

**Information technology -
Document Printing Application (DPA) -
Part 3: Management abstract service definition and procedures**

**Draft Final Text
International Standard 10175-3**

Contents

Section 1 - General.....1

1 Scope.....1

2 Normative references.....1

3 Definitions.....2

4 Abbreviations2

5 Conventions.....2

Section 2 - DP management abstract service definition.....3

6 DP abstract model.....3

7 Abstract association information.....3

8 Abstract operations.....3

 8.1 Common datatypes and values used in DP management abstract operations 4

 8.1.1 Imported datatypes4

 8.1.2 Datatypes common for most DPA abstract operations5

 8.2 DP-User Port Abstract Operation Definitions 5

 8.3 DP-Administration Port Abstract-Operation Definitions 5

 8.3.1 Clean5

 8.3.2 Control.....8

 8.3.3 Create10

 8.3.4 Delete.....15

 8.3.5 Disable.....18

 8.3.6 Enable.....20

 8.3.7 Pause.....21

 8.3.8 Resubmit-job23

 8.3.9 Resume28

 8.3.10 Set.....29

 8.3.11 Shutdown35

 8.4 Abstract-Errors..... 38

 8.5 Abstract-Events 38

Annex A (normative) **Formal definition of the DPA management abstract service39**

Annex B (normative) **Conformance requirements46**

 B.1 Conformance level 3.....46

 B.2 Conformance level 3M.....46

1 Information technology

2 Document Printing Application (DPA)

3 Part 3:

4 Management abstract service definitions and procedures

5 Section 1 - General

6 1 Scope

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

11 The Document Printing Application Standard (ISO/IEC 10175) consists of three parts:

12 Part 1: Abstract service definitions and procedures

13 Part 2: Protocol specification

14 Part 3: Management abstract service definition and procedures

15 This part of ISO/IEC 10175 extends the facilities specified in ISO/IEC 10175-1 by adding the capability for
16 administrators and operators to manage and control printers, print servers and all other print objects; in addition, this
17 part specifies additional capabilities for a print client to control a print job. The text and specifications contained in
18 this part of ISO/IEC 10175 are derived from IEEE 1387.4 (POSIX) - System Administration Part 4 - Printing
19 Interfaces. In particular, this part of ISO/IEC 10175:

- 20 - specifies additional administrative functions and services that may be provided by Document Printing
21 Application servers;
- 22 - specifies the Document Printing Application abstract service for the additional administrative functions
23 using the principles established by the Abstract Service Definition Conventions (ISO/IEC 10021-3);
- 24 - specifies the requirements for conformance with the Document Printing Application for these additional
25 administrative functions.

26 See clause 1 of ISO/IEC 10175-1 for a more extensive description of the Document Printing Application.

27 2 Normative references

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

- 33 ISO/IEC 7498-1: 1994, *Information technology - Open Systems Interconnection -*
34 *Basic Reference Model: The Basic Model.*
- 35 ISO/IEC 7498-2: 1989, *Information processing systems - Open Systems Interconnection -*
36 *Basic Reference Model - Part 2: Security Architecture.*
- 37 ISO/IEC 8824: 1990, *Information technology - Open Systems Interconnection - Specification of Abstract*
38 *Syntax Notation One (ASN.1).*
- 39 ISO/IEC 8825: 1990, *Information technology - Open Systems Interconnection - Specification of Basic*
40 *Encoding Rules for Abstract Syntax Notation One (ASN.1).*
- 41 ISO/IEC 10021-3: 1990, *Information technology - Text Communication - Message-Oriented Text Interchange*
42 *Systems (MOTIS)*
43 *- Part 3: Abstract Service Definition Conventions.*
- 44 IEEE 1387.4 / D9, *Draft Standard for Information Technology - POSIX System Administration - Part 4:-*
45 *Printing Interfaces, IEEE, March 1998.*

46 **3 Definitions**

47 For the purposes of this part of ISO/IEC 10175, the definitions given in ISO/IEC 10175-1 apply.

48 **4 Abbreviations**

49 For the purposes of this part of ISO/IEC 10175, the abbreviations given in ISO/IEC 10175-1 apply.

50 **5 Conventions**

51 For the purposes of this part of ISO/IEC 10175, the conventions given in ISO/IEC 10175-1 apply.

52 Section 2 - DP management abstract service definition

53 6 DP abstract model

54 For the purposes of this part of ISO/IEC 10175, the DP abstract model given in ISO/IEC 10175-1, clause 6, applies.
55 The abstract operations defined in this part of ISO/IEC 10175 are all accessed through the DP-Administration port.

56 7 Abstract association information

57 For the purposes of this part of ISO/IEC 10175, the abstract association information given in ISO/IEC 10175-1,
58 clause 7, applies; no additional requirements exist for this part of ISO/IEC 10175.

59 8 Abstract operations

60 This clause defines the following abstract-operations:

- 61 a) Clean
- 62 b) Control
- 63 c) Create
- 64 d) Delete
- 65 e) Disable
- 66 f) Enable
- 67 g) Pause
- 68 h) ResubmitJob
- 69 i) Resume
- 70 j) Set
- 71 k) Shutdown

72 Normally only an appropriately authorized administrator may use the above operations. However, this is a site
73 and/or enterprise-specific policy decision, not mandated by this International Standard. Also, see the text of 8.3.8,
74 ResubmitJob, for a possible exception.

75 The table below lists each of the above abstract-operations and all of the abstract-operations of ISO-IEC 10175-1,
76 and indicates the object classes on which they operate.

abstract-operation	DPA Part	job	document	server	printer		all others
					logical	physical	
CancelJob	1	yes*					
Clean	3			yes		yes	
Control	3			yes		yes	
Create	3			unspecified	yes	yes	yes
Delete	3	yes	yes	unspecified	yes	yes	yes
Disable	3			yes	yes	yes	
Enable	3			yes	yes	yes	
InterruptJob	1	yes*					
ListObjectAttributes	1	yes	yes	yes	yes	yes	yes

ModifyJob	1	yes*					
Pause	3			yes		yes	
PauseJob	1	yes*					
Print	1	yes*					
PromoteJob	1	yes*					
ResubmitJob	3	yes		yes	yes	impl. option	
Resume	3			yes		yes	
ResumeJob	1	yes*					
Set	3	yes	yes	yes	yes	yes	yes
Shutdown	3			yes		impl. option	

77 NOTES

78 1 * Most abstract-operations utilize an **object-class** argument element to specify the class of the **object-**
 79 **instance** (or **object-identification**). However, many of the operations of ISO/IEC 10175-1 do not since they only
 80 operate on a single class, the job. Operations that do not utilize an **object-class** argument element are marked with an
 81 asterisk (*) in the above table.

82 2 In POSIX, multiple object instances may be specified, either as an explicit list of individual object instances, as an
 83 object instance that may contain wildcard characters, e.g. an asterisk (*), or as a combination of both methods.
 84 However, DPA abstract-operations may only be applied to a single specific **object-instance** at a time. Thus the print
 85 client must invoke a separate DPA operation for each **object-instance**. But if the use of wildcard characters is
 86 supported by the print client and print server, the print client may invoke the specified abstract-operation on all objects
 87 of the specified **object-class** whose name matches the wildcard expansion. Support for wildcards is an implementation
 88 option.

89 3 The **Clean, Shutdown and Print** operations may operate asynchronously; i.e., these operations need not complete
 90 immediately before returning to the client, based on the state of the relevant object. All other DPA abstract-operations
 91 operate synchronously.

92 **8.1 Common datatypes and values used in DP management abstract operations**

93 **8.1.1 Imported datatypes**

94 This part of ISO/IEC 10175 imports and uses the following datatypes defined in ISO/IEC 10175-1:

95 From ISO/IEC 10175-1.Annex B:

- 96 Attribute,
- 97 AttributeId,
- 98 CommonArguments,
- 99 JobAttrModification,
- 100 ModifyOperator
- 101 ObjectIdentification,

- 102 AccessError,
- 103 AttributeError,
- 104 PrinterError,
- 105 SecurityError,
- 106 SelectionError,
- 107 ServiceError,
- 108 UpdateError

109 From ISO/IEC 10175-1.Annex C:

- 110 DistinguishedNameString.
- 111 JobIdentifier,
- 112 Message

113 In addition, the abstract-operations defined in this part of ISO/IEC 10175 employ the abstract service macros
114 ABSTRACT-ERROR and ABSTRACT-OPERATION, defined in ISO/IEC 10021-3.

115 8.1.2 Datatypes common for most DPA abstract operations

116 8.1.2.1 Common-arguments

117 The argument of each of the abstract-operations defined in this part of ISO/IEC 10175 includes an optional element
118 of type **CommonArguments**, imported from ISO/IEC 10175-1 Annex C, as indicated in 8.1.1, above. See 8.1.2 in
119 ISO/IEC 10175-1 for specifications of the attributes that may be included in an instance of the **CommonArguments**
120 type.

121 8.1.2.2 Message

122 The argument of each of the abstract-operations defined in this part of ISO/IEC 10175 includes an optional element
123 of type **Message**, imported from ISO/IEC 10175-1 Annex C, as indicated in 8.1.1, above. When a client specifies a
124 **Message** element in one of the abstract-operations defined in this part of ISO/IEC 10175, the server shall treat it in
125 the manner specified in this subclause.

126 The server shall attach the specified message to the **object-instance** identified in the operation. The message string
127 shall be a human-readable message which can be retrieved later by using the **ListObjectAttributes** operation.

128 If messages are specified in one locale and retrieved in another, a variety of results may occur depending on the
129 message specification from the client locale and at the server locale. See the table below for different possible results
130 to **message** at the server. See 8.1.2.3 in ISO/IEC 10175-1 for a description of **operation-locale**.

		if message at the server locale is:		
		unspecified	non-NULL string	NULL string
if message from the client locale is:				
unspecified		message at server remains unspecified	message at server remains the non-NULL string	message at server remains the NULL string
non-NULL string		message at server becomes the client non-NULL string	message at server changes to the client non-NULL string	message at server changes to the client non-NULL string
NULL string		message at server becomes the NULL string	message at server changes to the NULL string	message at server remains the NULL string

131 8.2 DP-User Port Abstract Operation Definitions

132 There are no DP-User Port abstract-operations defined in this part of ISO/IEC 10175.

133 8.3 DP-Administration Port Abstract-Operation Definitions

134 Clients may use the abstract-operations described in the following subclauses to administer and control the flow of
135 print-jobs through printers, and to administer and control print objects. Specific print server implementations may
136 limit usage of one or more of these abstract-operations to administrators with sufficient access rights, and/or may
137 vary the facilities provided by a given abstract-operation depending upon the access rights assigned to the individual
138 administrators served by the print server.

139 8.3.1 Clean

140 This abstract-operation allows an administrator to remove all jobs from the specified print server or printer.
141 (Removal of jobs from logical printers is an implementation option.) When the Clean operation is invoked by an
142 administrator with appropriate privileges, the server shall:

- 143 a) If the object is enabled, reject the abstract-operation and report an **inappropriate-object-state**
- 144 **AccessError**; otherwise,
- 145 b) If the object is a physical printer, pause the printer to stop the printer from producing output (see **Pause**
- 146 **operation**),
- 147 c) Remove all jobs contained in or scheduled for that object,
- 148 d) Restore the state of the object to what it was when the execution of the **Clean** operation began.

149 The server shall remove jobs that are being printed on the physical printer at the time the **Clean** operation is invoked
 150 in a manner that is consistent with the behavior of the **Delete** operation.

151 This operation may operate asynchronously; i.e., this operation need not complete immediately before returning,
 152 based on the state of the object.

153 This abstract-operation is formally defined as follows:

```

154 Clean ::= ABSTRACT-OPERATION
155     ARGUMENT      CleanArgument
156     RESULT       CleanResult
157     ERRORS      { AccessError,
158                   SecurityError,
159                   ServiceError }
```

160 NOTE - Normally only an appropriately authorized administrator may clean a printer or server object. However, this is
 161 a site and/or enterprise-specific policy decision, not mandated by this International Standard.

162 **8.3.1.1 Clean-argument**

163 The argument of this abstract-operation is formally defined as follows:

```

164 CleanArgument ::= SEQUENCE {
165     object-class      [0] OBJECT IDENTIFIER,
166                       -- id-oc-server, id-oc-printer
167     object-instance [1] ObjectIdentification,
168                       -- choice depends on object class
169     message          [2] Message OPTIONAL,
170     common-arguments [3] CommonArguments OPTIONAL }
```

171 The following argument elements shall be supported by the server:

172 **object-class**

173 This argument element shall specify the object class of the **object-instance**. The **object-class** argument element
 174 shall be one of the following values: **id-oc-printer** or **id-oc-server**.

