

INTERNATIONAL STANDARD

ISO/IEC 10175-1

First edition
1996-09-01

Information technology — Text and office systems — Document Printing Application (DPA) —

Part 1:

Abstract service definition and procedures

Technologies de l'information — Bureautique — Application impression de documents (DPA) —

Partie 1: Définition de service abstrait et procédures

This material is reproduced from ISO documents under International Organization for Standardization (ISO) Copyright License number IHS/ICC/1996. Not for resale. No part of these ISO documents may be reproduced in any form, electronic retrieval system or otherwise, except as allowed in the copyright law of the country of use, or with the prior written consent of ISO (Case postale 56, 1211 Geneva 20, Switzerland, Fax +41 22 734 10 79), IHS or the ISO Licensor's members.



Reference number
ISO/IEC 10175-1:1996(E)

AVA Documents
Document Sharing Hub

ISO/IEC 10175-1:1996(E)

Contents

Section 1 - General	1
1 Scope	1
2 Normative references	2
3 Definitions	4
3.1 General terminology	4
3.1.1 OSI basic reference model definitions	4
3.1.2 OSI basic reference model Security part definitions	4
3.1.3 Presentation service definition	5
3.1.4 Abstract syntax notation definitions	5
3.1.5 Font information interchange definitions	5
3.1.6 Directory definitions	5
3.1.7 Abstract service definition conventions definitions	5
3.1.8 Distributed-office-applications model - General model definitions	6
3.1.9 Distributed-office-applications model - DOR definitions	6
3.2 Specific terminology	6
3.2.1 administrator, system administrator	6
3.2.2 assured reproduction area	6
3.2.3 attribute	6
3.2.4 attribute type	6
3.2.5 available	6
3.2.6 busy state	7
3.2.7 composition and layout process	7
3.2.8 compulsory attribute	7
3.2.9 Control-Attribute-Package (CAP)	7
3.2.10 document	7
3.2.11 document copy	7
3.2.12 document-format	7
3.2.13 document production instructions (DPI)	7
3.2.14 document set	7
3.2.15 final form document	7
3.2.16 imposition	7

© ISO/IEC 1996

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

3.2.17 initial-value-job.....	8
3.2.18 job copy	8
3.2.19 job result set.....	8
3.2.20 logical medium	8
3.2.21 logical printer	8
3.2.22 logical resource.....	8
3.2.23 medium.....	8
3.2.24 non-compulsory attribute.....	8
3.2.25 object states.....	8
3.2.26 on-request state	8
3.2.27 output device.....	9
3.2.28 page.....	9
3.2.29 page image.....	9
3.2.30 page description language	9
3.2.31 physical medium	9
3.2.32 physical page	9
3.2.33 physical printer	9
3.2.34 physical resource.....	9
3.2.35 presentation	9
3.2.36 print-access-protocol	9
3.2.37 print-client.....	9
3.2.38 print-job.....	9
3.2.39 print operation management instructions (PMI)	10
3.2.40 print operation parameters (POP)	10
3.2.41 print-request	10
3.2.42 print-server	10
3.2.43 print-service.....	10
3.2.44 print-system	10
3.2.45 printing.....	10
3.2.46 privilege-attribute-certificate (PAC)	10
3.2.47 ready state.....	10
3.2.48 Reference Coordinate System	11
3.2.49 resource	11
3.2.50 revisable (form) document.....	11
3.2.51 service request.....	11
3.2.52 sheet	11
3.2.53 signature.....	11
3.2.54 SPDL document	11
3.2.55 supported	11
3.2.56 unavailable state.....	11
3.2.57 user.....	11
4 Abbreviations	11
5 Conventions.....	13
5.1 Conventions for abstract-services	13
5.2 Conventions for documentation in general	13
5.3 Conventions related to ASN.1	13

Section 2 - DP abstract-service definition	15
6 DP abstract model	15
6.1 Objects in the Document Printing environment	16
6.1.1 DP-Server object	17
6.1.2 DP-User object	17
6.1.3 DP-Administrator object	17
6.2 Ports used in the Document Print Service model	17
6.2.1 DP-User port	17
6.2.2 DP-Administration port	18
6.3 DPA object classes	18
6.3.1 Job object class	19
6.3.2 Document object class	20
6.3.3 Printer object class	20
6.3.4 Server object class	20
6.3.5 Medium object class	21
6.3.6 Font object class	21
6.3.7 Resource object class	21
6.3.8 Transfer-method object class	22
6.3.9 Delivery-method object class	22
6.3.10 Auxiliary-sheet-package object class	22
6.3.11 Auxiliary-sheet object class	22
6.3.12 Finishing object class	22
6.3.13 Output-method object class	22
6.3.14 Imposition object class	23
6.3.15 Scheduler object class	23
6.3.16 Initial-value-job object class	23
6.3.17 Initial-value-document object class	23
6.3.18 Resource-context object class	23
6.3.19 Derived classes and class inheritance for implementor extensions	24
6.3.20 Events and event classes	24
6.4 Attributes	24
6.4.1 The attribute-type	25
6.4.2 Attribute-id	25
6.4.3 Attribute-values	26
6.4.4 Attribute-type definition	26
6.4.4.1 The ATTRIBUTE macro	26
6.4.4.2 Attribute-syntax definition	27
6.4.4.3 The ATTRIBUTE-SYNTAX macro	27
6.4.5 Filters	28
6.4.5.1 Filter-item	29
6.4.5.2 Attribute-value-assertion	31
6.4.6 Compulsory and non-compulsory attributes	31
6.5 Document Processing and Print Operation Models	32
6.5.1 Objects in the DP-User environment	34
6.5.2 Objects in the DP-Server environment	35
6.5.3 Resources	35
6.5.4 Reference Coordinate System (RCS)	36
6.6 Security in DPA	37
6.6.1 Authentication	37
6.6.2 Access authorization	38

7 Abstract-association information	38
7.1 Abstract-bind-operation	38
7.1.1 Bind-argument Parameters	39
7.1.2 Bind-result parameters	40
7.1.3 Bind-error parameters	40
7.2 Abstract-unbind operation	40
8 Abstract-operations.....	41
8.1 Common datatypes and values used in DPA abstract-operations	41
8.1.1 Imported datatypes	41
8.1.2 Datatypes common for most DPA abstract-operations.....	41
8.1.2.1 Priority	41
8.1.2.2 Privileges.....	42
8.1.2.3 Operation-locale	42
8.1.2.4 Default-delivery-addresses	44
8.1.3 Upper and lower bounds.....	44
8.2 DP-User Port Abstract-Operation Definitions	45
8.2.1 Print.....	45
8.2.1.1 Print-argument	46
8.2.1.2 Print-result	47
8.2.1.3 Print Abstract-errors	48
8.2.2 Modify-job	48
8.2.2.1 Modify-job-argument.....	49
8.2.2.2 Modify-job-result	51
8.2.2.3 Modify-job Abstract-errors.....	51
8.2.3 Cancel-job.....	51
8.2.3.1 Cancel-job-argument	51
8.2.3.1.1 Cancelling a specified document.....	52
8.2.3.1.2 Cancelling the entire print-job	52
8.2.3.2 Cancel-job-result.....	52
8.2.3.3 Cancel-job Abstract-errors	52
8.2.4 List-object-attributes.....	52
8.2.4.1 List-object-attributes-argument	53
8.2.4.2 List-object-attributes-result	58
8.2.4.3 List-object-attributes Abstract-errors	61
8.2.4.4 Listing Objects of Derived Classes	61
8.2.4.4.1 Listing specified object instances	62
8.2.4.4.2 Listing all objects of a specified object class.....	63
8.3 DP-Administration Port Abstract-Operation Definitions	64
8.3.1 Promote-job.....	64
8.3.1.1 Promote-job-argument	64
8.3.1.2 Promote-job-result	64
8.3.1.3 Promote-job Abstract-errors	64
8.3.2 Interrupt-job.....	64
8.3.2.1 Interrupt-job-argument	65
8.3.2.2 Interrupt-job-result.....	65
8.3.2.3 Interrupt-job Abstract-errors	65

8.3.3 Pause-job.....	66
8.3.3.1 Pause-job-argument	66
8.3.3.2 Pause-job-result	66
8.3.3.3 PauseJob Abstract-errors	67
8.3.4 Resume-job	67
8.3.4.1 Resume-job-argument	67
8.3.4.2 Resume-job-result	68
8.3.4.3 ResumeJob Abstract-errors.....	68
8.4 Abstract-Errors	68
8.4.1 Access-error	68
8.4.2 Attribute-error.....	69
8.4.3 Document-access-error.....	71
8.4.4 Printer-error.....	72
8.4.5 Security-error.....	72
8.4.6 Selection-error	73
8.4.7 Service-error	73
8.4.8 Update-error.....	75
8.4.9 Error Precedence.....	75
8.5 Abstract-Events.....	75
8.5.1 Event-aborted class	76
8.5.2 Event-error class	76
8.5.3 Event-warning class	76
8.5.4 Event-report class.....	77
8.5.5 Event-state-changed class.....	78
8.5.6 Event Class identifiers.....	78
Section 3 - Print-service attributes	79
9 Print-service attributes.....	79
9.1 Overview of the attribute concept.....	79
9.1.1 Non-compulsory and compulsory attributes	79
9.1.2 Empty attribute-values versus the "none" attribute-value and server defaulting.....	79
9.1.3 SINGLE VALUE and MULTI VALUE attributes.....	80
9.1.4 Conformance to the attribute definitions.....	80
9.1.5 Common attribute syntaxes and datatypes	81
9.1.5.1 Text and textSyntax	81
9.1.5.2 DescriptiveName and descriptiveNameSyntax	82
9.1.5.3 Descriptor and descriptorSyntax	83
9.1.5.4 Message and messageSyntax.....	83
9.1.5.5 ErrorMessage and errorMessageSyntax.....	84
9.1.5.6 SimpleName and simpleNameSyntax.....	84
9.1.5.7 DistinguishedNameString and distinguishedNameStringSyntax.....	85
9.1.5.8 DistinguishedNameStringSequence and distinguishedNameStringSequenceSyntax.....	86
9.1.5.9 Delta time and deltaSyntax	86
9.1.5.10 Generalized time and generalizedTimeSyntax.....	86
9.1.5.11 Integer and integerSyntax	86
9.1.5.12 IntegerSequence and integerSequenceSyntax	87
9.1.5.13 Cardinal and cardinalSyntax	87
9.1.5.14 CardinalSequence and cardinalSequenceSyntax.....	87
9.1.5.15 PositiveInteger and positiveIntegerSyntax	87
9.1.5.16 IntegerRange and integerRangeSyntax.....	87

