
  DpBind, DpUnbind, Print, ModifyJob, CancelJob, ListObjectAttributes,
  PromoteJob, Interrupt, Pause, Resume
  FROM DPAAbstractService { iso-identified-organization(3) idc-ecma(0012) standard(0)
  number(140) modules(0) abstract-service (1) };
  -- Object Identifiers --
  id-ac-dpa-access, id-ac-dpa-reliable-access, id-ac-dpa-access-and-management,
  id-ac-dpa-reliable-access-and-management, id-as-acse, id-as-dpse, id-as-dpase, id-as-dpa,
  id-as-dpa-rtse, id-ase-dpse, id-ase-dpase
  FROM DPAProtocolObjectIdentifiers { iso-identified-organization(3) idc-ecma(0012)
  standard(0) number(140) modules(0) protocol-object-identifiers(5) };
```

-- Application Context owning RTSE--

dpa-access APPLICATION-CONTEXT

APPLICATION SERVICE ELEMENTS { aCSE }  
BIND DpBind  
UNBIND DpUnbind  
REMOTE OPERATIONS { rOSE }  
INITIATOR CONSUMER OF { dPSE }  
ABSTRACT SYNTAXES {  
id-as-acse, -- of ACSE --  
id-as-dpse, -- of DPSE, including ROSE --  
id-as-dpa } -- of DPABind and DPABind--  
::= id-ac-dpa-access

dpa-access-and-management APPLICATION-CONTEXT

APPLICATION SERVICE ELEMENTS { aCSE }  
BIND DpBind  
UNBIND DpUnbind  
REMOTE OPERATIONS { rOSE }  
INITIATOR CONSUMER OF { dPSE, dPASE }  
ABSTRACT SYNTAXES {  
id-as-acse, -- of ACSE --  
id-as-dpse, -- of DPSE, including ROSE --  
id-as-dpase, -- of DPASE, including ROSE --  
id-as-dpa } -- of DpBind and DpUnbind--  
::= id-ac-dpa-access-and-management

-- Application Context including RTSE--

dpa-reliable-access APPLICATION-CONTEXT

APPLICATION SERVICE ELEMENTS { aCSE, rTSE }  
BIND DpBind  
UNBIND DpUnbind  
REMOTE OPERATIONS { rOSE }  
INITIATOR CONSUMER OF { dPSE }  
ABSTRACT SYNTAXES {  
id-as-acse, -- of ACSE --  
id-as-dpse, -- of DPSE, including ROSE --  
id-as-dpa-rtse } -- of DpBind and DpUnbind including RTSE--  
::= id-ac-dpa-reliable-access

dpa-reliable-access-and-management APPLICATION-CONTEXT

APPLICATION SERVICE ELEMENTS { aCSE }

  BIND DpBind

  UNBIND DpUnbind

  REMOTE OPERATIONS { rOSE }

  INITIATOR CONSUMER OF { dPSE, dPASE }

  ABSTRACT SYNTAXES {

    id-as-acse, -- of ACSE --

    id-as-dpse, -- of DPSE, including ROSE --

    id-as-dpase, -- of DPASE, including ROSE --

    id-as-dpa-rtse } -- of DpBind and DpUnbind including RTSE --

  ::= id-ac-dpa-reliable-access-and-management

-- DPA Service Element --

dPSE APPLICATION-SERVICE-ELEMENT

  CONSUMER INVOKES {

    print,

    modify-job,

    cancel-job,

    list-object-attributes }

  SUPPLIER INVOKES { }

  ::= id-ase-dpse

dPASE APPLICATION-SERVICE-ELEMENT

  CONSUMER INVOKES {

    interrupt,

    pause,

    resume,

    promote-job }

  SUPPLIER INVOKES { }

  ::= id-ase-dpase

-- Remote Operations --

print	Print	::= 01
modify-job	ModifyJob	::= 02
cancel-job	CancelJob	::= 03
list-object-attributes	ListObjectAttributes	::= 04
promote-job	PromoteJob	::= 05
interrupt	Interrupt	::= 06
resume	Resume	::= 07
pause	Pause	::= 08

-- Remote Errors --

attribute-error	AttributeError	::= 01
name-error	NameError	::= 02
access-error	AccessError	::= 03
update-error	UpdateError	::= 04
security-error	SecurityError	::= 09
service-error	ServiceError	::= 10
document-access-error	DocumentAccessError	::= 11
printer-error	PrinterError	::= 12
selection-error	SelectionError	::= 13

END -- of DPAAccessProtocol --

## 5. CONFORMANCE

A DPA system claiming conformance to the DPA Access Protocol specified in this Standard shall comply with the requirements noted below.

### 5.1 Statement Requirements

The following shall be stated:

- the type of system for which conformance is claimed, DPA-user or DPA-server;
- the application-contexts defined in clause 4 for which conformance is claimed.

Table 2 classifies the support for application-contexts required for conformance to the DPA Access Protocol.

application context	DPA-user	DPA-server
dpa-access	Optional	Mandatory
dpa-access-and-management	Optional	Optional
dpa-reliable-access	Optional	Optional
dpa-reliable-access-and-management	Optional	Optional

Table 2 - Required support for application-contexts

### 5.2 Static Requirements

The system shall

- conform to the abstract-syntax definition(s) of the DPA Access Protocol defined in clause 4 of this part of this Standard required by the application-contexts for which conformance is claimed;
- conform to the Basic-Attribute-Set defined in Part 1 of this Standard;
- implement functional support for an attribute and attribute values in all operations defined in part 1, if they are implemented in any operation;
- implement functional support for standard-defined attributes and/or standard-defined attribute values, rather than, or in addition to implementation-defined attributes and/or attribute values with the same semantic definitions;
- implement the full range of values for integer and string attribute-types for any attributes of such types that are implemented.



### 5.3 Dynamic Requirements

The system shall

- conform to the wrapping onto used services, required by the application context defined in clause 4 of this part of this standard for which conformance is claimed;
- conform to the use of underlying services as defined in 3.3.



APPENDIX PART 2



### Appendix A Formal Assignment of Protocol Object Identifiers

All Object Identifiers this Part of the Standard assigns are formally assigned in the present appendix using ASN.1. The specified values are cited in the ASN.1 module of section 4 of this Part of this Standard.

This appendix is definitive for all values except those for ASN.1 modules of this Part of this Standard. The definitive assignments for those occur in the modules themselves.

---