175 **object-instance**

176 This argument element shall specify the **object-instance** based on the value of the **object-class** argument element.
 177 The administrator shall specify one **object-instance** value associated with the **object-class**.

178 For each supported **object-class**, the server shall remove all jobs from the object specified by the **object-instance**
 179 identified by the **ObjectIdentification** element.

180 A print server shall either accept or reject a request to clean an **object-instance**. The print server shall return any
 181 attribute warnings or abstract-errors to the print client.

182 Each **object-instance** shall take one of the following forms:

object-class Specification	ObjectIdentification element	object-instance Values
-----------------------------------	-------------------------------------	-------------------------------

id-oc-printer	simple-name	<p>Printer name representing a physical printer.</p> <p>The behavior of the Clean operation on a job that is currently printing shall be consistent with the behavior of the Delete operation on currently printing jobs. See 8.3.4 for a description of the Delete operation.</p> <p>Use of the Clean operation for a logical printer is not supported.</p> <p>NOTE 1 - As an example, if the printer is a physical printer, a server implementation shall delete all jobs that are currently printing on the specified physical printer, i.e. jobs that have their printers-assigned attribute set to the physical printer. If the printer is a logical printer, the server implementation shall reject the operation and return a clean-logical-printer-unsupported ServiceError.</p>
id-oc-server	simple-name	<p>Server name.</p> <p>Cleaning a print server shall cause all jobs to be removed (deleted) that have been submitted to it (to any of its supported printers). Therefore, all print jobs on all printers supported by the specified server shall be removed. The behavior of the Clean operation on a job that is currently printing shall be consistent with the behavior of the Delete operation on currently printing jobs. See 8.3.4 for a description of the Delete operation.</p> <p>NOTE 2 - As an example, a server implementation shall delete all jobs that it contains, including jobs that are currently being printed on a physical printer.</p>

183 **message**

184 See 8.1.2.

185 **common-arguments**

186 See 8.1.2 in ISO/IEC 10175-1 for a description of the **CommonArguments** type.

187 8.3.1.2 Clean-result

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

```
189     CleanResult ::= SEQUENCE {
190         object-status [0] SET OF Attribute OPTIONAL }
```

191 The server may return object-status attributes of the cleaned object.

192 8.3.1.3 Clean Abstract-errors

193 If the request fails, the server shall return one of the listed abstract-errors. The circumstances under which the server will return a particular abstract-error are defined in 8.4 of ISO/IEC 10175-1.

195 **8.3.2 Control**

196 This abstract-operation allows an administrator to perform a variety of operations on the specified print server or
 197 printer. The **operations** argument element contains an attribute for each operation to be performed by the **Control**
 198 operation. The operations that are currently supported by the **Control** operation are reset and tracing.

199 This abstract-operation is formally defined as follows:

```

200 Control ::= ABSTRACT-OPERATION
201     ARGUMENT      ControlArgument
202     RESULT        ControlResult
203     ERRORS        { AccessError,
204                   SecurityError,
205                   ServiceError }
    
```

206 NOTE - Normally only an appropriately authorized administrator may invoke the **Control** operation. However, this is
 207 a site and/or enterprise-specific policy decision, not mandated by this International Standard.

208 **8.3.2.1 Control-argument**

209 The argument of this abstract-operation is formally defined as follows:

```

210 ControlArgument ::= SEQUENCE {
211     object-class      [0] OBJECT IDENTIFIER,
212                     -- id-oc-server, id-oc-printer
213     object-instance  [1] ObjectIdentification,
214                     -- choice depends on object class
215     message          [2] Message OPTIONAL,
216     operations       [3] SEQUENCE OF Attribute,
217                     -- operands and parameters
218                     -- Current operations: reset and trace
219     common-arguments [4] CommonArguments OPTIONAL }
    
```

220 The following argument elements shall be supported by the server:

221 **object-class**

222 This argument element shall specify the object class of the **object-instance**. The **object-class** argument element
 223 shall be one of the following values: **id-oc-printer** or **id-oc-server**.

224 **object-instance**

225 This argument element shall specify the **object-instance** based on the value of the **object-class** argument element.
 226 The administrator shall specify one **object-instance** value associated with the **object-class**.

227 For each supported **object-class**, the server shall perform the **Control** operation defined by the **operations** argument
 228 element on the object specified by the **object-instance** identified by the **ObjectIdentification** element.

229 A print server shall either accept or reject a **Control** operation request on the **object-instance**. The print server shall
 230 return any attribute warnings or abstract-errors to the print client.

231 Each **object-instance** shall take one of the following forms:

object-class Specification	ObjectIdentification element	object-instance Values
id-oc-printer	simple-name	Printer name representing a physical printer. The behavior of the Control operation depends on the operations argument element, which may be one or more of the following attributes:

		<ul style="list-style-type: none"> • reset: The server shall reset the physical printer according to the value of the reset attribute. The server may perform either a cold start (id-val-reset-power-cycle), a warm start (id-val-reset-to-nvram), or a reset to factory default conditions (id-val-reset-to-factory-default). Support for id-val-reset-to-factory-default is an implementation option. • trace-enable: The server shall enable tracing for the physical printer if the trace-enable attribute is TRUE, and shall disable tracing if the or trace-enable attribute is FALSE. <p>Use of the Control operation for a logical printer is not supported.</p>
id-oc-server	simple-name	<p>Server name.</p> <p>The behavior of the Control operation depends on the operations argument element, which may be one or more of the following attributes:</p> <ul style="list-style-type: none"> • reset: The server shall reset itself according to the value of the reset attribute. The server may perform either a cold start (id-val-reset-power-cycle), a warm start (id-val-reset-to-nvram), or a reset to factory default conditions (id-val-reset-to-factory-default). Support for id-val-reset-to-factory-default is an implementation option. • trace-enable: The server shall enable tracing for itself if the trace-enable attribute is TRUE, and shall disable tracing if the or trace-enable attribute is FALSE.

232 **message**

233 See 8.1.2.

234 **operations**

235 This argument element shall specify the attributes and values that describe the specific **Control** operation to be
 236 applied to the **object-instance**. The operations that are currently supported by the **Control** operation are described
 237 by the following attributes:

- 238 • **reset:** This operation resets the printer or print server at one of the following levels:
 239 **id-val-reset-power-cycle** for a cold start, **id-val-reset-to-nvram** for a warm start, or
 240 **id-val-reset-to-factory-default** for a reset to factory default conditions.

241 NOTE - For jobs that are being printed on the physical printer at the time the **Control** operation is invoked
 242 with the **reset** attribute, the results are unspecified.

- 243 • **trace-enable:** This operation enables tracing when the value of **trace-enable** is **TRUE** and
 244 disables tracing when the value of **trace-enable** is **FALSE**. When tracing is
 245 enabled, the printer or print server logs its execution activity to a file.

246 **common-arguments**

247 See 8.1.2 in ISO/IEC 10175-1 for a description of the **CommonArguments** type.

248 **8.3.2.2 Control-result**

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

```
250     ControlResult ::= SEQUENCE {
251         object-status [0] SET OF Attribute OPTIONAL }
```

252 The server may return object-status attributes of the cleaned object.

253 **8.3.2.3 Control Abstract-errors**254 If the request fails, the server shall return one of the listed abstract-errors. The circumstances under which the server
255 will return a particular abstract-error are defined in 8.4 of ISO/IEC 10175-1.256 **8.3.3 Create**257 This abstract-operation allows an administrator to create a printing object and set its attributes to the values specified
258 within the argument elements. The **object-class** argument element shall indicate the class of the object. The server
259 shall create the specified object instance as an object of the specified **object-class**. When the Create operation is
260 invoked by an administrator with appropriate privileges, the server shall perform the following actions:

- 261 a) The server shall initialize the newly created object with the attributes specified by the **object-attributes**
262 argument element. This has the same effect as issuing an equivalent **Set** operation. Note however, that
263 some attributes may be settable with the **Create** operation, but may be restricted from further
264 modification with the **Set** operation. See the **Set** operation description and the restrictions on attribute
265 modification in 8.3.10. If the **copy-from** argument element is specified, the server creates a copy of the
266 object specified by the **copy-from** argument element, and the attributes specified by the **object-**
267 **attributes** argument element are applied subsequently.
- 268 b) If the specified object already exists and the **force-creation** argument element is set to **FALSE**, the
269 print server shall return the **object-already-exists SelectionError** for this operation. If the **force-**
270 **creation** argument element is set to **TRUE**, the print server shall immediately delete the existing object
271 and shall create the new object. If the printer is not able to immediately complete the deletion of the
272 existing object due to the state of the printer, the server shall rename the existing object so that the
273 creation of the new object can complete.
- 274 c) An **object-instance** created by the **Create** operation shall be permanent; the object is persistent, lasting
275 across re-initialization of the print system, or any of its components.
- 276 d) When an **object-instance** of class **id-oc-printer** is created, the server shall place it in the **idle** state with
277 the **enabled** attribute set to **FALSE** so print jobs cannot be accepted by the new printer. The
278 administrator may then invoke the **Enable** operation to place the printer object in the **idle** state with the
279 **enabled** attribute set to **TRUE**.
- 280 e) The server shall create an entry for the newly created **object-instance** in the name service, if such an
281 entry is required by the server implementation.

282 This argument element shall cause the newly created **object-instance** to be a copy of the object specified by the **copy-**
283 **from** argument element. The **Create** operation shall not operate on jobs or documents. It is implementation-defined
284 whether the **Create** operation can cause the server to create print servers and how this is accomplished. Print servers
285 may also be created by methods outside the scope of this International Standard. The **Create** operation shall operate
286 on all other supported object classes.

287 This abstract-operation is formally defined as follows:

```

288 Create ::= ABSTRACT-OPERATION
289     ARGUMENT      CreateArgument
290     RESULT        CreateResult
291     ERRORS        { AccessError,
292                   SecurityError,
293                   SelectionError,
294                   ServiceError }

```

295 NOTE - Normally only an appropriately authorized administrator may create an object. However, this is a site and/or
 296 enterprise-specific policy decision, not mandated by this International Standard.

297 8.3.3.1 Create-argument

298 The argument of this abstract-operation is formally defined as follows:

```

299 CreateArgument ::= SEQUENCE {
300     object-class      [0] OBJECT IDENTIFIER, -- id-oc-xxx
301                   -- any id-oc-xxx, excluding job and document
302     object-instance  [1] ObjectIdentification,
303                   -- instance identifier depends on class
304     message          [2] Message OPTIONAL,
305     copy-from        [3] ObjectIdentification OPTIONAL,
306                   -- initial value object (IVO)
307     force-creation   [4] BOOLEAN DEFAULT FALSE,
308                   -- recreate if necessary
309     object-attributes [5] SET OF Attribute OPTIONAL,
310                   -- attributes of new object instance
311     common-arguments [6] CommonArguments OPTIONAL }

```

312 The following argument elements shall be supported by the server:

313 **object-class**

314 This argument element shall specify the object class of the **object-instance** to be created. The server implementation
 315 shall support a specific conformance level, as described in Annex E of ISO/IEC 10175-1, and that conformance level
 316 will determine the object classes and the attributes that the server supports, as a minimum.

317 **object-instance**

318 This argument element shall specify the **object-instance** based on the value of the **object-class** argument element.
 319 The administrator shall specify one **object-instance** value associated with the **object-class**.

320 If the specified **object-class** is supported, the server shall create the object specified by the **object-instance** identified
 321 by the **ObjectIdentification** element.

322 A print server shall either accept or reject a request to create an **object-instance**. The print server shall return any
 323 attribute warnings or abstract-errors to the print client.