9.1.5.17 CardinalRange and cardinalRangeSyntax	88
9.1.5.18 MaximumInteger and maximumIntegerSyntax	88
9.1.5.19 MinimumInteger and minimumIntegerSyntax	88
9.1.5.20 Integer64 and integer64Syntax	89
9.1.5.21 Integer64Sequence and integer64SequenceSyntax	89
9.1.5.22 Cardinal64 and cardinal64Syntax	89
9.1.5.23 Cardinal64Sequence and cardinal64SequenceSyntax	89
9.1.5.24 PositiveInteger64 and positiveInteger64Syntax	89
9.1.5.25 Integer64Range and integer64RangeSyntax	90
9.1.5.26 Cardinal64Range and cardinal64RangeSyntax	90
9.1.5.27 MaximumInteger64 and maximumInteger64Syntax	90
9.1.5.28 MinimumInteger64 and minimumInteger64Syntax	91
9.1.5.29 Real and realSyntax	91
9.1.5.30 RealSequence and realSequenceSyntax	91
9.1.5.31 NonNegativeReal and nonNegativeRealSyntax	91
9.1.5.32 RealRange and realRangeSyntax	91
9.1.5.33 NonNegativeRealRange and nonNegativeRealRangeSyntax	92
9.1.5.34 Boolean and booleanSyntax	92
9.1.5.35 Percent and percentSyntax	92
9.1.5.36 ObjectIdentifier and objectIdentifierSyntax	92
9.1.5.37 ObjectIdentifierSequence and objectIdentifierSequenceSyntax	93
9.1.5.38 NameOrOid and nameOrOidSyntax	93
9.1.5.39 NameOrOidSequence and nameOrOidSequenceSyntax	93
9.1.5.40 DistinguishedName and distinguishedNameSyntax	93
9.1.5.41 RDNSequence and rdnSequenceSyntax	94
9.1.5.42 Realization and realizationSyntax	94
9.1.5.43 MediumDimensions and mediumDimensionsSyntax	94
9.1.5.44 Dimension and dimensionSyntax	95
9.1.5.45 XYDimensions and xyDimensionsSyntax	95
9.1.5.46 Locations and locationsSyntax	95
9.1.5.47 Area and areaSyntax	96
9.1.5.48 AreaSequence and areaSequenceSyntax	96
9.1.5.49 Edge and edgeSyntax	96
9.1.5.50 FontReference and fontReferenceSyntax	97
9.1.5.51 CardinalOrOid and cardinalOrOidSyntax	97
9.1.5.52 OidCardinalMap and oidCardinalMapSyntax	97
9.1.5.53 CardinalOrNameOrOid and cardinalOrNameOrOidSyntax	97
9.1.5.54 PositiveIntegerOrOid and positiveIntegerOrOidSyntax	98
9.1.5.55 EventHandlingProfile and eventHandlingProfileSyntax	98
9.1.6 Generic attributes, their datatypes, and syntaxes	99
9.1.6.1 Object-class	100
9.1.6.2 Descriptor	101
9.1.6.3 Descriptive-name	101
9.1.6.4 State	101
9.1.6.5 Message	102
9.1.6.6 Availability	102
9.1.6.7 List-of-managers	103
9.1.6.8 Notification-profile	103
9.1.6.9 Logging-profile	103
9.1.6.10 Associated-server	103
9.1.7 Internet-related attributes	104

9.2 Job Object attributes.....	104
9.2.1 Job-information attributes	104
9.2.1.1 Job-identifier	105
9.2.1.2 Job-identifier-on-printer.....	105
9.2.1.3 Job-identifier-on-client.....	105
9.2.1.4 Job-client-id.....	106
9.2.1.5 Job-name	106
9.2.1.6 Job-originator	106
9.2.1.7 Job-owner	107
9.2.1.8 Job-comment	107
9.2.1.9 Job-start-message.....	107
9.2.1.10 Job-end-message	107
9.2.1.11 Job-non-compulsory-attributes	108
9.2.1.12 Initial-value-job	108
9.2.2 Job-results-handling attributes.....	108
9.2.2.1 Results-profile	108
9.2.2.2 Job-finishing	111
9.2.2.3 Finishing-includes-job-sheets.....	112
9.2.2.4 Job-sheets	112
9.2.2.5 Document-sheets.....	113
9.2.3 Job-event-handling attributes	113
9.2.3.1 Job-abort-criteria	113
9.2.3.2 Job-warning-criteria.....	114
9.2.3.3 Document-abort-criteria.....	114
9.2.4 Job-scheduling-instruction attributes	115
9.2.4.1 Job-start-wait.....	115
9.2.4.2 Job-end-wait	115
9.2.4.3 Job-hold.....	116
9.2.4.4 Job-password	116
9.2.4.5 Job-print-after.....	116
9.2.4.6 Job-priority	117
9.2.4.7 Job-deadline-time	117
9.2.4.8 Job-discard-time	117
9.2.4.9 Job-retention-period.....	118
9.2.4.10 Job-message-to-operator.....	118
9.2.4.11 Job-scheduling	118
9.2.4.12 Job-validate.....	119
9.2.5 Printer-selection attributes.....	119
9.2.5.1 Printer-name-requested.....	119
9.2.5.2 Physical-printers-requested	120
9.2.5.3 Printer-locations-requested	120
9.2.5.4 Printer-models-requested	120
9.2.5.5 Printer-types-requested	120
9.2.5.6 Printer-speed-range-requested.....	120
9.2.6 Access-and-accounting attributes	121
9.2.6.1 User-name	121
9.2.6.2 Accounting-information.....	121
9.2.7 Job-security-handling	121
9.2.7.1 Job-security-confidentiality-level.....	122
9.2.7.2 Job-security-confidentiality-category	122
9.2.7.3 Job-security-confidentiality-policy	123
9.2.7.4 Job-security-integrity-level.....	123

9.2.7.5 Job-security-integrity-category	124
9.2.7.6 Job-security-integrity-policy	124
9.2.7.7 Job-security-label	124
9.2.8 Job-status	125
9.2.8.1 Current-job-state	125
9.2.8.1.1 Job state transition	127
9.2.8.2 Job-state-reasons	129
9.2.8.3 Previous-job-state	129
9.2.8.4 Printers-assigned	130
9.2.8.5 Printer-state-of-printers-assigned	130
9.2.8.6 Estimated-completion-time	131
9.2.8.7 Submission-time	131
9.2.8.8 Modification-time	131
9.2.8.9 Started-printing-time	131
9.2.8.10 Job-copies-completed	131
9.2.8.11 Pages-completed	132
9.2.8.12 Impressions-completed	132
9.2.8.13 Media-sheets-completed	132
9.2.8.14 Octets-completed	133
9.2.8.15 Total-job-octets	133
9.2.8.16 Intervening-jobs	134
9.2.8.17 Print-checkpoint	134
9.2.8.18 New-job-identifier	135
9.2.8.19 On-request-resources-required	135
9.2.8.20 Job-message-from-administrator	135
9.2.8.21 Last-job-events	135
9.2.8.22 Name-of-last-accessor	136
9.2.8.23 Error-count	136
9.2.8.24 Warning-count	136
9.2.8.25 Processing-time	136
9.2.8.26 Completion-time	136
9.2.8.27 Number-of-documents	137
9.2.8.28 Job-submission-complete	137
9.2.8.29 Ignored-attributes	137
9.2.9 Job size attributes	138
9.2.9.1 Job-page-count	138
9.2.9.2 Job-impression-count	139
9.2.9.3 Job-media-sheet-count	139
9.3 Document object attributes	140
9.3.1 Document-information attributes	140
9.3.1.1 Document-name	140
9.3.1.2 Document-file-name	141
9.3.1.3 Document-authors	141
9.3.1.4 Document-comment	141
9.3.1.5 Document-revision-date	141
9.3.2 Document production-instructions	142
9.3.2.1 Default-medium	142
9.3.2.2 Default-input-tray	143
9.3.2.3 Default-font	144
9.3.2.4 Default-resources	144

9.3.2.5 Default-character-set.....	145
9.3.2.5.1 Assignment of object identifiers for coded character sets.....	145
9.3.2.5.2 Representation of coded character sets in documents and attributes.....	146
9.3.2.6 Default-character-repertoire	151
9.3.2.7 Default-character-mapping	152
9.3.2.8 Default-printer-resolution	152
9.3.2.9 Medium-substitution	153
9.3.2.10 Input-tray-select.....	153
9.3.2.11 Font-substitution	154
9.3.2.12 Resource-context.....	154
9.3.2.13 Finishing	155
9.3.2.14 Finishing-includes-document-sheets	155
9.3.2.15 Output	155
9.3.2.16 Imposition attributes	156
9.3.2.16.1 Number-up	156
9.3.2.16.2 Plex	160
9.3.2.16.3 Binding-edge	161
9.3.2.16.4 Binding-edge-image-shift	162
9.3.2.16.5 x-image-shift	162
9.3.2.16.6 y-image-shift	162
9.3.2.17 Print-quality.....	163
9.3.2.18 Medium-fidelity	163
9.3.2.19 Sides	164
9.3.2.20 Page-select.....	164
9.3.2.21 Page-media-select	165
9.3.2.22 Copy-count	166
9.3.2.23 Reset-printer	166
9.3.2.24 Initial-value-document.....	166
9.3.3 Document-description attributes	167
9.3.3.1 Document-type	167
9.3.3.2 Transfer-method	168
9.3.3.3 Document-format.....	168
9.3.3.4 Document-content.....	169
9.3.3.5 Non-compulsory-attributes	169
9.3.4 Document-characteristic attributes	169
9.3.4.1 Intended-page-size	170
9.3.4.2 Content-orientation.....	171
9.3.4.3 Presentation-direction	172
9.3.4.4 Page-order-received	173
9.3.4.5 Media-used	174
9.3.4.6 Assured-reproduction-areas-used	174
9.3.4.7 Input-trays-used	174
9.3.4.8 Fonts-used	175
9.3.4.9 Resources-used	175
9.3.4.10 Character-sets-used.....	175
9.3.4.11 Character-repertoires-used	175
9.3.4.12 Character-mappings-used	175
9.3.4.13 Printer-resolutions-used	176
9.3.4.14 Content-orientations-used	176
9.3.4.15 Presentation-directions-used	176
9.3.4.16 System-hints	176
9.3.4.16.1 Octet-count.....	177
9.3.4.16.2 Page-count.....	177

9.3.4.16.3 Glyph-count.....	177
9.3.4.16.4 Font-count.....	177
9.3.4.16.5 Font-change-count.....	177
9.3.4.16.6 Maximum-fonts-per-page.....	177
9.3.4.16.7 Percent-graphics.....	178
9.3.4.16.8 Percent-images.....	178
9.3.5 Document-status attributes.....	178
9.3.5.1 Document-sequence-number.....	178
9.3.5.2 Document-state.....	178
9.3.5.3 Copies-completed.....	179
9.3.5.4 File-reference.....	179
9.3.5.5 Current-page-order.....	180
9.4 Printer Object attributes.....	181
9.4.1 Printer-name.....	182
9.4.2 Printer-name-aliases.....	182
9.4.3 Printer-locations.....	182
9.4.4 Printer-model.....	183
9.4.5 Printer-types.....	183
9.4.6 Printer-realization.....	184
9.4.7 Maximum-printer-speed.....	184
9.4.8 Printer-resolutions-supported.....	184
9.4.9 Printer-state.....	185
9.4.10 Printer-associated-printers.....	186
9.4.11 Printers-ready.....	186
9.4.12 Printer-initial-value-job.....	186
9.4.13 Printer-initial-value-document.....	187
9.4.14 Media-supported.....	187
9.4.15 Media-ready.....	187
9.4.16 Medium-source-sizes-supported.....	187
9.4.17 Medium-source-sizes-ready.....	189
9.4.18 Medium-fidelities-supported.....	189
9.4.19 Fonts-supported.....	189
9.4.20 Fonts-ready.....	189
9.4.21 Resources-supported.....	190
9.4.22 Resources-ready.....	190
9.4.23 Printer-resource-context.....	190
9.4.24 Job-finishings-supported.....	190
9.4.25 Job-finishings-ready.....	190
9.4.26 Finishing-includes-job-sheets-supported.....	191
9.4.27 Finishings-supported.....	191
9.4.28 Finishings-ready.....	191
9.4.29 Finishing-includes-doc-sheets-supported.....	191
9.4.30 Outputs-supported.....	191
9.4.31 Outputs-ready.....	192
9.4.32 Input-trays-supported.....	192
9.4.33 Input-trays-ready.....	192
9.4.34 Input-trays-medium.....	192
9.4.35 Output-bins-supported.....	193
9.4.36 Output-bins-ready.....	193
9.4.37 Output-bins-characteristics.....	193
9.4.38 Print-qualities-supported.....	194
9.4.39 Print-qualities-ready.....	194