```
DPAProtocolObjectIdentifiers { iso-identified-organization(3) idc-ecma(0012) standard(0) number(140)
modules(0) protocol-object-identifiers(5) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- PROLOGUE --

EXPORTS EVERYTHING ;
IMPORTS -- nothing -- ;

-- DPA Protocol --
ID ::= OBJECT IDENTIFIER
id-dpa-protocol ID ::= { iso-identified-organization(3) idc-ecma(0012) standard(0) number(140)
protocol(9) }

-- Contexts --
id-ac ID ::= { id-dpa-protocol 1 }-- application context --
id-as ID ::= { id-dpa-protocol 2 }-- abstract syntax --
id-ase ID ::= { id-dpa-protocol 3 }-- application service element --

-- Application Context --
id-ac-dpa-access ID ::= { id-ac 0 }
id-ac-dpa-access-and-management ID ::= { id-ac 1 }
id-ac-dpa-reliable-access ID ::= { id-ac 2 }
id-ac-dpa-reliable-access-and-management ID ::= { id-ac 3 }

-- Abstract Syntax --
id-as-acse ID ::= { id-as 0 }
id-as-dpse ID ::= { id-as 1 }
id-as-dpase ID ::= { id-as 2 }
id-as-dpa ID ::= { id-as 3 }
id-as-dpa-rpse ID ::= { id-as 4 }

-- Application Service Element --
id-ase-dpse ID ::= { id-ase 0 }
id-ase-dpase ID ::= { id-ase 1 }

END -- of DPAProtocolObjectIdentifiers --
```

6

14

6

6