324 Each **object-instance** shall take one of the following forms:

object-class Specification	ObjectIdentification element	object-instance Values
id-oc-printer	simple-name	Printer name representing a logical or physical printer. If a printer object is created with the same name as a previously existing printer object which had jobs submitted to it, the new printer object shall also have those jobs submitted to it. Since the printer is created in the idle state with the enabled attribute set to FALSE , when the user issues a Enable operation on the newly (re)created printer, that printer shall begin printing the jobs that were submitted to it.
id-oc-server	simple-name	Server name. The server shall create a new server object in an unspecified manner. The ability to create a server object is implementation-defined.
id-oc-initial-value-document	name-or-oid	Initial-value-document name. Contains the attributes of the document class that the server may default when constructing a document object.
id-oc-initial-value-job	name-or-oid	Initial-value-job name. Contains the attributes of the job class that the server may default when constructing a job object.
id-oc-auxiliary-sheet	name-or-oid	Start sheet, end sheet, separator sheet name.
id-oc-auxiliary-sheet-package	name-or-oid	Predefined package of start sheets, end sheets, and separator sheets referenced by a single name.
id-oc-delivery-method	object-identifier	Delivery method identifier; such as for print job output.
id-oc-finishing	name-or-oid	Finishing method name; such as stapling, folding, or binding.
id-oc-font	font-reference	Font name.
id-oc-imposition	name-or-oid	Imposition method name; such as 2-up or 4-up.
id-oc-medium	name-or-oid	Media name; such as iso-a4-white or north-american-letter.
id-oc-output	object-identifier	Output method identifier; such as sorting or collating.
id-oc-resource	name-or-oid	Resource identifier; such as overlay or logo.
id-oc-resource-context	name-or-oid	Resource context name; encapsulate details about the server's environment that, when used with a resource name, uniquely identifies a resource in the environment of that server.
id-oc-scheduler	object-identifier	Scheduler name; determines the order in which jobs are sent to physical printer.
id-oc-transfer-method	object-identifier	Transfer method identifier; such as ftam , with-request , or push-by-client .

325 **message**

326 See 8.1.2.

327 **copy-from**

328 This argument element shall cause the newly created **object-instance** to be a copy of the object specified by the **copy-**
 329 **from** argument element. Attributes contained in that object shall be copied into the newly created object. The object
 330 may be located on any accessible print server.

331 The syntax and interpretation of the specified value is dependent on the **object-class** argument element in the same
 332 manner as **object-instance**.

333 If the optional server-name component of the name is omitted, the server shall interpret the value of the **copy-from**
 334 argument element as the name of an object on the same server as the object being created. If the optional server-
 335 name component of the name is included, the server shall request a list of the attributes of the *copy-from-object-*
 336 *instance* from the specified server and initialize the newly created object with those attributes. The ability of a server
 337 to copy an object from another server is an implementation option. A conforming server implementation not
 338 supporting this feature shall return an **unsupported-parameter-value ServiceError** when the administrator
 339 specifies a **server-name** that is not identical to the name of the server on which the object is being created.

340 **force-creation**

341 This argument element specifies whether or not the server shall be allowed to create a new print object when an
 342 object with the specified name already exists.

343 The **force-creation** argument element shall be one of the following values: **TRUE** or **FALSE**.

344 When an administrator with appropriate privileges specifies the **force-creation** argument element with value **TRUE**,
 345 the server shall create the specified print object, whether or not an object with the specified name already exists. If a
 346 print object with the specified name already exists, the server shall delete the existing print object before it creates the
 347 new print object. The server shall return no abstract-error or warning in this case. If the specified object does not
 348 exist, this argument element shall have no effect.

349 When an administrator with appropriate privileges specifies the **force-creation** argument element with value
 350 **FALSE**, the server shall create the specified print object only if an object with the specified name does not already
 351 exist. If a print object with the specified name does not already exist, the server shall create the new print. If a print
 352 object with the specified name does already exist, the server shall not create the new print object, but shall return the
 353 **object-already-exists SelectionError**.

354 **object-attributes**

355 This argument element shall specify the attributes and values with which the new **object-instance** shall be
 356 initialized. If the **copy-from** argument element is specified, the attributes specified by the **object-attributes**
 357 argument element are applied subsequently.

358 **common-arguments**

359 See 8.1.2 in ISO/IEC 10175-1 for a description of the **CommonArguments** type.

360 **8.3.3.2 Create-result**

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

```
362 CreateResult ::= SEQUENCE {
363     object-status [0] SET OF Attribute }
```

364 The server shall return the supported subset of the following **object-status** attributes of the created object depending
 365 on the **object-class**:

Object	object-class Specification	Attributes
Printer	id-oc-printer	id-att-printer-name id-att-printer-realization id-att-printer-state id-att-availability id-att-associated-server

		id-att-printer-locations id-att-printer-associated-printers id-att-scheduler-ready
Server	id-oc-server	id-att-server-name id-att-server-state id-att-availability id-att-logical-printers-supported id-att-physical-printers-supported
Medium	id-oc-medium	id-att-medium-identifier id-att-associated-server id-att-medium-realization id-att-medium-type id-att-medium-associated-media id-att-medium-size id-att-medium-dimensions id-att-medium-sides id-att-medium-color
Imposition	id-oc-imposition	id-att-imposition-identifier id-att-associated-server id-att-imposition-specification id-att-number-up
Initial-value-job	id-oc-initial-value-job	id-att-initial-value-job-identifier id-att-associated-server id-att-logical-printers-ready id-att-physical-printers-ready id-att-printer-locations-requested id-att-printer-models-requested id-att-job-retention-period
Initial-value-document	id-oc-initial-value-document	id-att-initial-value-document-identifier id-att-associated-server id-att-logical-printers-ready id-att-physical-printers-ready id-att-copy-count id-att-sides id-att-document-format
Font	id-oc-font	id-att-font-identifier id-att-associated-server id-att-file-reference
Auxiliary-sheet	id-oc-auxiliary-sheet	id-att-auxiliary-sheet-identifier id-att-auxiliary-sheet-medium id-att-associated-server
Auxiliary-sheet-package	id-oc-auxiliary-sheet-package	id-att-auxiliary-sheet-package-identifier id-att-associated-server
Transfer-method	id-oc-transfer-method	id-att-transfer-method-identifier id-att-transfer-method-type id-att-document-reference-type id-att-associated-server
Resource	id-oc-resource	id-att-resource-name id-att-resource-type id-att-associated-server id-att-file-reference
Resource-context	id-oc-resource-context	id-att-resource-context-identifier

		id-att-context-address id-att-associated-server
Delivery-method	id-oc-delivery-method	id-att-delivery-method-identifier id-att-associated-server
Finishing	id-oc-finishing	id-att-finishing-identification id-att-finishing-realization id-att-associated-server id-att-finishing-specification id-att-finishing-associated-finishings
Output	id-oc-output	id-att-output-identifier id-att-associated-server
Scheduler	id-oc-scheduler	id-att-scheduler-identifier id-att-associated-server

366 8.3.3.3 Create Abstract-errors

367 If the request fails, the server shall return one of the listed abstract-errors. The circumstances under which the server
368 will return a particular abstract-error are defined in 8.4 of ISO/IEC 10175-1.

369 8.3.4 Delete

370 This abstract-operation allows an administrator to delete printing objects. The **object-class** argument element shall
371 indicate the class of the object. When the Delete operation is invoked by an administrator with appropriate
372 privileges, the server shall perform the following actions:

- 373 a) If the specified object is a printer the server shall first determine whether the printer has been disabled
374 and is without jobs: if the printer is not in this required state, the server shall reject the abstract-
375 operation and report an **inappropriate-object-state** AccessError; otherwise:
- 376 b) The server shall delete the specified print object(s) from the print system;
- 377 c) The server shall delete the print object entry in the name service, if an entry exists and is required by
378 the server implementation;
- 379 d) The server shall delete a currently printing job, as soon as the printer technology allows it to be deleted,
380 in an implementation-defined manner.

381 It is implementation-defined whether the **Delete** operation can delete print servers and how this is accomplished.

382 This abstract-operation is formally defined as follows:

```

383 Delete ::= ABSTRACT-OPERATION
384     ARGUMENT      DeleteArgument
385     RESULT        DeleteResult
386     ERRORS       { AccessError,
387                     SecurityError,
388                     ServiceError,
389                     UpdateError }

```

390 NOTE - Normally only an appropriately authorized administrator may delete an object. However, this is a site and/or
391 enterprise-specific policy decision, not mandated by this International Standard.

392 8.3.4.1 Delete-argument

393 The argument of this abstract-operation is formally defined as follows:

```

394 DeleteArgument ::= SEQUENCE {
395     object-class      [0] OBJECT IDENTIFIER,
396                     -- any id-oc, including job and server
397     object-instance  [1] ObjectIdentification,
398                     -- choice depends on class
399     message          [2] Message OPTIONAL,
400     common-arguments [3] CommonArguments OPTIONAL }

```

401 The following argument elements shall be supported by the server:

402 **object-class**

403 This argument element shall specify the object class of the **object-instance** to be deleted. The **object-class** argument
404 element shall be one of the following values: **id-oc-auxiliary-sheet**, **id-oc-auxiliary-sheet-package**, **id-oc-delivery-**
405 **method**, **id-oc-document**, **id-oc-finishing**, **id-oc-font**, **id-oc-initial-value-document**, **id-oc-initial-value-job**, **id-oc-**
406 **imposition**, **id-oc-job**, **id-oc-medium**, **id-oc-output**, **id-oc-printer**, **id-oc-resource**, **id-oc-resource-context**, **id-oc-**
407 **scheduler**, or **id-oc-transfer-method**.

408 **object-instance**

409 This argument element shall specify the **object-instance** based on the value of the **object-class** argument element.
410 The administrator shall specify one **object-instance** value associated with the **object-class**.

411 For each supported **object-class**, the server shall immediately delete the object specified by the **object-instance**
412 identified by the **ObjectIdentification** element.

413 A print server shall either accept or reject a request to delete an **object-instance**. The print server shall return any
414 attribute warnings or abstract-errors to the print client.

415 Each **object-instance** shall take one of the following forms:

object-class Specification	ObjectIdentification element	object-instance Values
id-oc-job	job-identifier	Global job identifier. If the job is not printing, the server shall immediately delete the job object, and all of its contained documents. The server shall delete a currently printing job as soon as the printer technology allows it to be deleted.
id-oc-document	document-identifier	Document identifier. The document sequence numbers of subsequent documents shall remain unchanged. If the last (or only) document in a multi-document print job is removed, the server shall delete the entire print job. The server shall delete a currently printing document as soon as the printer technology allows it to be deleted.
id-oc-printer	simple-name	Printer name representing a logical or physical printer. The server shall disable a printer before deleting it. See the Disable operation description in 8.3.5. The server shall stop printing any currently printing job on the printer, as soon as the printer technology allows it to be stopped, in an unspecified manner. Any print job submitted to print on a deleted printer shall remain in its current state, submitted to that printer, until such time as the printer object is recreated or the job is

		<p>resubmitted to another printer.</p> <p>NOTE 1 - As an example, if the printer is a physical printer, a server implementation may wait for the jobs to complete or may require that the administrator first clean the printer of all of its jobs. See the Clean operation description in 8.3.1. Thus, if the physical printer contains jobs, then the server may reject the operation and return the deletion-not-possible UpdateError to the print client.</p> <p>If the physical printer does not contain jobs, then the server may proceed with the Delete operation and remove the physical printer. If the deleted physical printer was the only one specified in the physical-printers-requested job attribute, and the physical-printers-requested attribute was compulsory, i.e. not listed in the non-compulsory-attributes attribute, then the server shall put the job in the held state with the on-request-resources-required attribute containing the physical-printers-requested attribute. Alternatively, if the deleted physical printer was again the only one specified in the physical-printers-requested job attribute, but the physical-printers-requested attribute was listed in the non-compulsory-attributes attribute, then the physical-printers-requested job attribute shall be added to the ignored-attributes attribute and the job shall remain in the print system and shall be reassigned to another physical printer. If the printer is a logical printer, the server implementation shall delete the logical printer. Once the logical printer has been deleted, the following server and printer attributes shall be updated to reflect this change: logical-printers-supported, logical-printers-ready, and printer-associated-printers.</p>
id-oc-server	simple-name	<p>Server name.</p> <p>The server shall immediately delete the server object in an unspecified manner. The ability to delete a server object is implementation-defined.</p> <p>NOTE 2 - As an example, a server implementation may require that the administrator first clean the server of all of its jobs. See the Clean operation description in 8.3.1. Thus, if the server contains jobs, then the server may reject the operation and return the deletion-not-possible UpdateError to the print client. If the server does not contain jobs, then it may proceed with the Delete operation.</p>
id-oc-initial-value-document	name-or-oid	<p>Initial-value-document name. Contains the attributes of the document class that the server may default when constructing a document object.</p>

id-oc-initial-value-job	name-or-oid	Initial-value-job name. Contains the attributes of the job class that the server may default when constructing a job object.
id-oc-auxiliary-sheet	name-or-oid	Start sheet, end sheet, separator sheet name.
id-oc-auxiliary-sheet-package	name-or-oid	Predefined package of start sheets, end sheets, and separator sheets referenced by a single name.
id-oc-delivery-method	object-identifier	Delivery method identifier; such as for print job output.
id-oc-finishing	name-or-oid	Finishing method name; such as stapling, folding, or binding.
id-oc-font	font-reference	Font name.
id-oc-imposition	name-or-oid	Imposition method name; such as 2-up or 4-up.
id-oc-medium	name-or-oid	Media name; such as iso-a4-white or north-american-letter.
id-oc-output	object-identifier	Output method identifier; such as sorting or collating.
id-oc-resource	name-or-oid	Resource identifier; such as overlay or logo.
id-oc-resource-context	name-or-oid	Resource context name; encapsulate details about the server's environment that, when used with a resource name, uniquely identifies a resource in the environment of that server.
id-oc-scheduler	object-identifier	Scheduler name; determines the order in which jobs are sent to physical printer.
id-oc-transfer-method	object-identifier	Transfer method identifier; such as ftam , with-request , or push-by-client .