9.4.40 Job-sheets and document-sheets.....	195
9.4.40.1 Job-sheets-supported	195
9.4.40.2 Job-sheets-ready	195
9.4.40.3 Document-sheets-supported	195
9.4.40.4 Document-sheets-ready	196
9.4.41 Character-sets-supported	196
9.4.42 Character-sets-ready	196
9.4.43 Character-repertoires-supported	196
9.4.44 Character-repertoires-ready	196
9.4.45 Character-mappings-supported.....	197
9.4.46 Character-mappings-ready	197
9.4.47 Start-message-supported.....	197
9.4.48 End-message-supported.....	197
9.4.49 Sides-supported	197
9.4.50 Sides-ready.....	197
9.4.51 Page-select-supported	198
9.4.52 Maximum-copies-supported	198
9.4.53 Document-formats-supported	198
9.4.54 Document-formats-ready	198
9.4.55 Native-document-formats-ready	199
9.4.56 Content-orientations-supported	199
9.4.57 Presentation-directions-supported.....	199
9.4.58 Sheet-count	200
9.4.59 Printer-needs-attention-time.....	200
9.4.60 Printer-needs-key-operator-attention-time	200
9.4.61 Printer-security.....	200
9.4.61.1 Printer-confidentiality-level-range	200
9.4.61.2 Printer-confidentiality-category-sets.....	201
9.4.61.3 Printer-confidentiality-policy	201
9.4.61.4 Printer-integrity-level-range.....	201
9.4.61.5 Printer-integrity-category-sets	201
9.4.61.6 Printer-integrity-policy.....	202
9.4.61.7 Printer-security-label-policy	202
9.4.62 Imposition-object attribute support	202
9.4.62.1 Plexes-supported	202
9.4.62.2 Numbers-up-supported.....	202
9.4.62.3 Binding-edges-supported	203
9.4.62.4 Binding-edge-image-shift-max-supported	203
9.4.62.5 X-image-shift-range-supported	203
9.4.62.6 Y-image-shift-range-supported	203
9.4.63 Schedulers-supported	203
9.4.64 Scheduler-ready	203
9.4.65 Job-size-range-supported.....	204
9.4.66 Job-size-range-ready.....	204
9.4.67 Printer-timeout-period.....	204
9.4.68 Job-abort-criteria-supported.....	204
9.4.69 Job-warning-criteria-supported.....	205
9.4.70 Document-abort-criteria-supported.....	205
9.4.71 Checkpoint-formats-supported.....	205
9.4.72 Initial-value-jobs-supported.....	205
9.4.73 Initial-value-documents-supported.....	206

9.4.74 Common Printer and Server object attributes	206
9.4.74.1 Job-attributes-supported	206
9.4.74.2 Document-attributes-supported	206
9.4.74.3 Multiple-documents-supported	207
9.4.74.4 Cancel-individual-document-supported	207
9.4.74.5 Translators-supported	207
9.4.74.6 Translators-ready	207
9.4.74.7 Job-result-set-delivery-methods-supported	207
9.4.74.8 Job-result-set-delivery-methods-ready	208
9.4.74.9 Notification-delivery-methods-supported	208
9.4.74.10 Notification-delivery-methods-ready	208
9.4.74.11 Log-delivery-methods-supported	208
9.4.74.12 Log-delivery-methods-ready	208
9.4.74.13 Are-foreign-jobs-visible	209
9.4.74.14 Enabled	209
9.5 Server object attributes	210
9.5.1 Server-name	210
9.5.2 Server-name-aliases	211
9.5.3 Server-address	211
9.5.4 Server-state	211
9.5.5 Object-classes-supported	211
9.5.6 Physical-printers-supported	211
9.5.7 Physical-printers-ready	211
9.5.8 Logical-printers-supported	212
9.5.9 Logical-printers-ready	212
9.5.10 Job-validates-supported	212
9.5.11 Events-supported	212
9.5.12 Modify-individual-document-supported	212
9.5.13 Transfer-methods-supported	212
9.5.14 Extended-problems-supported	213
9.5.15 Conformance-levels-supported	213
9.5.16 Server-resource-context	213
9.5.17 Job-states-supported	214
9.5.18 Job-state-reasons-supported	214
9.5.19 Document-states-supported	214
9.5.20 Printer-states-supported	214
9.5.21 Default-distinguished-name-syntax	214
9.5.22 Distinguished-name-syntaxes-supported	214
9.5.23 Locale	215
9.5.24 Locales-supported	215
9.6 Medium object attributes	215
9.6.1 Medium-identifier	216
9.6.2 Medium-associated-media	219
9.6.3 Medium-realization	219
9.6.4 Medium-type	220
9.6.5 Medium-size	220
9.6.6 Medium-dimensions	223
9.6.7 Medium-colour	223
9.6.8 Medium-tooth	224
9.6.9 Medium-grain	224
9.6.10 Medium-sides	224
9.6.11 Medium-weight	224

9.6.12 Medium-assured-reproduction-area	225
9.6.13 Medium-form-parts	225
9.6.14 Medium-holes-count	225
9.6.15 Medium-holes-edge	225
9.6.16 Medium-holes-axis-offset	225
9.6.17 Medium-holes-locations	225
9.6.18 Medium-holes-diameter	226
9.7 Font object attributes	226
9.7.1 Font-identifier	227
9.8 Resource object attributes	227
9.8.1 Resource-name	228
9.8.2 Resource-type	228
9.9 Transfer-method object attributes	229
9.9.1 Transfer-method-identifier	229
9.9.2 Transfer-method-type	230
9.9.3 Document-reference-type	230
9.10 Delivery-method object attributes	230
9.10.1 Delivery-method-identifier	231
9.10.2 Delivery-methods for various categories of delivery	231
9.10.2.1 Delivery-methods for hard copy output of a job result set	231
9.10.2.2 Delivery-methods for hardcopy output of logs	233
9.10.2.3 Delivery-methods for soft copy output of job result sets	233
9.10.2.4 Delivery-methods for soft copy output of event notifications and logging	234
9.11 Auxiliary-sheet-package object attributes	235
9.11.1 Auxiliary-sheet-package-identifier	236
9.11.2 Separator-sheet-of-set	239
9.11.3 Start-sheet-of-set	239
9.11.4 End-sheet-of-set	239
9.11.5 Separator-sheet-of-copy	240
9.11.6 Start-sheet-of-copy	240
9.11.7 End-sheet-of-copy	240
9.12 Auxiliary-sheet object attributes	240
9.12.1 Auxiliary-sheet-identifier	241
9.12.2 Auxiliary-sheet-medium	242
9.13 Finishing object attributes	242
9.13.1 Finishing Model	242
9.13.2 Finishing objects	244
9.13.3 Two-stage finishing	244
9.13.4 Generic attributes	245
9.13.5 Finishing-identifier	245
9.13.6 Finishing-associated-finishings	246
9.13.7 Finishing-realization	247
9.13.8 Finishing type and finishing-specification attribute	247
9.13.8.1 Finishing-message	250
9.13.8.2 Named-finishing	250
9.13.8.3 Finishing process parameters	250
9.13.8.3.1 Inherited finishing process parameters	250
9.13.8.3.2 Other generic finishing process parameters	251
9.13.8.3.3 Reference-size	251
9.13.8.3.4 Reference edge	251
9.13.8.3.5 Jog edge	251

9.13.8.3.6 Process offset.....	252
9.13.8.3.7 Head locations	252
9.13.8.4 Finishing process specification.....	252
9.13.8.5 Stitching specification.....	253
9.13.8.5.1 Named stitching	253
9.13.8.5.2 Stitching parameters.....	253
9.13.8.5.3 Stitching type	254
9.13.8.6 binding specification.....	254
9.13.8.6.1 named binding.....	254
9.13.8.6.2 binding parameters	254
9.13.8.6.3 binding type.....	254
9.13.8.6.4 binding colour	255
9.13.8.7 Trimming specification.....	255
9.13.8.7.1 Named trimming	256
9.13.8.7.2 Trimming parameters	256
9.13.8.8 Die cutting specification	256
9.13.8.8.1 Named die cutting	256
9.13.8.8.2 Die cutting parameters.....	257
9.13.8.8.3 Die cut position	257
9.13.8.8.4 Die cut name	257
9.13.8.9 Punching specification.....	257
9.13.8.9.1 Named punching	258
9.13.8.9.2 Punching parameters	258
9.13.8.9.3 Punch diameter.....	258
9.13.8.10 Perforating specification	258
9.13.8.10.1 Named perforation.....	259
9.13.8.10.2 Perforation parameters.....	259
9.13.8.10.3 Perforation type	259
9.13.8.11 Slitting specification	259
9.13.8.11.1 Named slitting	259
9.13.8.11.2 Slitting parameters	260
9.13.8.12 Insert specification.....	260
9.13.8.12.1 Named inserting	260
9.13.8.12.2 Inserting parameters	260
9.13.8.12.3 Insert sheet list	260
9.13.8.12.4 Insert sheet	260
9.13.8.12.5 Insert identifier.....	261
9.13.8.12.6 Insert name.....	261
9.13.8.12.7 Insert bin	261
9.13.8.12.8 Insert top surface	261
9.13.8.12.9 Insert after	261
9.13.8.12.10 Insert message	261
9.13.8.13 Covers specification	261
9.13.8.13.1 Named covers	261
9.13.8.13.2 Covers parameters	262
9.13.8.13.3 Front cover	262
9.13.8.13.4 Back cover.....	262
9.13.8.14 Folding specification.....	262
9.13.8.14.1 Named folding.....	262
9.13.8.14.2 Folding parameters	262
9.13.8.15 Other finishing specification	263
9.13.8.15.1 Finishing op type	263
9.13.8.15.2 Finishing op specification.....	263

9.13.8.15.3 Named finishing op	263
9.13.8.15.4 Specified finishing op parameters	263
9.13.8.15.5 Additional parameters	264
9.13.8.15.6 Other finishing specification process definition.....	264
9.14 Output-method object attributes	264
9.14.1 Output-identifier	264
9.15 Imposition object attributes	265
9.15.1 Imposition-identifier	266
9.15.2 Imposition-n-up	266
9.15.3 Logical-page-origin.....	266
9.15.4 Logical-page-x-offset.....	268
9.15.5 Logical-page-y-offset.....	268
9.15.6 Logical-page-scaling	268
9.16 Scheduler object attributes	269
9.16.1 Scheduler-identifier.....	269
9.17 Initial-value-job object attributes.....	269
9.17.1 Initial-value-job-identifier	270
9.18 Initial-value-document object attributes.....	270
9.18.1 Initial-value-document-identifier.....	271
9.19 Resource-context object attributes.....	271
9.19.1 Resource-context-identifier.....	272
9.19.2 Context-address.....	272
Section 4 - DPA realization	273
10 Overview.....	273
11 Supply of the DP-User abstract-service.....	273
11.1 Supply of the DP-User port abstract-services	273
11.1.1 Performance of the Print abstract-operation.....	273
11.1.2 Performance of the ModifyJob abstract-operation	275
11.1.3 Performance of the CancelJob abstract-operation.....	275
11.1.4 Performance of the ListObjectAttributes abstract-operation	276
12 Supply of the DP-administrator abstract-service.....	277
12.1 Supply of the DP-Administration port abstract-services.....	277
12.1.1 Performance of the PromoteJob abstract-operation.....	277
12.1.2 Performance of the InterruptJob abstract-operation.....	277
12.1.3 Performance of the PauseJob abstract-operation.....	278
12.1.4 Performance of the ResumeJob abstract-operation	278
12.2 Scheduling jobs on physical printers.....	278
13 Port realization	278

Annex A Formal assignment of DPA standard object identifiers	279
Annex B Formal definition of the DPA abstract service.....	315
Annex C Formal definition of DPA standard attributes.....	332
Annex D Formal definition of DPA standard parameter upper and lower bounds	411
Annex E Conformance requirements	412
E.1 Conformance level 1	412
E.2 Conformance level 2	413
E.3 Conformance level 1M	414
E.4 Conformance level 2M	415
E.5 Conformance requirements for the abstract operations	415
E.5.1 Print operation conformance requirements	415
E.5.1.1 PrintArgument conformance requirements	415
E.5.1.2 PrintResult conformance requirements	416
E.5.2 ModifyJob operation conformance requirements	416
E.5.2.1 ModifyJobArgument conformance requirements	416
E.5.2.2 ModifyJobResult conformance requirements	417
E.5.3 CancelJob operation conformance requirements	417
E.5.3.1 CancelJobArgument conformance requirements	417
E.5.3.2 CancelJobResult conformance requirements	418
E.5.4 ListObjectAttributes operation conformance requirements	418
E.5.4.1 ListObjectAttributesArgument conformance requirements	418
E.5.4.2 ListObjectAttributesResult conformance requirements	419
E.5.5 PromoteJob operation conformance requirements	419
E.5.5.1 PromoteJobArgument conformance requirements	419
E.5.5.2 PromoteJobResult conformance requirements	419
E.5.6 InterruptJob operation conformance requirements	419
E.5.6.1 InterruptJobArgument conformance requirements	419
E.5.6.2 InterruptJobResult conformance requirements	420
E.5.7 PauseJob operation conformance requirements	420
E.5.7.1 PauseJobArgument conformance requirements	420
E.5.7.2 PauseJobResult conformance requirements	420
E.5.8 ResumeJob operation conformance requirements	420
E.5.8.1 ResumeJobArgument conformance requirements	420
E.5.8.2 ResumeJobResult conformance requirements	421
E.6 Conformance requirements for the CHOICE types contained in attributes	421
E.7 Conformance requirements for the OPTIONAL components contained in attributes	422

Annex F Document Format Definitions	424
F.1 Document formats.....	424
F.1.1 Relationship between ISO/IEC 10175 OIDs and IANA enum values	427
F.2 Document-production-instruction (DPI) attributes.....	428
F.2.1 New-line-option	428
F.2.2 Carriage-control-type	429
F.2.3 Table-reference-characters	430
F.2.4 Shift-out-shift-in	430
F.3 Printer object attributes	431
F.3.1 New-line-options-supported	431
F.3.2 Carriage-control-types-supported	431
F.3.3 Table-reference-characters-supported.....	431
F.3.4 Shift-out-shift-in-supported.....	431
F.4 Document-format-variants.....	431
F.4.1 DDIF (Digital Document Interchange Format).....	433
F.4.2 DEC ANSI-Compliant Printing Protocol (DEC-PPL).....	433
F.4.3 HPGL (Hewlett Packard Graphics Language).....	433
F.4.4 Interpress	433
F.4.5 ISO/IEC 6429 (Control Functions for Coded Character Sets)	434
F.4.6 Line-data	434
F.4.7 MODCA (Mixed object document content architecture)	435
F.4.8 PCL (Hewlett Packard Printer Control Language)	435
F.4.9 PostScript	436
F.4.10 PPDS (Page Printer Data Stream)	437
F.4.11 ReGIS (Remote graphics instruction set)	437
F.4.12 SCS (SNA character string)	437
F.4.13 Simple-text	437
F.4.14 SPDL (Standard Page Description Language)	439
F.5 Document-format identifiers	440
F.6 Simple-text Object Identifiers	440
F.7 ASN.1 for document-format specific attributes of ANNEX F.....	440
Annex G List of Object Identifier Characteristics.....	445
Annex H Future Work	482
H.1 Amendments to Part 1 of ISO/IEC 10175	482
H.2 A New Part 3 for ISO/IEC 10175, System Management Operations	482
Index	483

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 10175-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 18, *Document processing and related communication*.

ISO/IEC 10175 consists of the following parts, under the general title *Information technology — Text and office systems — Document Printing Application (DPA)*:

- *Part 1: Abstract service definition and procedures*
- *Part 2: Protocol specification*
- *Part 3: Management abstract service definition and procedures*

Annexes A to E form an integral part of this part of ISO/IEC 10175. Annexes F to H are for information only.

Information technology - Text and office systems - Document Printing Application (DPA) -

Part 1:

Abstract service definition and procedures

Section 1 - General

1 Scope

ISO/IEC 10175 consists of three parts:

- Part 1: Abstract service definitions and procedures
- Part 2: Protocol specification
- Part 3: Management abstract service definition and procedures

This part of ISO/IEC 10175:

- specifies a client-server model of printing in accordance with the Distributed-office-applications Model (ISO/IEC 10031-1);
- specifies functions and services provided by Document Printing Application servers;
- specifies the Document Printing Application abstract service using the principles established by the Abstract Service Definition Conventions (ISO/IEC 10021-3);
- specifies the usage of other services.

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, ISO/IEC 10175 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 the desires of those users. For the purposes of ISO/IEC 10175, 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. ISO/IEC 10175 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 ISO/IEC 10175 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 Open Systems Interconnection (OSI) Basic Reference Model (ISO/IEC 7498-1).

In addition, recognition has been given to the potential need for printing systems to be able to operate within and across diverse networking environments. In particular, the Document Printing Application has contributed to, and benefited from, related standardization activities in other venues such as the Internet Engineering Task Force (IETF), the Desktop Management Task Force, the POSIX project of the IEEE Computer Society, and X/Open.

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 and ancillary equipment. This cycle includes other processes such as document creation and interchange through public and private networks.

ISO/IEC 10175 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-job;
- a capability for printing systems, and associated facilities, to protect against unauthorized printing of documents.

Accordingly, the abstract-service defined by ISO/IEC 10175 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 abstract-service 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 abstract-service also allows users to inquire about jobs, modify the characteristics and progress of jobs, and obtain feedback about a job.

Many different document formats have been developed for printing purposes, and are in wide use. For this reason, the Document Printing Application has been developed with a view toward supporting arbitrary document formats in a transparent manner. That is, the specific content or format of an electronic document is independent of the access protocol defined by ISO/IEC 10175. 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 render the document successfully.

However, in spite of this generality of focus, ISO/IEC 10175 is particularly oriented toward alignment with the Standard Page Description Language (SPDL - ISO/IEC 10180). This will assure that DPA will include the features needed to assist in the transport and faithful rendering of SPDL documents.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 10175. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 10175 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 216: 1975¹⁾, *Writing paper and certain classes of printed matter - Trimmed sizes - A and B series.*

¹⁾ Currently under revision

ISO 269: 1985,	<i>Correspondence envelopes - Designation and sizes.</i>
ISO 639: 1988 ¹⁾ ,	<i>Code for the representation of names of languages.</i>
ISO/IEC 646: 1991,	<i>Information technology - ISO 7-bit coded character set for information interchange.</i>
ISO/IEC 2022: 1994,	<i>Information technology - Character code structure and extension techniques.</i>
ISO 2375: 1985,	<i>Data processing - Procedure for registration of escape sequences.</i>
ISO 3166: 1993 ¹⁾ ,	<i>Codes for the representation of names of countries.</i>
ISO/IEC 6429: 1992,	<i>Information technology - Control functions for coded character sets.</i>
ISO/IEC 6937: 1994,	<i>Information technology - Coded graphic character set for text communication - Latin alphabet.</i>
ISO/IEC 7350: 1991,	<i>Information technology - Registration of repertoires of graphic characters from ISO/IEC 10367.</i>
ISO/IEC 7498-1: 1994,	<i>Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model.</i>
ISO/IEC 7498-2: 1989,	<i>Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture.</i>
ISO 8571-3: 1988,	<i>Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 3: File Service Definition.</i>
ISO 8613-1: 1989,	<i>Information processing - Text and office systems - Office Document Architecture (ODA) and interchange format - Part 1: Introduction and general principles.</i>
ISO/IEC 8822: 1994,	<i>Information technology - Open Systems Interconnection - Presentation service definition.</i>
ISO/IEC 8824: 1990,	<i>Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).</i>
ISO/IEC 8825: 1990,	<i>Information technology - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).</i>
ISO 8859-1: 1987,	<i>Information processing - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1.</i>
ISO/IEC 9541-1: 1991,	<i>Information technology - Font information interchange - Part 1: Architecture.</i>
ISO/IEC 9541-2: 1991,	<i>Information technology - Font information interchange - Part 2: Interchange Format.</i>
ISO/IEC 9594-2: 1990,	<i>Information technology - Open Systems Interconnection - The Directory - Part 2: Models.</i>
ISO/IEC 9594-3: 1990,	<i>Information technology - Open Systems Interconnection - The Directory - Part 3: Abstract service definition.</i>
ISO/IEC 9945-1: 1996,	<i>Information technology - Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C Language] [ANSI/IEEE Std 1003.1].</i>

- ISO/IEC 10021-3: 1990, *Information technology - Text Communication - Message-Oriented Text Interchange Systems (MOTIS)*
- Part 3: Abstract Service Definition Conventions.
- ISO/IEC 10021-4: 1990, *Information technology - Text Communication - Message-Oriented Text Interchange Systems (MOTIS)*
- Part 4: Message Transfer System: Abstract Service Definition and Procedures.
- ISO/IEC 10031-1: 1991, *Information technology - Text and office systems - Distributed-office-applications model*
- Part 1: General model.
- ISO/IEC 10031-2: 1991, *Information technology - Text and office systems - Distributed-office-applications model*
- Part 2: Distinguished-object-reference and associated procedures.
- ISO/IEC 10166-1: 1991, *Information technology - Text and office systems - Document filing and retrieval (DFR)*
- Part 1: Abstract service definition and procedures.
- ISO/IEC 10180: 1995, *Information technology - Processing languages - Standard Page Description Language (SPDL).*
- ISO/IEC 10367: 1991, *Information technology - Standardized coded graphic character sets for use in 8-bit codes.*
- ISO/IEC 10538: 1991, *Information technology - Control functions for text communication.*
- ISO/IEC 10646-1: 1993, *Information technology - Universal Multiple-Octet Coded Character Set (UCS)*
- Part 1: Architecture and Basic Multilingual Plane.
- ISO/IEC 10740-1: 1993, *Information technology - Text and office systems - Referenced Data Transfer (RDT)*
- Part 1: Abstract service definition.
- RFC 1759, *Printer MIB*, Proposed Internet Standard, R. Smith, F. Wright, T. Hastings, S. Zilles, J. Gyllenskog, March 1995.

3 Definitions

3.1 General terminology

3.1.1 OSI basic reference model definitions

This part of ISO/IEC 10175 makes use of the following terms defined in ISO/IEC 7498-1:

- Application Layer
- application-entity
- application-service-element
- Presentation Layer
- protocol
- service definition

3.1.2 OSI basic reference model Security part definitions

This part of ISO/IEC 10175 makes use of the following terms defined in ISO 7498-2:

- access control
- authentication
- authorization
- credentials
- security policy

3.1.3 Presentation service definition

This part of ISO/IEC 10175 makes use of the following term defined in ISO/IEC 8822:

- abstract syntax

3.1.4 Abstract syntax notation definitions

This part of ISO/IEC 10175 makes use of the following terms defined in ISO/IEC 8824:

- ASN.1
- external type
- Generalized Time
- macro
- object identifier
- UTC Time
- PrintableString
- VisibleString
- T61String
- octet string

3.1.5 Font information interchange definitions

This part of ISO/IEC 10175 makes use of the following terms defined in ISO/IEC 9541:

- font
- font reference
- font resource
- glyph
- glyph image
- glyph representation

3.1.6 Directory definitions

This part of ISO/IEC 10175 makes use of the following terms defined in ISO/IEC 9594:

- attribute
- attribute macro
- attribute value
- filter

3.1.7 Abstract service definition conventions definitions

This part of ISO/IEC 10175 makes use of the following terms defined in ISO/IEC 10021-3:

- abstract bind
- abstract error
- abstract model
- abstract operation
- abstract service
- abstract unbind
- argument

asymmetric
port
symmetric

3.1.8 Distributed-office-applications model - General model definitions

This part of ISO/IEC 10175 makes use of the following terms defined in ISO/IEC 10031-1:

accessesee
accessor
distributed-office-application
distinguished-object-reference
privilege-attributes
referenced-object-access
security-attributes
security-object
security-subject

3.1.9 Distributed-office-applications model - DOR definitions

This part of ISO/IEC 10175 makes use of the following terms defined in ISO/IEC 10031-2:

produce-time
Quality of Service

3.2 Specific terminology

For the purposes of ISO/IEC 10175, the following definitions apply.