416 **message**

417 See 8.1.2.

418 **common-arguments**

419 See 8.1.2 in ISO/IEC 10175-1 for a description of the **CommonArguments** type.

420 8.3.4.2 Delete-result

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

```
422     DeleteResult ::= SEQUENCE {
423         object-status    [0] SET OF Attribute OPTIONAL }
```

424 No specific set of attributes are required to be returned.

425 8.3.4.3 Delete Abstract-errors

426 If the request fails, the server shall return one of the listed abstract-errors. The circumstances under which the server will return a particular abstract-error are defined in 8.4 of ISO/IEC 10175-1.

428 8.3.5 Disable

429 This abstract-operation allows an administrator to disable the acceptance of new print jobs (**Print**, **ResubmitJob**, or **InterruptJob** operations) on the specified print server or printer. A disabled print server or printer shall continue to accept other operations defined in this International Standard.

432 Any print job that had previously been submitted to a printer that is now disabled shall be unaffected. Any currently printing job on a printer that is disabled shall continue printing to completion.

434 An administrator with appropriate privileges shall use the **Enable** operation to enable acceptance of print jobs on a specified server or printer.

436 This abstract-operation is formally defined as follows:

```

437     Disable ::= ABSTRACT-OPERATION
438         ARGUMENT      DisableArgument
439         RESULT        DisableResult
440         ERRORS        { AccessError,
441                         SecurityError,
442                         ServiceError }

```

443 NOTE - Normally only an appropriately authorized administrator may disable an object. However, this is a site and/or
444 enterprise-specific policy decision, not mandated by this International Standard.

445 8.3.5.1 Disable-argument

446 The argument of this abstract-operation is formally defined as follows:

```

447     DisableArgument ::= SEQUENCE {
448         object-class      [0] OBJECT IDENTIFIER,
449                             -- id-oc-server, -id-oc-printer
450         object-instance   [1] ObjectIdentification,
451                             -- choice depends on class
452         message           [2] Message OPTIONAL,
453                             -- value of id-att-server-message,
454                             -- id-att-printer-message
455         common-arguments [3] CommonArguments OPTIONAL }

```

456 The following argument elements shall be supported by the server:

457 object-class

458 This argument element shall specify the object class of the **object-instance**. The **object-class** argument shall be one
459 of the following values: **id-oc printer** or **id-oc-server**.

460 object-instance

461 This argument element shall specify the **object-instance** based on the value of the **object-class** argument element.
462 The administrator shall specify one **object-instance** value associated with the **object-class**.

463 For each supported **object-class**, the server shall disable the object specified by the **object-instance** identified by the
464 **ObjectIdentification** element.

465 A print server shall either accept or reject a request to disable a printer or server. The print server shall return any
466 attribute warnings or abstract-errors to the print client.

467 Each **object-instance** shall take one of the following forms:

object-class Specification	ObjectIdentification element	object-instance Values
id-oc-printer	simple-name	Printer name representing a logical or physical printer. For both logical and physical printers, the print server shall stop accepting new print job requests (from the Print , ResubmitJob , and InterruptJob operations) on the specified printer. If the printer is a physical printer, it shall finish printing any jobs that are currently printing on that physical printer.
id-oc-server	simple-name	Server name. After a Disable operation is issued, the print server shall reject all new print requests, i.e. shall reject the Print , ResubmitJob , or InterruptJob operations.

		The print server shall continue to accept requests from other operations defined in this International Standard, such as ListObjectAttributes , CancelJob , or ModifyJob .
--	--	---

468 **message**

469 See 8.1.2.

470 **common-arguments**

471 See 8.1.2 in ISO/IEC 10175-1 for a description of the **CommonArguments** type.

472 **8.3.5.2 Disable-result**

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

```
474     DisableResult ::= SEQUENCE {
475         object-status    [0] SET OF Attribute OPTIONAL }
```

476 The server may return object-status attributes of the disabled object.

477 **8.3.5.3 Disable Abstract-errors**

478 If the request fails, the server shall return one of the listed abstract-errors. The circumstances under which the server will return a particular abstract-error are defined in 8.4 of ISO/IEC 10175-1.

480 **8.3.6 Enable**

481 This abstract-operation allows an administrator to enable the acceptance of new print jobs (**Print**, **ResubmitJob**, or **InterruptJob** operations) on the specified print server or printer. Any jobs that had previously been submitted to the print server or printer shall continue unaffected.

484 An administrator with appropriate privileges shall use the **Disable** operation to disable acceptance of new print jobs on a specified server or printer.

486 This abstract-operation is formally defined as follows:

```
487     Enable ::= ABSTRACT-OPERATION
488         ARGUMENT      EnableArgument
489         RESULT        EnableResult
490         ERRORS        { AccessError,
491                         SecurityError,
492                         ServiceError }
```

493 NOTE - Normally only an appropriately authorized administrator may enable an object. However, this is a site and/or enterprise-specific policy decision, not mandated by this International Standard.

495 **8.3.6.1 Enable-argument**

496 The argument of this abstract-operation is formally defined as follows:

```
497     EnableArgument ::= SEQUENCE {
498         object-class    [0] OBJECT IDENTIFIER,
499                         -- id-oc-server, -id-oc-printer
500         object-instance [1] ObjectIdentification,
501                         -- choice depends on class
502         message         [2] Message OPTIONAL,
503                         -- value of id-att-server-message,
504                         -- id-att-printer-message
505         common-arguments [3] CommonArguments OPTIONAL }
```


506 The following argument elements shall be supported by the server:

507 **object-class**

508 This argument element shall specify the object class of the **object-instance**. The **object-class** argument shall be one
509 of the following values: **id-oc-printer** or **id-oc-server**.

510 **object-instance**

511 This argument element shall specify the **object-instance** based on the value of the **object-class** argument element.
512 The administrator shall specify one **object-instance** value associated with the **object-class**.

513 For each supported **object-class**, the server shall enable the object specified by the **object-instance** identified by the
514 **ObjectIdentification** element.

515 A print server shall either accept or reject a request to enable a printer or server. The print server shall return any
516 attribute warnings or abstract-errors to the print client.

517 Each **object-instance** shall take one of the following forms:

object-class Specification	ObjectIdentification element	object-instance Values
id-oc-printer	simple-name	Printer name representing a logical or physical printer. If the printer is a logical printer, the print server shall resume accepting new print job requests for that logical printer. If the printer is a physical printer, the print server shall resume accepting new print jobs and submitting jobs to that physical printer.
id-oc-server	simple-name	Server name. After an Enable operation is issued, the print server shall accept valid new print requests from the Print , ResubmitJob , or InterruptJob operations.

518 **message**

519 See 8.1.2.

520 **common-arguments**

521 See 8.1.2 in ISO/IEC 10175-1 for a description of the **CommonArguments** type.

522 **8.3.6.2 Enable-result**

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

```
524     EnableResult ::= SEQUENCE {
525         object-status    [0] SET OF Attribute OPTIONAL }
```

526 The server may return object-status attributes of the enabled object.

527 **8.3.6.3 Enable Abstract-errors**

528 If the request fails, the server shall return one of the listed abstract-errors. The circumstances under which the server
529 will return a particular abstract-error are defined in 8.4 of ISO/IEC 10175-1.

530 **8.3.7 Pause**

531 This abstract-operation allows an administrator to pause a print server or a printer. An administrator with
532 appropriate privileges shall use the **Resume** operation to resume the paused printer or server.

533 This abstract-operation is formally defined as follows:

```

534 Pause ::= ABSTRACT-OPERATION
535     ARGUMENT      PauseArgument
536     RESULT        PauseResult
537     ERRORS        { AccessError,
538                   AttributeError,
539                   SecurityError,
540                   SelectionError,
541                   ServiceError }
    
```

542 NOTE - Normally only an appropriately authorized administrator may pause an object. However, this is a site and/or
 543 enterprise-specific policy decision, not mandated by this International Standard.

544 **8.3.7.1 Pause-argument**

545 The argument of this abstract-operation is formally defined as follows:

```

546 PauseArgument ::= SEQUENCE {
547     object-class      [0] OBJECT IDENTIFIER,
548                       -- id-oc-printer, id-oc-server
549     object-identification [1] ObjectIdentification,
550     message           [2] Message OPTIONAL,
551                       -- value of id-att-printer-message,
552                       -- id-att-server-message, id-att-message
553     common-arguments  [3] CommonArguments OPTIONAL }
    
```

554 The following argument elements shall be supported by the server:

555 **object-class**

556 This argument element shall specify the object class of the **object-identification**. The **object-class** argument shall be
 557 one of the following values: **id-oc-printer** or **id-oc-server**.

558 **object-identification**

559 This argument element shall specify the **object-identification** based on the value of the **object-class** argument
 560 element. The administrator shall specify one **object-identification** value associated with the **object-class**.

561 For each supported **object-class**, the server shall pause the object specified by the **object-identification** identified by
 562 the **ObjectIdentification** element.

563 A print server shall either accept or reject a request to pause an **object-identification**. The print server shall return
 564 any attribute warnings or abstract-errors to the print client.

565 Each **object-identification** shall take one of the following forms:

object-class Specification	ObjectIdentification element	object-identification Values
id-oc-printer	simple-name	Printer name representing a physical printer. If a job is currently processing or printing the server shall: a) stop printing the job at an unspecified pausable point, b) change the printer's printer-state to paused . c) change the job's job-state-of-printers-assigned to paused . Any job that was currently printing is left with a job-state of processing or printing , and still assigned to that printer.

		<p>Use of the Pause and Resume operations for a logical printer is not supported.</p> <p>NOTE - A server implementation may treat the paused state as a stronger pause than the job-start-wait, job-end-wait, and job-password-wait states, since for the other pause states operator actions instead of the Resume operation may cause the printer to resume printing. Thus an administrator may invoke the Pause operation when the printer is already paused for a different reason, thus requiring the administrator to invoke the Resume operation only once to resume printing. Administrators may prefer this at a time when they do not want the printer to resume printing automatically. Regardless of the reason for the printer to be paused, the administrator need only issue one Resume operation to resume printing.</p>
id-oc-server	simple-name	<p>Server name.</p> <p>When pausing a server, the server shall:</p> <ol style="list-style-type: none"> stop distributing print jobs to physical printers (this means print jobs that are currently in the pending job state shall be left in that state), and change the server-state to on-request, <p>All other server operations shall be unaffected.</p>

566 **message**

567 See 8.1.2.

568 **common-arguments**

569 See 8.1.2 in ISO/IEC 10175-1 for a description of the **CommonArguments** type.

570 8.3.7.2 Pause-result

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

```

572     PauseResult ::= SEQUENCE {
573         affected-job-identifier    [0] JobIdentifier OPTIONAL,
574                                 -- returned if a job was printing on the
575                                 -- printer being paused
576         paused-object-status      [1] SET OF Attribute OPTIONAL }

```

577 The server may return object-status attributes of the paused object.

578 8.3.7.3 Pause Abstract-errors

579 If the request fails, the server shall return one of the listed abstract-errors. The circumstances under which the server will return a particular abstract-error are defined in 8.4 of ISO/IEC 10175-1.

581 8.3.8 Resubmit-job

582 This abstract-operation allows an appropriately authorized user to request that print jobs be resubmitted by a print server to a specified target printer. Both the target printer and its associated print server can be the same as, or different from, the original printer and print server. The resubmit-job request may specify one particular job, or all jobs that have been submitted to a particular printer or server, but which have not yet begun printing.

586 Normally only the owner of a particular job, or an appropriately authorized administrator or operator, may request
 587 resubmission of that job, and only an appropriately authorized administrator or operator may request resubmission of
 588 all jobs on a printer or server, . However, this is a site and/or enterprise-specific policy decision, not mandated by this
 589 International Standard.

590 Within this discussion of the ResubmitJob operation, the adjectives *new* and *old* are used to distinguish between
 591 instances of jobs, printers and servers For example, a requester *submits* a job initially to a printer on a particular
 592 server; later, the requester submits a ResubmitJob request to the server, which is now referred to as the *old* server.
 593 The request specifies that the job is to be resubmitted to a *new printer*, which is supported by a *new server*.

594 Similarly, the term *old job* refers to the job as *submitted* initially to the old printer and server, while the term *new job*
 595 refers to the job that has been *resubmitted*, and which is executed on the new printer and server, regardless of
 596 whether either the new printer or the new server is the same or different from the old printer and server. This
 597 terminology is used regardless of whether the job is resubmitted to the same or a different printer, on the same or a
 598 different server.

599 If the new print server is different from the old print server, the old print server shall act as a temporary client, using
 600 the Print abstract-operation to transfer each component of the job(s) to the new print server. If the resubmitted job is
 601 to remain on the original server, the mechanism for resubmission is implementation-specific.

602 **Single job resubmission**

603 A server may accept a ResubmitJob request if the identified job is in one of the following states:

- pending
- paused
- held
- retained

604 The server shall reject the ResubmitJob request and return an **inappropriate-object-state AccessError** to the print
 605 client if:

- the job is in a valid state, but **job-state-reasons** indicates ‘documents-needed’
- the job is in any of the other states, i.e.: preprocessing, processing (and printing), interrupted, terminating, completed

606 If the server determines that the job is in an acceptable state and condition, the server begins preparation of the job
 607 for resubmission to the new printer/server. The old server shall initialize the new job from the old job; that is, the
 608 new job shall contain all of the original job and document attributes and any attributes that the old print server had
 609 defaulted. This allows the new print job to be as similar to the original print job as possible. The ResubmitJob
 610 argument may also supply a new value for any job or document attribute that can be modified by a ModifyJob
 611 operation. In addition, the requester may supply a new value for the **accounting-information** attribute if the new job
 612 is to be charged to a different account.

613 NOTE - in a multi-document job, if a document attribute is included in the set of attributes to be modified in the new
 614 job, that attribute modification will be applied to each of the documents in the new job.

615 If the new printer is supported by a server that is different from the old server, the old server shall act as a normal
 616 print client to the new server, and shall transfer the new job to the new server by means of a series of Print operations
 617 created from the old job’s job and document attributes, plus .the attributes supplied in the ResubmitJob argument. If
 618 the new server is the same as the old server, the method of transferring the new job is implementation-specific;
 619 however, the effect of the ResubmitJob request, as seen by the client, shall be the same as if the two servers are
 620 different.

621 When the new print server accepts the initial **create-job** print request from the old server, the new server shall assign
 622 a new job identifier and shall return the new identifier to the old print server (which may or may not be the same as

623 the new print server). The old print server shall maintain the old instance of the print job in the state it was in when
624 the ResubmitJob request was submitted.

625 When the new print server accepts the **close-job** print request, this signals that the new job has been transferred to
626 the new server successfully. The old server shall then perform the following actions:

- return an object-status to the ResubmitJob client indicating successful resubmission of the job, and providing the **new-job-identifier**
- adjust the state of the old job, if necessary:
 - if the old job had been in the pending, paused or held state when the ResubmitJob request was received, the old server shall transition the old job through the terminating state to the retained state;
 - if the old job had already been in the retained state when the ResubmitJob request was received, the old server shall leave the old job in the retained state.

627 The old job shall remain in the **retained** state until the **job-retention-period** expires, or the job is deleted by some
628 other means; the old job identifier shall remain valid until the old job exits the completed state.

629 If the new server rejects one of the the new job print requests from the old server:

- If the new server rejects the initial **create-job** print request, no new job identifier is created, and the job is not instantiated on the new server;
- If the new server rejects any of the subsequent print requests for a new job, the old server shall cancel the new job on the new server and shall set the **job-retention-period** to 0 (zero) to remove all traces of the new job on the new server;
- the old job shall remain in the same state it had been in prior to the ResubmitJob request;
- the old server shall return an error to the ResubmitJob client; the error return shall contain the same error that was returned to the old server in the error return from the new server.

630 **Resubmission of all jobs on a printer or server**

631 A ResubmitJob request may be submitted to a print server to request that all jobs currently on that server or on a
632 particular associated printer be transferred to a new printer. An operator might invoke a ResubmitJob operation, for
633 example, to to prepare a server or printer for shut down.

634 Not all jobs can be successfully resubmitted by this means. In order to be a candidate for resubmission when its
635 containing server or printer is being resubmitted, a job must be in one of the following states:

- pending
- paused
- held

636 The server shall not resubmit a job if it is not in a proper state and condition for resubmission; i.e., if:

- the job is in a valid state, but **job-state-reasons** indicates 'documents-needed'
- the job is in one of the states: preprocessing, processing (and printing), interrupted, terminating, retained, completed

637 Processing of each of the jobs in a printer/server resubmission shall proceed as specified for the single job
638 resubmission, with the exception of the return to the ResubmitJob client. ResubmitJob is a synchronous operation;
639 that is, the old server does not return a result to the client until after each of the jobs on the old printer or server has
640 either been resubmitted successfully to the new printer/server, or has failed to be resubmitted for some reason.

641 Upon completion of the ResubmitJob operation, the old server shall return an object-status for each job that was on
642 the old server when the ResubmitJob request was submitted. The object-status for a particular job shall contain the
643 **new-job-identifier** if the job was successfully resubmitted. But if the job was *not* successfully resubmitted, the
644 object-status shall contain *only* the old job identifier for that job. In order to determine the specific reason for the

645 unsuccessful resubmission of a particular job, the requestor may invoke another ResubmitJob request for that single
646 job.

647 For each job that was resubmitted successfully, a **job-resubmitted** report event shall be generated by the old server if
648 so requested in the job's **notification-profile** attribute. The requester shall receive said notification which shall
649 include the old and new job identifier. If a job has *not* been resubmitted successfully, no notification shall be
650 generated, and the job shall remain on the old server, in the state it was in when the ResubmitJob request was
651 submitted.

652
653 This abstract-operation is formally defined as follows:

```
654     ResubmitJob ::= ABSTRACT-OPERATION
655         ARGUMENT      ResubmitJobArgument
656         RESULT        ResubmitJobResult
657         ERRORS        { AccessError,
658                         AttributeError,
659                         PrinterError,
660                         SecurityError,
661                         SelectionError,
662                         ServiceError,
663                         UpdateError }
```

664 8.3.8.1 Resubmit-job-argument

665 The argument of this abstract-operation is formally defined as follows:

```
666     ResubmitJobArgument ::= SEQUENCE {
667         object-class      [0] OBJECT IDENTIFIER,
668                         -- id-oc-job, id-oc-printer,
669                         -- id-oc-server
670         object-identification [1] ObjectIdentification,
671         printer             [2] DistinguishedNameString,
672         job-attr-modification [3] Sequence of JobAttrModification OPTIONAL,
673         message            [4] Message OPTIONAL,
674         common-arguments   [5] CommonArguments OPTIONAL }
```

675 The following argument elements shall be supported by the server:

676 **object-class**

677 This argument element shall specify the object class of the **object-identification**. The **object-class** argument shall be
678 one of the following values: **id-oc-job**, **id-oc-printer**, or **id-oc-server**

679 **object-identification**

680 This argument element shall specify the **object-identification** based on the value of the **object-class** argument
681 element.

682 For each supported **object-class**, the server shall resubmit the specified print jobs to the printer specified by the
683 **printer** argument element. The specified jobs are those associated with the object specified by the **object-**
684 **identification** identified by the **ObjectIdentification** element. A job must be in the proper state and condition for
685 resubmission of the job to be successful [see 8.3.8, preceding].

686 Each **object-identification** shall take one of the following forms:

object-class Specification	ObjectIdentification element	object-identification Values
id-oc-job	job-identifier	Global job identifier. The server shall resubmit the specified print job.
id-oc-printer	simple-name	Printer name representing a logical or physical printer. The server shall resubmit all currently submitted print jobs for the specified printer.
id-oc-server	simple-name	Server name. The server shall resubmit all currently submitted print jobs for all printers that it supports.

687 **printer**

688 This argument element shall specify a single printer name to which the specified print jobs are to be resubmitted.

689 **job-attr-modification**

690 A ResubmitJob client may specify attributes of the job that are to be modified for submission to the new server and
691 printer. The conditions governing this component of the ResubmitJob argument are exactly as defined for the same
692 component of the ModifyJob argument, with one exception: the client may specify a new value for the **accounting-**
693 **information** attribute to enable the new job to be charged to a different account.

694 **message**

695 See 8.1.2.2.

696 **common-arguments**

697 See 8.1.2 in ISO/IEC 10175-1 for a description of the **CommonArguments** type.

698 **8.3.8.2 Resubmit-job-result**

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

```

700     ResubmitJobResult ::= SEQUENCE {
701         result-set      [0] SEQUENCE OF ObjectStatus }
702         -- one result-set for each job resubmitted
703         -- (or for each job attempted to be resubmitted)
704
705     ObjectStatus ::= SEQUENCE {
706         object-status [0] SET OF Attribute OPTIONAL }
707         -- job-identifier and new-job-identifier shall be
708         -- returned at least. For any jobs that could not
709         -- be resubmitted, the new-job-identifier attribute
710         -- shall be omitted as the only error indication.
```

710 NOTE - The reason for the apparent overspecification of SEQUENCE in the above two productions is to permit future
711 additions to be made without unduly disrupting existing implementations.

712 **object-status**

713 The server shall at least return the following job-status attributes of each resubmitted job:

new-job-identifier - for each successfully resubmitted (i.e., new) job

job-identifier - of the old job

714 For each job that is successfully resubmitted to the same server or another server, both the **new-job-identifier** and
 715 the **job-identifier** attributes shall be returned. For each job that is not resubmitted successfully, only the **job-**
 716 **identifier** attribute shall be returned.

717 When more than one job is specified, as may be the case when the **object-identification** argument element is of
 718 **object-class id-oc-printer** or **id-oc-server**, the absence of the **new-job-identifier** attribute is the only error
 719 indication that the print client receives in the ResubmitJobResult.

720 8.3.8.3 Resubmit-job Abstract-errors

721 If the request fails, the server shall return one of the listed abstract-errors. The circumstances under which the server
 722 will return a particular abstract-error are defined in 8.4 of ISO/IEC 10175-1.

723 When the ResubmitJob request has failed because the new server has returned an error in response to a Print
 724 operation submitted by the old server, the old server shall then return that error to the ResubmitJob client.

725 8.3.9 Resume

726 This abstract-operation allows an administrator to resume a paused print server or a paused printer.

727 An administrator with appropriate privileges shall use the **Pause** operation to pause the printer and server.

728 This abstract-operation is formally defined as follows:

```

729     Resume ::= ABSTRACT-OPERATION
730             ARGUMENT      ResumeArgument
731             RESULT        ResumeResult
732             ERRORS        { AccessError,
733                             AttributeError,
734                             SecurityError,
735                             SelectionError,
736                             ServiceError }
  
```

737 NOTE - Normally only an appropriately authorized administrator may resume an object. However, this is a site and/or
 738 enterprise-specific policy decision, not mandated by this International Standard.

739 8.3.9.1 Resume-argument

740 The argument of this abstract-operation is formally defined as follows:

```

741     ResumeArgument ::= SEQUENCE {
742         object-class          [0] OBJECT IDENTIFIER,
743                             -- id-oc-printer, id-oc-server
744         object-identification [1] ObjectIdentification,
745         message               [2] Message OPTIONAL,
746                             -- value of id-att-printer-message,
747                             -- id-att-server-message, id-att-message
748         common-arguments     [3] CommonArguments OPTIONAL }
  
```

749 The following argument elements shall be supported by the server:

750 object-class

751 This argument element shall specify the object class of the **object-identification**. The **object-class** argument shall be
 752 one of the following values: **id-oc-printer** or **id-oc-server**.