3.2.1 administrator, system administrator

A person with administrative responsibility for one or more components in a distributed system. Within ISO/IEC 10175, 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 - The decision as to whether a system administrator will be accorded special privileges is a site and/or enterprise specific issue, not mandated by ISO/IEC 10175. The term administrator may be used as the generic equivalent of the formal designator "DP-Administrator"; however, the two are not strictly equivalent, and the formal designator should be used when necessary to refer specifically the actions and objects related to the DP-Administration port.

3.2.2 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.

3.2.3 attribute

A characteristic of an object.

3.2.4 attribute type

A datatype defined by means of ASN.1 notation, used for defining and specifying the attributes of objects.

3.2.5 available

An object is said to be available when it is in either of the two states: *ready* or *on-request*. The term available is used when there is no distinction between ready and on-request.

3.2.6 busy state

That state of an object instance indicating that it is temporarily inaccessible due to dynamic constraints such as resource locking or running out of a resource (memory, paper, toner, etc.).

3.2.7 composition and layout process

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.

3.2.8 compulsory attribute

An attribute is said to be compulsory if a server either 1) shall perform the associated action if the server and printer support the attribute and its values or 2) shall reject the job if the attribute or its values are unsupported by either the server or printer.

3.2.9 Control-Attribute-Package (CAP)

A collection of attributes used to control access to a print-server, or to an object controlled by that print-server.

3.2.10 document

In ISO/IEC 10175, 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 *final form document* and *revisable form document*.

3.2.11 document copy

The output from each copy of a document object; a document set has more than one document copy if the document's copy-count attribute is present and greater than 1.

3.2.12 document-format

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

3.2.13 document production instructions (DPI)

Print operation parameters that affect the appearance of the document or documents to printed in a print-job. Some print operation parameters 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 potential conflict by means of an algorithm that is beyond the scope of ISO/IEC 10175. Different subsets of document production instructions may be defined for different specific document formats.

3.2.14 document set

The output from a single document object of all its document copies; a document set has more than one document copy if the document object's copy-count attribute is present and greater than 1.

3.2.15 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. See *document format*.

3.2.16 imposition

Placement of page images on one side of a sheet. Examples of imposition include (1) the shifting of a page image with respect to the medium on which it is to be rendered, for the purpose of producing a binding margin, and (2) the placement of multiple page images on a single sheet of paper such that when cut, folded and trimmed, the resulting pages will be in the proper orientation and in correct numerical sequence.

3.2.17 initial-value-job

An object that resides on a server, used for initializing the attributes of job objects and contained documents when a job is submitted by a client.

3.2.18 job copy

The output from each copy of a job in a job result set; a job result set has more than one job copy if the job's results-profile job-copies element is present and greater than 1; a job copy has more than one document set if the job contains more than one document object.

3.2.19 job result set

The output from a single value of the job's results-profile attribute; a job result set has more than one job copy if the job's job-copies element of the corresponding results-profile value is present and greater than 1.

3.2.20 logical medium

A DPA data object which (1) provides a generic description applicable to one or more *physical media* and/or (2) specifies medium properties by means of which a desired effect is to be achieved when a document is rendered.

3.2.21 logical printer

A data object 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 describe one or more printing devices having the same capabilities and characteristics; alternatively, a single printing device may be described 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. medium weights that may be (manually) loaded if the relevant logical printer is selected.

3.2.22 logical resource

An electronic representation of an object that may be referenced by, or used in combination with a printable document to render the printed output. Examples of logical resources include fonts and overlays such as electronic forms, logos and personal signatures. See *resource*, *physical resource*.

3.2.23 medium

The material or surface upon which a document is rendered in visible form by a presentation device; alternatively, a logical representation of such a material or surface. Examples include paper and the display screens of video display terminals. See *logical medium* and *physical medium*.

3.2.24 non-compulsory attribute

A (job or document) attribute is said to non-compulsory if a server either 1) shall perform the associated action if the server and printer support the attribute and its value(s) or 2) shall ignore (and possibly substitute for) any unsupported value(s).

3.2.25 object states

An object can be in one of a number of states. Except for the job and printer objects, an instance of a DPA object can be in one of the following states: *ready*, *on-request*, *unavailable*, *busy*, or *unknown*.

Note: An object is said to be available when it is in either the *ready* or *on-request* states.

3.2.26 on-request state

That state of an object instance indicating that it is able to be used, but only after human intervention has occurred.

3.2.27 output device

The hardware that renders documents; the output device is represented by the physical printer object in the print-system.

3.2.28 page

A page image or a *physical page*, depending on where the page is in the printing cycle. This general term is used where it is neither required nor desired to distinguish between the two more specific page forms.

3.2.29 page image

Image data that has been formatted for placement on a *physical page*; this term is generally applied to the electronic representation of a page.

3.2.30 page description language

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

3.2.31 physical medium

A data object that represents a particular physical instance or realization of a medium.

3.2.32 physical page

One side of a leaf of a book, document, etc. This term is distinguished from the term *physical medium* by the fact that the physical page bears an image that has been rendered upon it, or it is identified in some other manner as a product of the printing process.

3.2.33 physical printer

A data object that represents a particular output device which is capable of rendering documents.

3.2.34 physical resource

A physical object that may be used at some point in the production of a printed document. Examples of physical resources include physical media and physical devices such as printers and finishing equipment. See *resource*, *logical resource*.

3.2.35 presentation

The process of rendering or transforming an electronic document into visible form; for the purposes of ISO/IEC 10175, presentation usually means printing a document on paper or other tangible media, and may include finishing operations such as binding and punching and stapling, etc.; however presentation, in general, also includes other forms of rendering, such as display on video terminals, or transmission to remote facsimile devices, and may include output to microfilm and other devices.

3.2.36 print-access-protocol

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

3.2.37 print-client

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

3.2.38 print-job

A class of data object created by a print-server to collect information related to the progress of a particular print-request, or related sequence of print-requests. When a print-server accepts the initial (or only) print-request of a

sequence 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.

3.2.39 print operation management instructions (PMI)

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.

3.2.40 print operation parameters (POP)

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 ISO/IEC 10175. The set of print operation parameters are divided into two subsets, referred to as *print operation management instructions* and *document production instructions*, respectively.

3.2.41 print-request

An operation invoked by a print-client to cause a print-server to print a document or to initiate the printing of a set of documents. Parameters are sent with the print-request to identify the document, and to specify and control the act of printing and producing the document, 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 ISO/IEC 10175.

3.2.42 print-server

The entity that accepts requests for print-service in the form of the abstract-operations defined by ISO/IEC 10175; a generic term for the DP-Server defined formally in Clause 6. The fundamental type of service request is a print-request.

3.2.43 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.

3.2.44 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.

3.2.45 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.

3.2.46 privilege-attribute-certificate (PAC)

A certified set of access privileges that can be presented by a user or administrator to establish access rights.

3.2.47 ready state

That state of an object instance indicating that it is able to be used without human intervention.

3.2.48 Reference Coordinate System

A Cartesian coordinate system with units of millimetres on both axes which provides unambiguous specification of locations on the medium. See 6.5.4.

3.2.49 resource

An object or group of objects used in the printing of documents. See *logical resource* and *physical resource*.

3.2.50 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. See *document format*.

NOTE - There may be data formats that allow both revisable and final forms to be present at the same time in a single document (e.g., ISO/IEC 8613-1). In this case these terms refer to the particular aspect of information being present in a document.

3.2.51 service request

Within ISO/IEC 10175, this term refers to a request for any type of operation implemented by a print-server.

3.2.52 sheet

One instance of a medium, which may have either one or two imageable surfaces.

3.2.53 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 and slit along one fold to form an 8-page signature, and so on. The proper arrangement of pages in a signature is called imposition.

3.2.54 SPDL document

A 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.

3.2.55 supported

An attribute value or an object instance is said to be supported if it is the policy of an administrator of a server for the server to offer the attribute or object instance to clients. A supported attribute value or object instance may be available or temporarily unavailable.

3.2.56 unavailable state

That state of an object instance indicating that it is supported, but is not *currently* available (with or without human intervention).

3.2.57 user

Generic term applied to the initiator and/or recipient of some print-server action; may include *administrators* as well as other users with no special access rights or privileges.

4 Abbreviations

AFII	Association for Font Information Interchange
ARA	Assured reproduction area
ASCII	American Standards Code for Information Interchange

ASN.1	Abstract Syntax Notation One
CAP	Control Attribute Package
DCE	Distributed Computing Environment
DFR	Document Filing and Retrieval
DMTF	Desktop Management Task Force
DOAM	Distributed Office Applications Model
DoD	U.S. Department of Defense
DOR	Distinguished Object Reference
DPA	Document Printing Application
DPI	Document production instructions
DPS	Document Print Service
ENUM	Enumerated type
FIFO	First in - first out
FTAM	File Transfer and Access Method
IANA	Internet Assigned Number Authority
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
MIB	Management Information Base
MIF	Management Information Format
MOTIS	Message Oriented Text Interchange System
NATO	North Atlantic Treaty Organization
OID	Object Identifier
ONC	Open Network Computing
OSI	Open Systems Interconnection
PAC	Privilege Attribute Certificate
PCL	Printer Control Language
PDL	Page description language
PMI	Print operation management instructions
POP	Print operation parameters
POSIX	Portable Operating System Interface
QoS	Quality of Service

RCS	Reference Coordinate System
RDT	Referenced Data Transfer
RPC	Remote Procedure Call
SNMP	Simple Network Management Protocol
SPDL	Standard Page Description Language

5 Conventions

5.1 Conventions for abstract-services

This Part of ISO/IEC 10175 uses the following ASN.1-based descriptive conventions for the indicated purposes:

- ASN.1 itself, to specify the abstract-syntax of information-objects and their components, common data-types, and state-variables;
- The ASN.1 OBJECT and PORT macro and associated abstract-service definition conventions of ISO/IEC 10021-3, to specify the DPA Port;
- The ASN.1 ABSTRACT-BIND, ABSTRACT-UNBIND, ABSTRACT-OPERATION, and ABSTRACT-ERROR macros of ISO/IEC 10021-3, to specify the DPA abstract-service;
- The ASN.1 ATTRIBUTE MACRO and ATTRIBUTE SYNTAX MACRO from ISO/IEC 9594-2 to specify attributes and attribute syntaxes; the attribute concepts used in ISO/IEC 10175 are similar to those defined in ISO/IEC 9594-2, however, the Attribute and related datatypes have been modified to accommodate additional requirements of the Document Printing Application.

NOTE - ASN.1 specifications in ISO/IEC 10175 make use of the ISO/IEC 8824:1990 feature "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 the text of ISO/IEC 10175, the following conventions apply:

- Descriptive text is rendered in Times Roman, or a similar font;
- 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.

5.3 Conventions related to ASN.1

For the ASN.1-specified entities presented in ISO/IEC 10175, the following rules apply:

- ASN.1 syntax productions are rendered in Courier, or a similar monospaced font.
- ASN.1-specified names are rendered in bold text; these include type references, value references, identifiers, etc.
- ASN.1 type references begin with capital letters; a compound type reference consists of a sequence of components, each beginning with a capital letter, with the components concatenated together without intervening hyphens. Example: **PositiveInteger**.
- ASN.1 value references and identifiers begin with lower-case letters; a compound value reference or identifier consists of a sequence of components, each beginning with a lower-case letter, with the components separated by hyphens. Example: **upper-bound**.

- e) Attribute-names and attribute-identifiers are special cases of value-references (see 6.4).
 - 1) Attribute-names are constructed by the same rules as value-references (see 6.4). Example: **descriptive-name**.
 - 2) An attribute-identifier is the notation representation of the object identifier assigned to a particular attribute; an attribute-identifier consists of the string "**id-att-**", followed by the corresponding attribute-name. Example: **id-att-descriptive-name**.
- f) Attribute-syntax-names and attribute-syntax-identifiers are special cases of value-references (see 6.4).
 - 1) An attribute-syntax-name begins with a lower-case letter and immediately precedes the string "**Syntax**"; a compound attribute-syntax-name consists of a sequence of components concatenated together without intervening hyphens; each component after the first begins with a capital letter. Example: **descriptiveNameSyntax**.
 - 2) An attribute-syntax-identifier is the notation representation of the object identifier assigned to a particular attribute-syntax; an attribute-syntax-identifier consists of the string "**id-syn-**", followed by the corresponding attribute-syntax-name, altered to conform to the conventions for value references, above. Example: **id-syn-descriptive-name**.
- g) Object identifiers are used widely in ISO/IEC 10175 to represent values of attributes. The rules for construction and assignment of object identifier values used in this context are detailed in Annex A. References in descriptive text to particular object identifier values take one of two forms, depending on the context:
 - 1) object identifier names conform to the value reference rules in paragraph c, above. Example: **medium-type-envelope**.
 - 2) an object identifier-reference consists of the string "**id-val-**" followed by the corresponding object identifier-name. Example: **id-val-medium-type-envelope**.

Section 2 - DP abstract-service definition

6 DP abstract model

This clause provides an abstract functional model of the Document Printing Application (DPA) defined by this International Standard. For an introduction and description of the abstract-service concept and its definition conventions, see ISO/IEC 10021-3.

The Document Printing Abstract Model is illustrated in Figure 6-1. This model identifies two entities, a *DP-User* and a *DP-Administrator*, which interact with a third entity, a *DP-Server*, to provide a comprehensive facility for the printing of documents in a network environment.

The abstract-services defined by this International Standard, collectively referred to as DPA, enable 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*.

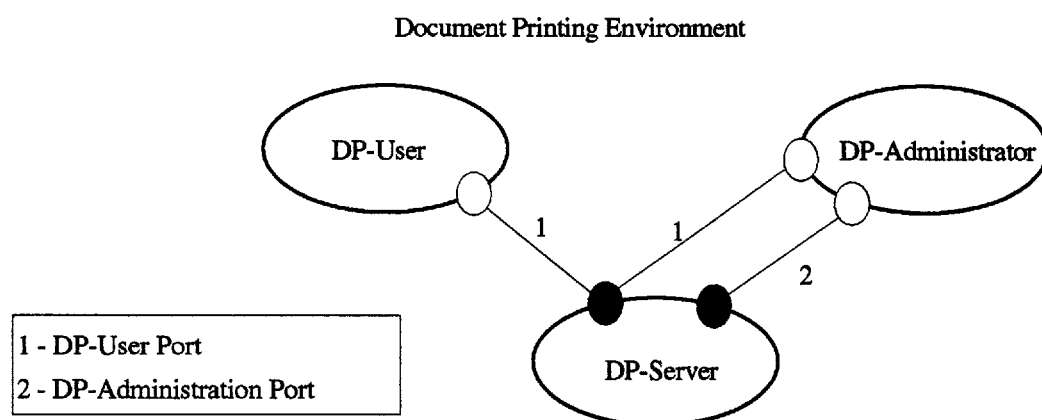


Figure 6-1 - Document Printing Abstract Model

A *DP-Server* is modeled as an object whose overall behavior 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 modeled as objects which obtain the services provided by a *DP-Server* through one or two port types, respectively. 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-Administration* port.

This means that a *DP-User* is limited to performing only those operations defined on the *DP-User* port, but a *DP-Administrator* can perform any operation available to a *DP-User*, in addition to those operations defined on the *DP-Administration* port. In the context of this International Standard, therefore, any discussion of “*DP-User*” features,

functions, or actions applies equally to the “DP-Administrator”, except in those specific cases where it is necessary to distinguish between them, and contrast the respective features of the two objects.

The main purpose for defining two ports, and two types of objects accessing those ports, is to provide a simple means for a server implementation to discriminate between normal users and administrative users in terms of the services offered. An administrative user, in general, is one who is responsible for normal operation and maintenance of a printing system, and who may therefore need to invoke operations that deal with the normal functioning of a print-server or printer. Normal users, in general, do not need to invoke those operations that are defined on the DP-Administration port.

NOTES

1 In practice, not all DP-Server implementations will implement the DP-Administration port operations, since these operations are not required in very simple DP-Servers. And those DP-Servers that do implement the operations of both ports will not necessarily distinguish between DP-Users and DP-Administrators. These decisions are the province of the particular implementation developers and site personnel, so long as an implementation conforms to one of the conformance classes defined in Annex E.

2 Within this International Standard, the generic term “user” is employed where there is no need to distinguish between the class of user: DP-User or DP-Administrator. In such cases it may be assumed that the user may be of either kind.

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 - 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.

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 in a visible form. 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 parts are exposed, and the operations may be supported by some form of communication between the objects.

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

The Document Print Server (DP-Server) makes its services available through the ports defined in this subclause. Two ports are defined in order to segregate the operations associated with normal printing activities, available to all users, from those operations intended for administrative control of a DP-Server.

Note that not all DP-Servers will implement the DP-Administration port, since simple servers will have no need of the administrative operations. In addition, a particular DP-Server implementation may, or may not, distinguish between DP-Users and DP-Administrators.

In conjunction with other International Standards, this International Standard provides security mechanisms to enable the accessors and accessees of these ports to determine each other's identities. However, this International Standard does not mandate the use of these security mechanisms, nor is any security policy established by this International Standard; these are the province of the implementors and using organizations.

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 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 ISO/IEC 10175.]

This port is formally defined as follows:

```
dp-administration-port PORT
  CONSUMER INVOKES {
    PromoteJob,
    InterruptJob,
    PauseJob,
    ResumeJob }
  ::= id-pt-dp-administration
```

6.3 DPA object classes

As described in the preceding subclauses, 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. In response, the DP-Server provides details of the status and outcome of an operation in the form of *results*.

To accomplish the action(s) requested via operation invocation, the DP-Server manages and manipulates 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 subdivided into the following classes:

- a) Job
- b) Document
- c) Printer
- d) Server
- e) Medium
- f) Font
- g) Resource
- h) Transfer-method
- i) Delivery-method
- j) Auxiliary-sheet package
- k) Auxiliary-sheet

- l) Finishing
- m) Output-method
- n) Imposition
- o) Scheduler
- p) Initial-value-job
- q) Initial-value-document
- r) Resource-context

The following subclauses 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.

Conforming servers are not required to implement support for all object classes. However, a server may support certain attributes of an object class even though it does not support the class itself. The values of some attributes of some objects, such as job, document, printer, and server, identify object instances of other classes, such as medium, auxiliary-sheet-package, finishing, and imposition. Servers may implement such attributes without implementing the object classes which these attribute values identify, as long as the semantics of such attributes is implemented for the specified object instance. For example, the **default-medium** document attribute identifies medium object instances, such as **iso-a4-white**. A server need not implement the medium object class in order to implement the **default-medium** document attribute, as long as the server prints the document on a white ISO A4 medium.

Refer to conformance requirements detailed in Annex E to this part of ISO/IEC 10175, and in part 2 of ISO/IEC 10175.

6.3.1 Job object class

When a user invokes a Print abstract-operation, the DP-Server (also 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 attributes of a job object convey and record information about the status, progress, and other characteristics of the job. In addition, a document to be printed is treated as a component of the job object, to be transferred transparently during the processing of the job.

Some print-servers and printers are capable of processing more than one document within a single print-job. Such *multiple-document jobs* require one Print abstract-operation invocation for each of the documents that comprise the job.

A job is created as a consequence of the first (or only) print-request of a related sequence of print-requests. An argument parameter of the print-request indicates whether or not it is the final print-request of the job, thus providing the delimiter for the job.

In addition to printable documents, other entities, called logical resources, may be transferred to a print-server during the course of a print-job. If such a resource is to be sent to the print-server with a print-job, the resource is conveyed by a single print-request that indicates that it is carrying a resource rather than a printable document. Examples of resources include electronic forms and overlays.

There are no restrictions on the sequencing of resource-bearing print-requests within the job, except that a resource must be present in the environment of the printer or server prior to being needed during the actual processing of the job (which normally begins after all print-requests have been submitted for the job).

6.3.2 Document object class

A document to be printed may be transmitted (immediately) within the sequence of parameters that compose a print-request, or it may be transferred at a later time by other means described subsequently.

In either case, each document in a print-job is represented in the print-server by a **document object**, which collects together summary information about the characteristics of the document and its status in the print cycle. The document object is created in response to the print-request that conveys the electronic document or provides the information by which the document is to be transferred subsequently. When a server creates a job object in response to a Print abstract-operation, each document object is contained within the job object.

6.3.3 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 fonts, 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** object represents some form of physical device capable of rendering graphical output on some physical medium; however, ISO/IEC 10175 makes no stipulation as to the actual physical characteristics embodied in a particular physical printer.

A **logical printer** object represents 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 be associated with several physical printers; conversely, a single physical printer may be associated with 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 abstract-operation. By directing a print request to a particular physical printer, the user is informing the associated print-server that a particular physical printer shall be used.

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.

A printer object may be specified to be either logical, physical, or both.

6.3.4 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.

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.

ISO/IEC 10175 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 utilise 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 neither addressed nor required by ISO/IEC 10175.

6.3.5 Medium object class

A **medium object** represents a medium on which a document (or part of a document) may be rendered. The most common example is paper, but also included are transparencies and virtually any physical material or device on which an image may be rendered.

Two different categories of media are addressed: *logical media* and *physical media*. A medium object may be specified to be logical, physical, or both. A **physical medium** is a medium that has distinct physical characteristics such as size, weight, type or pre-printed information. The final output of a printer will normally be on physical 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 physical media. A logical medium object may include a list of physical media from which the print-server is to select.

6.3.6 Font object class

As defined in ISO/IEC 9541, a font resource consists of a collection of glyph representations together with descriptive and font metric information relevant to the collection of glyph representations as a whole. A **font object** is a summary of information about one particular font resource that is available for use in rendering documents during the course of a print-job.

A **font object** is created and managed by a print-server in response to one of two situations:

- a) the font resource is submitted with one print-request of a print-job; the lifetime of such a font is limited to the job in which the font is submitted, and may be used in printing any of the documents that compose the job;
- b) the font resource is installed on a print-server or printer by some means outside the scope of this part of ISO/IEC 10175; such fonts may be persistent, i.e., not limited to the lifetime of any one job, but not necessarily permanent.

By means of operations 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 - the term font resource is used in a generic sense; that is, depending on implementation and other considerations, a font object may represent a font resource that does not conform to ISO/IEC 9541.

6.3.7 Resource object class

A **resource object** is similar to a font object, and is expected to represent some other logical resource such as an overlay or logo that may be used during the processing of a print-job.