753 object-identification

754 This argument element shall specify the **object-identification** based on the value of the **object-class** argument
 755 element. The administrator shall specify one **object-identification** value associated with the **object-class**.

756 For each supported **object-class**, the server shall start the paused object specified by the **object-identification**
757 identified by the **ObjectIdentification** element.

758 A print server shall either accept or reject a request to resume an **object-identification**. The print server shall return
759 any attribute warnings or abstract-errors to the print client.

760 Each **object-identification** shall take one of the following forms:

object-class Specification	ObjectIdentification element	object-identification Values
id-oc-printer	simple-name	Printer name representing a physical printer. When resumed, a physical printer shall resume printing its currently assigned print job, if any, at the same unspecified pausable point where it was stopped. Use of the Pause and Resume operations for a logical printer is not supported.
id-oc-server	simple-name	Server name. when resumed, the print server shall begin distributing print jobs to printers.

761 **message**

762 See 8.1.2.

763 **common-arguments**

764 See 8.1.2 in ISO/IEC 10175-1 for a description of the **CommonArguments** type.

765 8.3.9.2 Resume-result

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

```
767     ResumeResult ::= SEQUENCE {
768         object-status      [0] SET OF Attribute OPTIONAL }
```

769 The server may return object-status attributes of the resumed object.

770 8.3.9.3 Resume Abstract-errors

771 If the request fails, the server shall return one of the listed abstract-errors. The circumstances under which the server
772 will return a particular abstract-error are defined in 8.4 of ISO/IEC 10175-1.

773 8.3.10 Set

774 This abstract-operation allows an administrator to set attribute values of a specified printing object.

775 Some restrictions on modifications to attributes of object classes apply and are described below.

776 The server shall be able to modify the following Job Object attributes:

- all Job Information attributes except for id-att-job-identifier, id-att-job-originator, id-att-job-owner, and id-att-initial-value-job
- all Job Results Handling attributes
- all Job Event Handling attributes
- all Job Scheduling Instruction attributes
- all Printer Selection attributes except for id-att-printer-name-requested

777 The server shall not modify the following Job Object attributes: Access and Accounting attributes, Job Security
778 attributes, Job Status attributes, and Job Size attributes.

- The server shall be able to modify the following Document Object attributes:
- all Document Information attributes
- all Document Production Instructions attributes except for id-att-initial-value-document
- all Document Description attributes except for id-att-document-content and id-att-transfer-method
- all Document Characteristic attributes

779 The server shall not modify the following Document Object attributes: Document Status attributes.

780 The list of attributes in each of the above categories shall contain at least the list of attributes defined in the ISO/IEC
781 10175-1 standard definition of these categories. The addition of attributes to these general categories beyond those
782 defined in the ISO/IEC 10175-1 standard is implementation defined.

783 Restrictions on modifications to other attributes of other object classes are implementation-defined.

784 There are four "modification" operations: **replace**, **add-values**, **remove-values**, **set-to-default**.

785 NOTE 1 - The client may change whether a document attribute or a job attribute is compulsory or non-compulsory. If
786 an attribute is to be specified as non-compulsory, then the non-compulsory-attributes attribute (for a document
787 attribute) or the job-non-compulsory-attributes attribute (for a job attribute) shall contain the attribute-id of the attribute
788 that the server is to consider as non-compulsory. If an attribute is to be specified as compulsory, then the non-
789 compulsory-attributes attribute or the job-non-compulsory-attributes attribute shall *not* contain the attribute-id of the
790 attributes that the server is to consider as compulsory.

791 The server shall perform the requested modifications on the specified print object. Changes made by the server shall
792 be persistent, that is, objects shall retain their modified values across system restarts.

793 The behavior of a server when using the **Set** abstract-operation to modify job attributes or document attributes shall
794 be the same as the behavior when using the **ModifyJob** abstract-operation. See 8.2.2 in ISO/IEC 10175-1 for a
795 description of the **ModifyJob** abstract-operation.

796 However, for printer and server object classes, the ability of a server to successfully set a printer attribute or a server
797 attribute, and the resulting effect, depends upon the attribute involved and upon the presence of the attribute-type
798 object identifier in the **printer-non-compulsory-attributes** attribute or the **server-non-compulsory-attributes**
799 attribute. If the server cannot perform the specified modification for any compulsory attribute (e.g., due to
800 conflicting requirements, unavailability of a feature, or current status of the object), the server shall reject the **Set**
801 abstract-operation.

802 In general, the server shall ignore or reject modifications to particular printer or server attributes in a **Set** abstract-
803 operation in a manner consistent with the treatment they would have received had they been submitted in the **Create**
804 abstract-operation arguments. However, if it is necessary to reject a specified modification to any attribute (e.g., a
805 compulsory attribute with an unsupported value is specified), then the server shall reject the entire **Set** abstract-
806 operation and the printer or server attributes will remain unchanged.

807 In particular, if an attempt is made to remove an attribute from the value of the **printer-non-compulsory-attributes**
808 or **server-non-compulsory-attributes**, the server shall reject the **Set** abstract-operation if such a value would have
809 caused the original **Create** abstract-operation to be rejected. In this case, the server shall report an **invalid-non-**
810 **compulsory-attribute-modification AttributeError**.

811 The server shall reject any **Set** abstract-operation that attempts to modify (1) the attributes that this International
812 Standard specifies as non-modifiable or (2) the attributes for which the server does not support modification, and
813 shall return an **illegal-modification AttributeError** to the print client. If the server rejects a modification because it
814 does not support modification of an attribute or attributes, the server shall identify all such attributes in the
815 **AttributeError** returned.

816 This abstract-operation is formally defined as follows:

```

817 Set ::= ABSTRACT-OPERATION
818     ARGUMENT      SetArgument
819     RESULT        SetResult
820     ERRORS        { AccessError,
821                   SecurityError,
822                   ServiceError }

```

823 NOTE 2 - Normally only an appropriately authorized administrator set object attributes. However, this is a site and/or
824 enterprise-specific policy decision, not mandated by this International Standard.

825 8.3.10.1 Set-argument

826 The argument of this abstract-operation is formally defined as follows:

```

827 SetArgument ::= SEQUENCE {
828     object-class      [0] OBJECT IDENTIFIER, -- id-oc-xxx
829     object-identification [1] ObjectIdentification,
830     message           [2] Message OPTIONAL,
831     attr-modification [3] SEQUENCE OF AttrModification OPTIONAL,
832     common-arguments [4] CommonArguments OPTIONAL }
833
834 AttrModification ::= SEQUENCE {
835     attribute-id      [0] AttributeId,
836     attribute-values [1] SET OF ANY -- DEFINED BY attribute-id -- OPTIONAL,
837     modify-operator [2] ModifyOperator DEFAULT replace }

```

837

838 The following argument elements shall be supported by the server:

839 **object-class**

840 This argument element shall specify the object class of the **object-identification** to modify. The **object-class**
841 argument shall be one of the following values: **id-oc-auxiliary-sheet**, **id-oc-auxiliary-sheet-package**, **id-oc-**
842 **delivery-method**, **id-oc-document**, **id-oc-finishing**, **id-oc-font**, **id-oc-initial-value-document**, **id-oc-initial-value-**
843 **job**, **id-oc-imposition**, **id-oc-job**, **id-oc-medium**, **id-oc-output**, **id-oc-printer**, **id-oc-resource**, **id-oc-resource-**
844 **context**, **id-oc-scheduler**, **id-oc-server**, or **id-oc-transfer-method**.

845 **object-identification**

846 This argument element shall specify the **object-identification** based on the value of the **object-class** argument
847 element. The administrator shall specify one **object-identification** value associated with the **object-class**.

848 For each supported **object-class**, the server shall set the attributes for the object specified by the **object-**
849 **identification** identified by the **ObjectIdentification** element.

850 A print server shall either accept or reject a request to set attributes for an **object-identification**. The print server
851 shall return any attribute warnings or abstract-errors to the print client.

852 Each **object-identification** shall take one of the following forms:

object-class Specification	ObjectIdentification element	object-identification Values
id-oc-job	job-identifier	Global job identifier. The server shall modify or set job and document attributes of previously submitted print jobs which have not yet begun printing on the physical printer. If the job has already begun printing, the results are unspecified.

		The effect of modifying job and/or document attributes on a print job is the same as if that job or document had been submitted originally with the modified attribute values. Scheduling of the job may be impacted, depending on the print server implementation and the attribute modifications made.
id-oc-document	document-identifier	Document identifier. Documents are numbered starting with 1. The ability to modify a currently printing document is implementation-defined. The effect of modifying document attributes on a document is the same as if that document had been submitted originally with the modified attribute values. Scheduling of the print job may be impacted, depending on the print server implementation and the attribute modifications made.
id-oc-printer	simple-name	Printer name representing a logical or physical printer.
id-oc-server	simple-name	Server name.
id-oc-initial-value-document	name-or-oid	Initial-value-document name. Contains the attributes of the document class that the server may default when constructing a document object.
id-oc-initial-value-job	name-or-oid	Initial-value-job name. Contains the attributes of the job class that the server may default when constructing a job object.
id-oc-auxiliary-sheet	name-or-oid	Start sheet, end sheet, separator sheet name.
id-oc-auxiliary-sheet-package	name-or-oid	Predefined package of start sheets, end sheets, and separator sheets referenced by a single name.
id-oc-delivery-method	object-identifier	Delivery method identifier; such as for print job output.
id-oc-finishing	name-or-oid	Finishing method name; such as stapling, folding, or binding.
id-oc-font	font-reference	Font name.
id-oc-imposition	name-or-oid	Imposition method name; such as 2-up or 4-up.
id-oc-medium	name-or-oid	Media name; such as iso-a4-white or north-american-letter.
id-oc-output	object-identifier	Output method identifier; such as sorting or collating.
id-oc-resource	name-or-oid	Resource identifier; such as overlay or logo.
id-oc-resource-context	name-or-oid	Resource context name; encapsulate details about the server's environment that, when used with a resource name, uniquely identifies a resource in the environment of that server.
id-oc-scheduler	object-identifier	Scheduler name; determines the order in which jobs are sent to physical printer.
id-oc-transfer-method	object-identifier	Transfer method identifier; such as ftam , with-request , or push-by-client .

853 message

854 See 8.1.2.

855 attr-modification

856 This argument element shall specify the modification to be performed on the **object-identification**, and contains the
857 following necessary information:

- 858 a) **attribute-id**: identification of attribute to be modified
- 859 b) **attribute-values**: value(s) to be assigned to the attribute (if appropriate)
- 860 c) **modify-operator**: type of modification to be performed. Possible modification types are described in
861 the following table:

modify-operator Value	Description
replace	Replaces the entire value of of the attribute identified by attribute-id with the value(s) specified in attribute-values , or adds the attribute to the print object, if the attribute was not present.
add-values	Adds the values specified in attribute-values to the attribute. identified by attribute-id
remove-values	Removes the value(s) specified in attribute-values from the attribute identified by attribute-id to the server-default.
set-to-default	Sets the value of the attribute identified by attribute-id to the server-default.

862 **common-arguments**

863 See 8.1.2 in ISO/IEC 10175-1 for a description of the **CommonArguments** type.

864 **8.3.10.2 Set-result**

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

```
866 SetResult ::= SEQUENCE {  
867     object-status [0] SET OF Attribute OPTIONAL }
```

868 The server shall return the supported subset of the **object-status** attributes of the object whose attributes were set.