As with font objects, a resource object is created and managed by a print-server in response to one of two situations:

- a) the resource object is submitted with one print-request of a print-job; the lifetime of such an object is limited to the job in which the object is submitted, and may be used in printing any of the documents that compose the job;
- b) the resource object is installed on a print-server or printer by some means outside the scope of this part of ISO/IEC 10175; such objects may be persistent, i.e., not limited to the lifetime of any one job, but not necessarily permanent.

6.3.8 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.

6.3.9 Delivery-method object class

A **delivery-method object** represents some particular means available to the print-server for delivering printed documents, 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.

6.3.10 Auxiliary-sheet-package object class

An **auxiliary-sheet-package object** represents a set of one or more auxiliary-sheets that are to be printed at positions relative to a job or relative to the documents in a job. For jobs, the auxiliary-sheets are to be printed before job components (job result sets and job copies), after components and/or between components of the same type. For documents, the auxiliary-sheets are to be printed before document components (document sets and document copies), after components and/or between components of the same type. A set of standard object identifiers representing various combinations of auxiliary-sheets is specified to facilitate interoperability.

6.3.11 Auxiliary-sheet object class

An **auxiliary-sheet object** represents a particular form of auxiliary-sheet that may be printed at the beginning of the requested print output, or at the end of the output, or between different parts of a print-job output to separate them. Different print-servers will have different numbers and formats of such things available, summarized by the auxiliary-sheet objects.

6.3.12 Finishing object class

Finishing encompasses the different kinds of processes that may be applied to printed output to produce completed documents. This includes such things as stapling (with various characteristics), binding, hole-punching or drilling, etc. Each specific kind of finishing available on a particular printer may be represented by a **finishing object**, which records the characteristics of the process, and provides an identifier by which it may be invoked.

A finishing object may be characterized and defined as either *logical* or *physical*, or both.

A **physical finishing object** defines the characteristics of a particular finishing process or mechanism that can be applied by one or more printers. A stapler installed on a printer is an example of such a physical mechanism; it would be represented in the server by a finishing object that details the features and parameters of the mechanism.

A **logical finishing object**, conversely, describes a finishing effect, but not necessarily the process or mechanism to be employed to achieve the effect. Thus, for example, a request for 3-hole punching of a document could be achieved by a punch or drill mechanism, or by substitution of a prepunched medium that otherwise satisfies the specified medium characteristics.

In addition, a logical finishing object can specify a set of finishing processes that are to be applied in sequence. For example, a single logical finishing object could be defined to cause an output document to be: (1) punched for eventual insertion into a 2-ring binder, (2) stapled along one edge, and (3) bound with tape. The entire set of finishing processes could then be invoked in a print-request by referencing the finishing object by means of its assigned identifier.

6.3.13 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.

6.3.14 Imposition 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 medium. The members of one group of such utility functions are termed convenience impositions.

The **imposition object** describes the placement of multiple logical page images onto a single physical page, such as 2-up or 4-up. In 2-up imposition, consecutive pairs of page images are selected from the composed document, and transformed such that the page images are rotated 90 degrees, reduced in size, and printed side-by-side, i.e., two page images on one surface of a physical sheet.

This International Standard defines several imposition objects, but a site might define special imposition objects with behavior that is different from the impositions performed by the standard objects or the standard numeric values of the **number-up** document attribute.

NOTE 1 - the **number-up** attribute value is either (1) a non-negative integer or (2) a simple-name or an OID that names an imposition object.

When an imposition object has been defined on a printer/server, an imposition function can be invoked in a print-request by supplying the identifier of the imposition object that represents it.

An imposition object describes an imposition effect, but not the method to be employed to achieve the effect. Further, the definition of an imposition object is independent of any physical details of the printer used to render the requested output.

NOTE 2 - the behavior of the imposition object depends on the values of other attributes.

6.3.15 Scheduler object class

Print servers may support different scheduling algorithms for selecting and processing print-jobs on different printers. The details of such algorithms are outside the scope of ISO/IEC 10175. Summary information about such algorithms, however, may be collected in a **scheduler object**, enabling administrators to create and select different algorithms, and users to inquire about those available and currently in effect.

6.3.16 Initial-value-job object class

Initial-value-job objects contain the attributes of the job class that the server may default when constructing or processing a job object. Initial-value-job objects may be referenced explicitly by a print-job or indirectly through a printer object.

The server shall maintain an initial-value-job object with the same set of attributes that the job object class supports and that may be specified on a print-request. There are, however, some exceptions and these attributes are listed in subclause 9.17.

6.3.17 Initial-value-document object class

Initial-value-document objects contain the attributes of the document class that the server may default when constructing or processing a document object. Initial-value-document objects may be referenced explicitly by a document or indirectly through a printer object.

The server shall maintain an initial-value-document object with the same set of attributes that the document object class supports and that may be specified on a print-request. There are, however, some exceptions and these attributes are listed in subclause 9.18.

6.3.18 Resource-context object class

Resource-context objects encapsulate details about the server's environment which, when used with a resource name, uniquely identify a resource in the server's environment. Certain information about a server's environment, for example the file system structure, need not be known by a client. This object encapsulates such details, allowing the

client to identify resources on a name only basis. It also allows administrators to change the location of resources on a system without requiring corresponding changes in the client.

Since resource-context objects reside in the domain of the server, a client may not be able to predetermine if a name for a resource-context is an explicit reference in the domain of the server, and hence should be represented as a Text name, or a resource-context object name as identified by the **name** CHOICE of the NameOrOid data type. Therefore, a conforming client shall pass the name of a resource-context object in the name field of the NameOrOid CHOICE of the ResourceContext data type. Upon receipt of an attribute with this data type, the server shall determine if this name is the name of a resource defined explicitly in the domain of the server (e.g. the filename of a resource). If so, the server shall recast the value of the attribute from the name field of the NameOrOid data type to the Text data type.

6.3.19 Derived classes and class inheritance for implementor extensions

In the object oriented (OO) concept of class inheritance, the specification of a derived class includes all of the attributes and operations of the base class, but adds additional attributes and/or operations that are relevant to the specialized derived class. In other words, the base class is more generic and the derived class is more specialized.

Each object of an object class, including all derived classes of the base class, shall have an unambiguous name or identifier. In other words, no two objects of the same object class or any of its derived classes shall have the same name or identifier.

ISO/IEC 10175-1 does not use object class inheritance in defining any of its standard object classes, for reasons of not introducing such a powerful tool when it is not really needed. However, implementors may wish to use the class inheritance approach to adding attributes to objects defined by ISO/IEC 10175, though they are not required to when adding implementor-defined attributes. See 8.2.4.4 for a complete example of implementor-defined derived classes for the printer class.

6.3.20 Events and event classes

Different classes of abstract-events are defined in ISO/IEC 10175 in a manner similar to that of object classes.

An event indicates that something has occurred during the processing of a print-job, or in the operation of a printer or server. 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). Every error detection is an event, but not all events involve errors.

Both **event types** and **event classes** are defined. The former are used to identify the specific event (e.g., the error event that signifies a specified deadline has passed) while the latter enable the user to identify the sets of events for which the server is to provide notifications (e.g., "log all error events"). See 8.5 for further details about event types and event classes.

Subclause 9.2.3 defines the job-event-handling attributes which are utilized to convey the event information. These attributes enable clients to indicate which events and event classes are to be logged for later analysis, and which are to cause notifications to be sent to the client immediately. ISO/IEC 10175 does not dictate any rules governing the use of these logs or notifications, however; these aspects are outside the scope of this International Standard.

6.4 Attributes

Each DPA-Object is represented by a set of attributes which characterize 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 identify the objects upon which to operate, to affect and modify the manner in which an operation is performed, to return details of the status and outcome of an operation, and to provide information about DPA-Objects.

In general, the principal reason for using attributes in an abstract-service definition is the ability to extend the facilities provided by a particular implementation, without the necessity of revising the governing standard. Each instance of an attribute includes both an identifier for that particular *attribute-type*, followed by a structure containing zero or more value elements appropriate to that attribute-type. Because of this consistent, self-identifying form, it is possible in DPA to define and integrate a new attribute-type into a print-server implementation in order to

provide access to some new facility, and for a print-client implementation to integrate and use the new attribute-type as soon as the new facility is available.

The attribute concept and the associated ASN.1 specifications originated in the OSI Directory Standard (ISO/IEC 9594-2). However, certain requirements of the Document Printing Application necessitated changes to the original definitions. For this reason, revised versions of these definitions appear, and are utilized in, ISO/IEC 10175.

6.4.1 The attribute-type

A DPA attribute consists of an **attribute-id** component, which identifies the specific type of **attribute**, and the corresponding **attribute-values** component, which encapsulates the information to be conveyed by the attribute. Figure 6-2 illustrates the attribute concept.

The DPA attribute is formally defined by the following ASN.1 production:

```
Attribute ::= SEQUENCE {
    attribute-id      [0] AttributeId,
    attribute-values  [1] SET OF ANY -- DEFINED BY attribute-id -- }
```

Particular DPA attributes are referred to as **attribute-types**. All attributes of a given object must be of distinct

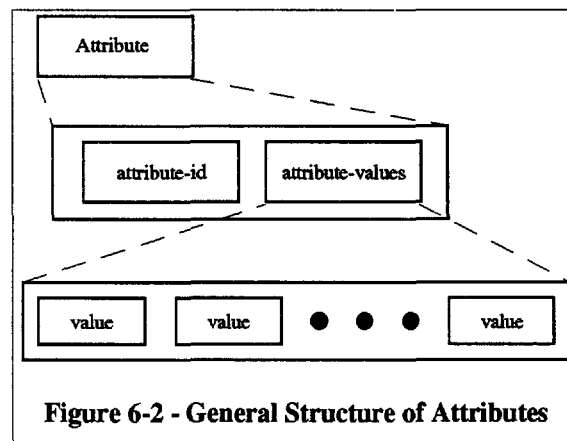


Figure 6-2 - General Structure of Attributes

attribute-types; i.e., for any particular object instance, no more than one instance of any particular attribute-type may exist at one time.

6.4.2 Attribute-id

Some attribute-types are standardized by ISO/IEC 10175. Other attribute-types will be defined by national administrative authorities and private organizations. Some externally defined attribute-types will not be specific to DPA such as, 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 attribute-type is defined.

```
AttributeId ::= OBJECT IDENTIFIER
```

Attribute-types defined by ISO/IEC 10175 are specified in clause 9. The procedures for constructing and registering object identifiers for other attribute-types may be found in ISO/IEC 8824; registration and dissemination of information regarding these attribute-types is the responsibility of the organizations that define them.

6.4.3 Attribute-values

Defining an attribute-type also involves specifying the ASN.1 datatype to which every value in such attributes must conform. As defined above, the datatype of the **attribute-values** component is the same for all attribute-types, i.e., SET OF ANY. However, encapsulated within each attribute-value are **value elements** whose types are specifically defined for each attribute-type, and identified by means of the associated attribute-id.

For some attribute-types, the **attribute-values** component may contain no more than a single element of the specified type. Such an attribute-type is said to be single-valued. For others, an **attribute-values** may (but need not) contain more than one component, all of the same ASN.1 datatype. 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 6.4.4.1).

6.4.4 Attribute-type definition

In summary, the definition of an attribute-type involves :

- assigning an object identifier to the **attribute-id**;
- indicating or defining the datatype of the associated **attribute-values** element;
- indicating whether an attribute of this attribute-type may have more than one value (i.e., whether the **attribute-values** element may contain more than one component of the specified type);
- indicating whether an attribute of this attribute-type may be used for filtering using equality, substring, and/or ordering or set-valued relations (see subclause 6.4.5).

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

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" |
                        "SET-COMPARISON" | "SET-INTERSECTION"
END

```

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

- the object identifier assigned to the **attribute-id** is the value supplied in the value assignment of the macro;

- b) the datatype of the **attribute-values** element(s) is that indicated by the AttributeSyntax production: a separately defined attribute-syntax identified by an object identifier and matching rules. 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 more than one value element if it is **MULTI VALUE** or empty.

NOTE - The attribute macro defined above is derived from ISO/IEC 9594-2, but has been revised to incorporate additional matching rules (see 6.4.5.1). Matching rules are discussed in 6.4.4.3.

6.4.4.2 Attribute-syntax definition

The definition of an attribute-syntax involves:

- a) assigning an object identifier to the attribute-syntax;
- b) indicating the datatype, in ASN.1, of the attribute-syntax;
- c) defining appropriate rules for matching an asserted value with a target attribute value. None, some, or all of the following matching rules may be defined for a particular attribute-syntax:
 - 1) equality. Applicable to any attribute-syntax. The asserted value must conform to the datatype of the attribute-syntax;
 - 2) substrings. Applicable to any attribute-syntax with a string datatype. The asserted value must be a sequence ('SEQUENCE OF'), each of whose elements conforms to the datatype;
 - 3) ordering. Applicable to any attribute-syntax for which a rule can be defined that will allow an asserted value to be described as less than, equal to, or greater than a target value. The asserted value must conform to the datatype of the attribute-syntax.
 - 4) set-comparison. Applicable to any attribute-syntax for which a rule can be defined that will allow an asserted value, or set of value elements, to be described as a subset or superset of the identified **attribute-values** set. The asserted value(s) must conform to the datatype of the attribute-syntax.
 - 5) set-intersection. Applicable to any attribute-syntax for which a rule can be defined that will allow an asserted value to be described as having one or more members in common with the identified **attribute-values** set. The asserted value(s) must conform to the datatype of the attribute-syntax.

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

- a) treats values of attributes of this attribute-syntax as having the type ANY, i.e., the DP-Server does not check that those values conform with the datatype indicated for the attribute-syntax;
- b) shall not attempt to match asserted values against target values of such an attribute-type.

If a matching rule is defined, the DP-Server:

- a) treats values of attributes of this attribute-syntax as having type ANY defined by the datatype indicated for the attribute-syntax;
- b) shall only match according to the matching rules defined for that attribute-syntax;
- c) shall only match an asserted value of a suitable datatype.

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" |
               "SET-COMPARISON" | "SET-INTERSECTION"

```

END

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

- the object identifier assigned to the attribute-syntax is the value supplied in the value assignment of the macro;
- the datatype of the attribute-syntax is that identified by the Syntax production, i.e., that following the macro name;
- the defined matching rules are **equality**, if "EQUALITY" appears in the MatchTypes production, **substrings** if "SUBSTRINGS" appears, **ordering** if "ORDERING" appears, **set-comparison** if SET-COMPARISON appears, and **set-intersection** if SET-INTERSECTION 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.

NOTES

- The attribute-syntax macro defined above is derived from ISO/IEC 9594-2, but has been revised to incorporate additional matching rules. (see 6.4.5.1)
- 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.

6.4.5 Filters

A filter parameter specifies a test that is either satisfied by a particular DPA-Object or not. The **Filter** is expressed in terms of assertions about the (1) presence or (2) values of certain attributes of the DPA-Object; the test is satisfied if and only if the **Filter** 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 **FilterItem**, or an expression involving simpler **Filters** combined by using the logical operators **and**, **or**, and **not**.

Where the **Filter** is:

- an **item**, it is true if and only if the corresponding **FilterItem** is true;
- an **and**, it is true unless any of the **Filters** in the SET are false;
NOTE - Thus, if there are no **Filters** in the SET, the **and** evaluates to true.
- an **or**, it is false unless any of the **Filters** in the SET are true;
NOTE - Thus, if there are no **Filters** in the SET, the **or** evaluates to false.
- a **not**, it is true if and only if the specified **Filter** is false.

6.4.5.1 Filter-item

A **FilterItem** is an assertion about the (1) presence or (2) value(s) of a particular attribute-type in the DPA-Object under test. Each such assertion is either true or false.

```

SubstringMatchCriteria ::= ENUMERATED {
    exact                (0),
    case-insensitive (1),
    same-letter          (2), -- ignoring accents, case, etc.
    approximate          (3) -- implementation-defined -- }

FilterItem ::= CHOICE {
    equality                [0] AttributeValueAssertion,
    substrings              [1] SEQUENCE {
        attribute-id        [0] AttributeId,
        match-criteria      [1] SubstringMatchCriteria,
        initial-string      [2] ANY OPTIONAL, -- DEFINED BY attribute-id
        any-string          [3] SEQUENCE OF ANY OPTIONAL,
                                -- DEFINED BY attribute-id
        final-string        [4] ANY OPTIONAL }, -- DEFINED BY attribute-id
    greater-or-equal       [2] AttributeValueAssertion,
        -- asserted value is greater than or equal to the attribute value
    less-or-equal          [3] AttributeValueAssertion,
        -- asserted value is less than or equal to the attribute value
    present                [4] AttributeId,
        -- asserted attribute is present (with any value)
    subset-of              [5] AttributeValueAssertion,
        -- asserted value is a subset of attribute value
    superset-of            [6] AttributeValueAssertion,
        -- asserted value is a superset of attribute value
    non-null-set-intersection [7] AttributeValueAssertion,
        -- at least one of the members of the asserted value is present
        -- in the attribute value -- }

```

Every **FilterItem** includes an **attribute-id** that identifies the particular attribute concerned.

Any assertion about the value of such an attribute is only evaluated if the **attribute-id** is recognized and defined for the DPA-Object, and the purported **attribute-values** conforms to the attribute-syntax defined for that attribute-type.

NOTE 1 - The **AttributeError invalid-attribute-syntax** is reported if the purported attribute-values does not conform to the attribute-syntax. No error is reported if the attribute-type is not defined for the object instance.

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 2 - The **AttributeError inappropriate-matching** is reported if the matching rule was not defined for that attribute-type.

Where the filter-item asserts:

- a) **equality**, it evaluates to TRUE if and only if the value supplied in the assertion is equal to the value of the attribute.

If the subject attribute is multi-valued, the assertion evaluates to TRUE if and only if the set of members supplied in the assertion is equal to the set of members in the attribute value.

- b) **substrings**, it evaluates to TRUE if and only if all of the substrings specified in the assertion appear in the attribute 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.

If **initial-string** is present, the substring shall match the initial substring of the attribute value; if **final-string** is present, the substring shall match the final substring of the attribute value; if **any-string** is present, each of the specified component substrings shall match a substring in the attribute value, in the order specified.

If the subject attribute is multi-valued, the assertion shall contain exactly one member, i.e., one sequence of substrings that will be matched separately to each member value of the subject attribute; the assertion evaluates to TRUE if and only if the supplied value evaluates to TRUE *for at least one* member of the attribute value, as defined above..

- c) **greater-or-equal**, it evaluates to TRUE if and only if the value supplied in the assertion is greater than or equal to the value of the attribute.

If the subject attribute is multi-valued, the assertion shall contain exactly one member; the assertion evaluates to TRUE if and only if the supplied value is greater than or equal to *at least one* member of the attribute value.

- d) **less-or-equal**, it evaluates to TRUE if and only if the value supplied in the assertion is less than or equal to the value of the attribute.

If the subject attribute is multi-valued, the assertion shall contain exactly one member; the assertion evaluates to TRUE if and only if the supplied value is less than or equal to *at least one* member of the attribute value.

- e) **present**, it evaluates to TRUE if and only if such an attribute is present in the object.

- f) **subset-of**, it evaluates to TRUE if and only if all of the assertion value is present in, or lies within, the attribute value.

This matching rule applies both to multi-valued attributes and to single-valued attributes that specify ranges or two-dimensional areas.

In the case of multi-valued attributes, the assertion evaluates to TRUE if and only if each of the assertion value members is present in the attribute value set.

For attributes that specify a range (or two-dimensional area), the assertion evaluates to TRUE if and only if the assertion range/area lies within the attribute value range/area. For examples, see 9.1.5.16, *IntegerRange* and *integerRangeSyntax*, and 9.1.5.47, *Area* and *areaSyntax*.

- g) **superset-of**, it evaluates to TRUE if and only if all of the attribute value is present in, or lies completely within, the assertion value.

This matching rule applies both to multi-valued attributes and to single-valued attributes that specify ranges or two-dimensional areas.

In the case of multi-valued attributes, the assertion evaluates to TRUE if and only if each of the attribute value members is present in the assertion value set.

For attributes that specify a range (or two-dimensional area), the assertion evaluates to TRUE if and only if the attribute range/area lies within the assertion value range/area. For examples, see 9.1.5.16, *IntegerRange* and *integerRangeSyntax*, and 9.1.5.47, *Area* and *areaSyntax*.

- h) **non-null-set-intersection**, it evaluates to TRUE if and only if at least one of the assertion members is present in the attribute value.

This matching rule applies only to multi-valued attributes.

NOTE 3 - An ordering rule must be stated for each different datatype used as the syntax of an attribute for which "MATCHES FOR ORDERING" is declared (i.e., an attribute which may be the object of greater-or-equal and less-or-equal attribute-value-assertions).

6.4.5.2 Attribute-value-assertion

An **AttributeValueAssertion** is a proposition concerning the values of a DPA-Object, which, in the context of that object instance and matching relation, may be true, false, or undefined. It involves an **attribute-id** and an **attribute-values**:

```
AttributeValueAssertion ::= SEQUENCE {
    attribute-id          [0] AttributeId,
    attribute-values      [1] SET OF ANY -- DEFINED BY attribute-id -- }
```

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 the specified matching relation (i.e., equality, substrings, ordering, set-comparison, or set-intersection);
 - 3) the **attribute-values** component of the assertion does not conform to the datatype defined for the **attribute-values** component of the specified attribute-type;
- b) **true**, if the DPA-Object contains an attribute of that attribute-type whose **attribute-values** component matches the **attribute-values** component in the assertion, according to the associated matching rule;
- c) **false**, otherwise.

6.4.6 Compulsory and non-compulsory attributes

Normally, a print-server is expected to honour a print-request explicitly. That is, a print-server will attempt to satisfy each and every parameter of the print-request exactly as submitted; unless instructed otherwise, a print-server shall reject any print-requests that cannot be honored completely, as specified.

However, the Document Printing Application assumes that there will exist many different print-server configurations and software implementations, 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.

Conversely, print-clients acting on behalf of human users will have different requirements. In some cases documents must be rendered exactly as specified; no substitutions can be allowed in font, choice of media, or finishing operations for example. But in many other cases the human-user wants only to print a set of documents (e.g., mail messages) in *some* form; the choice of font, media, etc., is of little importance, as long as readable documents are produced.

In order to deal consistently with this variability in user-requirements and available functionality, DPA provides print-clients with a means of indicating that certain print-request attributes are non-compulsory. Two attribute-types exist for this purpose: **job-non-compulsory-attributes** and **non-compulsory-attributes**, defined in subclauses 9.2.1.11 and 9.3.3.5, respectively. As indicated by their names, they enable clients to identify attributes of the job-object and the document-object that are to be treated as non-compulsory.

If an attribute-type is submitted in a print-argument and is *not* identified as non-compulsory:

- a) the server shall recognize the **attribute-id**, and shall understand the semantics of the attribute and its value;
- b) the requested attribute and value shall be supported (in the case of several compulsory attributes, all on the assigned printer(s)).

If either of the above conditions cannot be satisfied, the server shall return an **AttributeError** and reject the job. The **AttributeProblem** reported in each case shall be, respectively:

- a) undefined-attribute-type and