869 The following table contains a list of attributes for each **object-class**.

Object	object-class Specification	Attributes
Job	id-oc-job	id-att-job-identifier id-att-job-name id-att-current-job-state id-att-job-state-reasons id-att-intervening-jobs id-att-printer-name-requested id-att-printers-assigned id-att-total-job-octets id-att-job-owner
Document	id-oc-document	id-att-document-sequence-number id-att-document-format id-att-document-state id-att-octet-count id-att-copy-count id-att-sides id-att-document-type
Printer	id-oc-printer	id-att-printer-name id-att-printer-realization

		id-att-printer-state id-att-availability id-att-associated-server id-att-printer-locations id-att-printer-associated-printers id-att-scheduler-ready
Server	id-oc-server	id-att-server-name id-att-server-state id-att-availability id-att-logical-printers-supported id-att-physical-printers-supported
Medium	id-oc-medium	id-att-medium-identifier id-att-associated-server id-att-medium-realization id-att-medium-type id-att-medium-associated-media id-att-medium-size id-att-medium-dimensions id-att-medium-sides id-att-medium-color
Imposition	id-oc-imposition	id-att-imposition-identifier id-att-associated-server id-att-imposition-specification id-att-number-up
Initial-value-job	id-oc-initial-value-job	id-att-initial-value-job-identifier id-att-associated-server id-att-logical-printers-ready id-att-physical-printers-ready id-att-printer-locations-requested id-att-printer-models-requested id-att-job-retention-period
Initial-value-document	id-oc-initial-value-document	id-att-initial-value-document-identifier id-att-associated-server id-att-logical-printers-ready id-att-physical-printers-ready id-att-copy-count id-att-sides id-att-document-format
Font	id-oc-font	id-att-font-identifier id-att-associated-server id-att-file-reference
Auxiliary-sheet	id-oc-auxiliary-sheet	id-att-auxiliary-sheet-identifier id-att-auxiliary-sheet-medium id-att-associated-server
Auxiliary-sheet-package	id-oc-auxiliary-sheet-package	id-att-auxiliary-sheet-package-identifier id-att-associated-server
Transfer-method	id-oc-transfer-method	id-att-transfer-method-identifier id-att-transfer-method-type id-att-document-reference-type id-att-associated-server
Resource	id-oc-resource	id-att-resource-name id-att-resource-type

		id-att-associated-server id-att-file-reference
Resource-context	id-oc-resource-context	id-att-resource-context-identifier id-att-context-address id-att-associated-server
Delivery-method	id-oc-delivery-method	id-att-delivery-method-identifier id-att-associated-server
Finishing	id-oc-finishing	id-att-finishing-identification id-att-finishing-realization id-att-associated-server id-att-finishing-specification id-att-finishing-associated-finishings
Output	id-oc-output	id-att-output-identifier id-att-associated-server
Scheduler	id-oc-scheduler	id-att-scheduler-identifier id-att-associated-server

870 8.3.10.3 Set Abstract-errors

871 If the request fails, the server shall return one of the listed abstract-errors. The circumstances under which the server
872 will return a particular abstract-error are defined in 8.4 of ISO/IEC 10175-1.

873 8.3.11 Shutdown

874 This abstract-operation allows an administrator to shutdown a specified print server or physical printer.

875 The means for restarting a printer or server which has been shutdown with this operation is provided by the **Control**
876 operation with the **reset** attribute set to **reset-power-cycle**.

877 The **when** argument element shall specify how much processing must occur before the shutdown can take place. The
878 server shall stop printing currently printing jobs either immediately (as soon as the printer technology allows), or as
879 indicated by the **when** argument element.

880 The server shall keep any jobs that had been submitted to the printer or server and continue processing them when
881 the printer or server is restarted. The server may also submit the jobs to an alternate physical printer, if one is
882 available.

883 When a print server begins a shutdown operation, it first disables itself. This shall prevent new print jobs from being
884 accepted. Currently scheduled print jobs on a print server being shutdown shall be saved by the server the print jobs
885 shall be re-scheduled when the print server is restarted.

886 The order in which jobs will be printed shall not be changed by the invocation of the **Shutdown** operation.

887 The ability to shutdown physical printers is an implementation option.

888 This operation may operate asynchronously; i.e., this operation need not complete immediately before returning.

889 This abstract-operation is formally defined as follows:

```
890     Shutdown ::= ABSTRACT-OPERATION
891         ARGUMENT      ShutdownArgument
892         RESULT        ShutdownResult
893         ERRORS        { AccessError,
894                        SecurityError,
895                        ServiceError }
```

896 NOTE - Normally only an appropriately authorized administrator may shutdown an object. However, this is a site
897 and/or enterprise-specific policy decision, not mandated by this International Standard.

898 **8.3.11.1 Shutdown-argument**

899 The argument of this abstract-operation is formally defined as follows:

```

900 ShutdownArgument ::= SEQUENCE {
901     object-class      [0] OBJECT IDENTIFIER,
902                     -- id-oc-server, -id-oc-printer
903     object-instance  [1] ObjectIdentification,
904                     -- choice depends on class
905     message          [2] Message OPTIONAL,
906     when             [3] OBJECT IDENTIFIER,
907                     -- now, after-current, after-all
908     common-arguments [4] CommonArguments OPTIONAL }
    
```

909 The following argument elements shall be supported by the server:

910 **object-class**

911 This argument element shall specify the object class of the **object-instance**. The **object-class** argument shall be one of the following values: **id-oc-printer** or **id-oc-server**.

913 **object-instance**

914 This argument element shall specify the **object-instance** based on the value of the **object-class** argument element. The administrator shall specify one **object-instance** value associated with the **object-class**.

916 For each supported **object-class**, the server shall shutdown the object specified by the **object-instance** identified by the **ObjectIdentification** element.

918 A print server shall either accept or reject a request to shut down an **object-instance**. The print server shall return any attribute warnings or abstract-errors to the print client.

920 Each **object-instance** shall take one of the following forms:

object-class Specification	ObjectIdentification element	object-instance Values
id-oc-printer	simple-name	Printer name representing a physical printer. The server shall: <ul style="list-style-type: none"> a) Disable the physical printer. This prevents the physical printer from accepting new print requests. b) Based on the value of the when argument element, stop printing as soon as possible (now), allow the physical printer to complete its current print job (after-current), or allow the physical printer to complete all jobs currently submitted to it (after-all). c) Place the physical printer in the shutdown state. Restart is accomplished by the Control operation with the reset attribute set to id-val-reset-power-cycle. Use of the Shutdown operation for a logical printer is not supported.
id-oc-server	simple-name	Server name. To shut down a server, the print server shall: <ul style="list-style-type: none"> a) Disable the server. This prevents the print server from accepting new print jobs. b) If the value of the when argument element is not

		<p>after-all, place the print server in the on-request state. This prevents print jobs from being assigned to physical printers.</p> <p>c) Based on the value of the when argument element, for each physical printer stop printing as soon as possible (now), allow physical printers to complete their current print jobs (after-current), or allow the physical printers to complete all jobs currently submitted to them (after-all).</p> <p>d) Place each physical printer in the paused state.</p> <p>e) Remove entries in the name service for all objects supported by this print server, if such entries exist and are required by the server implementation.</p> <p>f) Place the print server in the terminating state.</p> <p>g) Retain (save) any jobs not yet printed to be re-scheduled on server start-up.</p>
--	--	---

921 **message**

922 See 8.1.2.

923 **when**

924 This argument element shall specify how much processing must occur before the shutdown can take place. The
925 when argument shall be one of the following values:

Descriptive Name:	Object Identifier	action taken:
now	id-val-generic-now	Aborts current jobs as soon as possible
after-current	id-val-generic-after-current	Currently printing jobs finish printing
after-all	id-val-generic-after-all	All submitted print jobs finish printing

926 NOTE - these object identifier values are defined in Annex A of ISO/IEC 10175-1.

927 If some condition prevents a printer or server from shutting down promptly, an administrator may invoke another
928 Shutdown operation with a more severe **when** value; i.e., if the original **when** value was 'after-all', the administrator
929 may invoke Shutdown with a **when** value of 'after-current' or 'now'. If the original **when** value was 'after-current',
930 the administrator may invoke a subsequent Shutdown with a **when** value of 'now'. But note that the converse is not
931 true; i.e., if an administrator attempts to invoke Shutdown with a **when** value of 'after-all' while a Shutdown
932 operation that had been invoked with a **when** value of 'now' or 'after-current' is currently executing, the new
933 Shutdown operation will be rejected.

934 **common-arguments**

935 See 8.1.2 in ISO/IEC 10175-1 for a description of the **CommonArguments** type.

936 **8.3.11.2 Shutdown-result**

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

```
938 ShutdownResult ::= SEQUENCE {
939     object-status [0] SET OF Attribute OPTIONAL }
```

940 The server shall return specified printer-status attributes of the shutdown object.

941 **8.3.11.3 Shutdown Abstract-errors**

942 If the request fails, the server shall return one of the listed abstract-errors. The circumstances under which the server
943 will return a particular abstract-error are defined in 8.4 of ISO/IEC 10175-1.

944 **8.4 Abstract-Errors**

945 See 8.4 of ISO/IEC 10175-1.

946 Many reasons for failure of an operation are generally not specified by the operation descriptions. Operations may
947 terminate prematurely and return an abstract-error to the print client if they encounter invalid usage of argument
948 elements (see 8.4.2 of ISO/IEC 10175-1, **Attribute-error**); difficulties accessing objects (see 8.4.1 of ISO/IEC
949 10175-1, **Access-error**, 8.4.6 of ISO/IEC 10175-1, **Selection-error**, or 8.4.8 of ISO/IEC 10175-1, **Update-error**); or
950 difficulties associated with the privileges of the process (see 8.4.5 of ISO/IEC 10175-1, **Security-error**). When an
951 unrecoverable error condition is encountered, the operation shall return an abstract-error to the print client (see 8.4.4
952 of ISO/IEC 10175-1, **Printer-error** or 8.4.7 of ISO/IEC 10175-1, **Service-error**).

953 **8.5 Abstract-Events**

954 See 8.5 of ISO/IEC 10175-1.

955

956 **Annex A**

957 (normative)

959 **Formal definition of the DPA management abstract service**

960

961
962 This Annex is a supplement to section two, formally defining the Document Printing management abstract service. It
963 employs ASN.1 and the ABSTRACT-OPERATION and ABSTRACT-ERROR macros of ISO/IEC 10021-3.

964 The order of the ASN.1 productions in this Annex follows the order of the operation definitions in clause 8. In case
965 of discrepancies between this Annex and clause 8, this Annex shall take precedence.

966

967 -- File: DPA_3A.ASN, Version 1.0, 5-December-1997
968 -- Version 0.1, 20-October-1996
969 -- Version 1.0, 5-December-1997: Removed temporary notes;
970 -- Removed temporary definitions of abstract-errors
971 -- [these are now defined in Part 1 via Technical Corrigendum 1]

972

973

974 -- ANNEX A

975

976 -- FORMAL DEFINITION OF THE DPA MANAGEMENT ABSTRACT SERVICE

977

978 -- This normative annex is an integral part of this part of ISO/IEC
979 -- 10175.

980

981 -- This Annex is a supplement to section two and formally defines the
982 -- Document Printing management abstract service. It employs ASN.1 and the
983 -- ABSTRACT-OPERATION and ABSTRACT-ERROR macros of ISO/IEC 10021-3.

984

985

986 DPAManagementAbstractService {iso standard dpa(10175) part-3(3) modules(0)
987 abstract-service(1) }

988 DEFINITIONS IMPLICIT TAGS ::=

989 BEGIN

990 -- PROLOGUE --

991 -- EXPORTS everything --

992

993 IMPORTS

994

995 -- Abstract service macros --

996 ABSTRACT-ERROR, ABSTRACT-OPERATION

997 FROM AbstractServiceNotation

998 { joint-iso-ccitt mhs-motis(6) asdc(2) modules(0) notation(1) }

999

1000

1001 -- Data types from DPA abstract service (see ISO/IEC 10175-1 ANNEX B) --

1002 Attribute, AttributeId, CommonArguments, JobAttrModification, ModifyOperator,

1003 ObjectIdentification, AccessError, AttributeError, PrinterError, SecurityError,

```

1004 SelectionError, ServiceError, UpdateError
1005     FROM DPAAbstractService
1006         { iso standard dpa(10175) part-1(1) modules(0) abstract-service(1) }
1007
1008 -- Data types from DPA attributes (see ISO/IEC 10175-1 ANNEX C) --
1009 Message, JobIdentifier, DistinguishedNameString
1010     FROM DPASstandardAttributes
1011         {iso standard dpa(10175) part-1(1) modules(0) attributes(2) }
1012
1013
1014 -- end of IMPORTS

1015 -----
1016
1017 -- ABSTRACT OPERATIONS --
1018
1019 -- Clean --
1020
1021 Clean ::= ABSTRACT-OPERATION
1022     ARGUMENT      CleanArgument
1023     RESULT        CleanResult
1024     ERRORS        { AccessError,
1025                   SecurityError,
1026                   ServiceError }
1027
1028 CleanArgument ::= SEQUENCE {
1029     object-class    [0] OBJECT IDENTIFIER,
1030                   -- id-oc-server, id-oc-printer
1031     object-instance [1] ObjectIdentification,
1032                   -- choice depends on object class
1033     message         [2] Message OPTIONAL,
1034     common-arguments [3] CommonArguments OPTIONAL }
1035
1036
1037 CleanResult ::= SEQUENCE {
1038     object-status [0] SET OF Attribute OPTIONAL }
1039
1040 -----
1041
1042 -- Control --
1043
1044 Control ::= ABSTRACT-OPERATION
1045     ARGUMENT      ControlArgument
1046     RESULT        ControlResult
1047     ERRORS        { AccessError,
1048                   SecurityError,
1049                   ServiceError }
1050

```

```

1051 ControlArgument ::= SEQUENCE {
1052     object-class      [0] OBJECT IDENTIFIER,
1053                       -- id-oc-server, id-oc-printer
1054     object-instance  [1] ObjectIdentification,
1055                       -- choice depends on object class
1056     message          [2] Message OPTIONAL,
1057     operations       [3] SEQUENCE OF Attribute,
1058                       -- operands and parameters
1059                       -- Current operations: reset and trace
1060     common-arguments [4] CommonArguments OPTIONAL }
1061
1062 ControlResult ::= SEQUENCE {
1063     object-status [0] SET OF Attribute OPTIONAL }
1064
1065 -----
1066
1067 -- Create --
1068
1069 Create ::= ABSTRACT-OPERATION
1070     ARGUMENT      CreateArgument
1071     RESULT        CreateResult
1072     ERRORS        { AccessError,
1073                   SecurityError,
1074                   SelectionError,
1075                   ServiceError }
1076
1077 CreateArgument ::= SEQUENCE {
1078     object-class      [0] OBJECT IDENTIFIER, -- id-oc-xxx
1079                       -- any id-oc-xxx, excluding job and document
1080     object-instance  [1] ObjectIdentification,
1081                       -- instance identifier depends on class
1082     message          [2] Message OPTIONAL,
1083     copy-from        [3] ObjectIdentification OPTIONAL,
1084                       -- initial value object (IVO)
1085     force-creation   [4] BOOLEAN DEFAULT FALSE,
1086                       -- recreate if necessary
1087     object-attributes [5] SET OF Attribute OPTIONAL,
1088                       -- attributes of new object instance
1089     common-arguments [6] CommonArguments OPTIONAL }
1090
1091 CreateResult ::= SEQUENCE {
1092     object-status [0] SET OF Attribute }
1093
1094 -----
1095
1096 -- Delete --
1097
1098 Delete ::= ABSTRACT-OPERATION
1099     ARGUMENT      DeleteArgument
1100     RESULT        DeleteResult
1101     ERRORS        { AccessError,
1102                   SecurityError,
1103                   ServiceError,
1104                   UpdateError }

```

```

1104
1105 DeleteArgument ::= SEQUENCE {
1106     object-class      [0] OBJECT IDENTIFIER,
1107                     -- any id-oc, including job and server
1108     object-instance  [1] ObjectIdentification,
1109                     -- choice depends on class
1110     message          [2] Message OPTIONAL,
1111     common-arguments [3] CommonArguments OPTIONAL }
1112
1113 DeleteResult ::= SEQUENCE {
1114     object-status    [0] SET OF Attribute OPTIONAL }
1115
1116
1117 -- Disable --
1118
1119 Disable ::= ABSTRACT-OPERATION
1120     ARGUMENT      DisableArgument
1121     RESULT        DisableResult
1122     ERRORS        { AccessError,
1123                   SecurityError,
1124                   ServiceError }
1125
1126 DisableArgument ::= SEQUENCE {
1127     object-class      [0] OBJECT IDENTIFIER,
1128                     -- id-oc-server, -id-oc-printer
1129     object-instance  [1] ObjectIdentification,
1130                     -- choice depends on class
1131     message          [2] Message OPTIONAL,
1132                     -- value of id-att-server-message,
1133                     -- id-att-printer-message
1134     common-arguments [3] CommonArguments OPTIONAL }
1135
1136 DisableResult ::= SEQUENCE {
1137     object-status    [0] SET OF Attribute OPTIONAL }
1138
1139 -- Enable --
1140
1141 Enable ::= ABSTRACT-OPERATION
1142     ARGUMENT      EnableArgument
1143     RESULT        EnableResult
1144     ERRORS        { AccessError,
1145                   SecurityError,
1146                   ServiceError }
1147
1148 EnableArgument ::= SEQUENCE {
1149     object-class      [0] OBJECT IDENTIFIER,
1150                     -- id-oc-server, -id-oc-printer
1151     object-instance  [1] ObjectIdentification,
1152                     -- choice depends on class
1153     message          [2] Message OPTIONAL,
1154                     -- value of id-att-server-message,
1155                     -- id-att-printer-message
1156     common-arguments [3] CommonArguments OPTIONAL }

```

```

1157 EnableResult ::= SEQUENCE {
1158     object-status    [0] SET OF Attribute OPTIONAL }
1159
1160 -----
1161
1162 -- Pause --
1163
1164 Pause ::= ABSTRACT-OPERATION
1165     ARGUMENT      PauseArgument
1166     RESULT        PauseResult
1167     ERRORS        { AccessError,
1168                     AttributeError,
1169                     SecurityError,
1170                     SelectionError,
1171                     ServiceError }
1172
1173 PauseArgument ::= SEQUENCE {
1174     object-class      [0] OBJECT IDENTIFIER,
1175                     -- id-oc-printer, id-oc-server
1176     object-identification [1] ObjectIdentification,
1177     message           [2] Message OPTIONAL,
1178                     -- value of id-att-printer-message,
1179                     -- id-att-server-message, id-att-message
1180     common-arguments  [3] CommonArguments OPTIONAL }
1181
1182 PauseResult ::= SEQUENCE {
1183     affected-job-identifier [0] JobIdentifier OPTIONAL,
1184                     --returned if a job was printing on the
1185                     --printer being paused
1186     paused-object-status  [1] SET OF Attribute OPTIONAL }
1187
1188 -----
1189
1190 -- ResubmitJob --
1191
1192 ResubmitJob ::= ABSTRACT-OPERATION
1193     ARGUMENT      ResubmitJobArgument
1194     RESULT        ResubmitJobResult
1195     ERRORS        { AccessError,
1196                     AttributeError,
1197                     PrinterError,
1198                     SecurityError,
1199                     SelectionError,
1200                     ServiceError,
1201                     UpdateError }
1202
1203 ResubmitJobArgument ::= SEQUENCE {
1204     object-class      [0] OBJECT IDENTIFIER,
1205                     -- id-oc-job, id-oc-printer,
1206                     -- id-oc-server
1207     object-identification [1] ObjectIdentification,
1208     printer            [2] DistinguishedNameString,

```

```

1209     job-attr-modification [3] Sequence of JobAttrModification OPTIONAL,
1210     message                 [4] Message OPTIONAL,
1211     common-arguments       [5] CommonArguments OPTIONAL }
1212
1213 ResubmitJobResult ::= SEQUENCE {
1214     result-set      [0] SEQUENCE OF ObjectStatus }
1215     -- one result-set for each job resubmitted
1216     -- (or for each job attempted to be resubmitted)
1217
1218 ObjectStatus ::= SEQUENCE {
1219     object-status [0] SET OF Attribute OPTIONAL }
1220     -- job-identifier and new-job-identifier shall be
1221     -- returned at least. For any jobs that could not
1222     -- be resubmitted, the new-job-identifier attribute
1223     -- shall be omitted as the only error indication.
1224
1225
1226 -----
1227
1228 -- Resume --
1229 Resume ::= ABSTRACT-OPERATION
1230     ARGUMENT      ResumeArgument
1231     RESULT        ResumeResult
1232     ERRORS        { AccessError,
1233                   AttributeError,
1234                   SecurityError,
1235                   SelectionError,
1236                   ServiceError }
1237
1238 ResumeArgument ::= SEQUENCE {
1239     object-class      [0] OBJECT IDENTIFIER,
1240     -- id-oc-printer, id-oc-server
1241     object-identification [1] ObjectIdentification,
1242     message           [2] Message OPTIONAL,
1243     -- value of id-att-printer-message,
1244     -- id-att-server-message, id-att-message
1245     common-arguments  [3] CommonArguments OPTIONAL }
1246
1247 ResumeResult ::= SEQUENCE {
1248     object-status     [0] SET OF Attribute OPTIONAL }
1249
1250 -----
1251
1252 -- Set --
1253
1254 Set ::= ABSTRACT-OPERATION
1255     ARGUMENT      SetArgument
1256     RESULT        SetResult
1257     ERRORS        { AccessError,
1258                   SecurityError,
1259                   ServiceError }
1260
1261 SetArgument ::= SEQUENCE {

```



```

1262     object-class           [0] OBJECT IDENTIFIER, -- id-oc-xxx
1263     object-identification [1] ObjectIdentification,
1264     message                 [2] Message OPTIONAL,
1265     attr-modification      [3] SEQUENCE OF AttrModification OPTIONAL,
1266     common-arguments       [4] CommonArguments OPTIONAL }
1267
1268     AttrModification ::= SEQUENCE {
1269         attribute-id        [0] AttributeId,
1270         attribute-values    [1] SET OF ANY -- DEFINED BY attribute-id -- OPTIONAL,
1271         modify-operator     [2] ModifyOperator DEFAULT replace }
1272
1273     setResult ::= SEQUENCE {
1274         object-status       [0] SET OF Attribute OPTIONAL }
1275
1276     -----
1277
1278     -- Shutdown --
1279     Shutdown ::= ABSTRACT-OPERATION
1280         ARGUMENT          ShutdownArgument
1281         RESULT            ShutdownResult
1282         ERRORS            { AccessError,
1283                           SecurityError,
1284                           ServiceError }
1285
1286     ShutdownArgument ::= SEQUENCE {
1287         object-class       [0] OBJECT IDENTIFIER,
1288                             -- id-oc-server, -id-oc-printer
1289         object-instance    [1] ObjectIdentification,
1290                             -- choice depends on class
1291         message            [2] Message OPTIONAL,
1292         when               [3] OBJECT IDENTIFIER,
1293                             -- now, after-current, after-all
1294                             -- [see Part 1, Annex A for OID definition]
1295         common-arguments  [4] CommonArguments OPTIONAL }
1296
1297     ShutdownResult ::= SEQUENCE {
1298         object-status      [0] SET OF Attribute OPTIONAL }
1299
1300     -----
1301
1302     -- ABSTRACT ERRORS --
1303
1304     See Part 1, Annex B, of this International Standard.
1305
1306
1307     END -- of DPAManagementAbstractService --
1308

```

1309
1310
1311
1312
1313
1314

Annex B

(normative)

Conformance requirements

1315 This Annex is a supplement to section two. Its purpose is to define different levels of DP-Server functionality in
1316 terms of the level of support provided for the abstract operations and attributes defined in this International Standard.
1317 These levels are to be used in specifying the conformance levels which particular DP-Servers purport to implement,
1318 and thus, which features and facilities clients of those DP-Servers may expect to access.

1319 Two levels of functionality are defined in ISO/IEC 10175-1, levels 1 and 2; for each level, an additional set of
1320 attributes is specified for multiple-document job support, indicated by the letter M. Thus, level 1M indicates the
1321 server supports the basic set of operations and attributes, and can support multiple-document jobs. See ISO/IEC
1322 10175-1, Annex E, for specifications of these conformance requirements.

1323 This part of ISO/IEC 10175 defines one additional level of conformance, level 3; additional conformance
1324 requirements are also specified for level 3 systems providing multiple-document support, designated level 3M.

1325 The Conformance level operation and attribute sets are defined in the following subclauses:

1326 B.1 Conformance level 3

1327 A server conforming to level 3 shall support the operations, objects, and attributes specified for level 2 conformance,
1328 and in addition shall support the following operations and attributes, as a minimum:

1329 a) Abstract operations supported:

1330 Clean
1331 Control
1332 Create
1333 Delete
1334 Disable
1335 Enable
1336 Pause
1337 Resume
1338 Set
1339 Shutdown

1340 b) Attributes supported:

1341 **enabled**

1342 B.2 Conformance level 3M

1343 This classification is applicable to level 3 servers that support multiple-document jobs.

1344 A server conforming to level 3M shall support the operations, objects, and attributes specified for level 3
1345 conformance, and shall in addition adhere to the conformance requirements specified for level 2M.

1346

