INTERNATIONAL STANDARD

ISO/IEC 10175-3

First edition
2000-12-15

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

Part 3: Management abstract service definitions and procedures

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

Partie 3: Définitions et procédures de service abstrait de gestion

Reference number
ISO/IEC 10175-3:2000(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

Foreword

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

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

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

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 10175 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 10175-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*.

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

— *Part 1: Abstract service definition and procedures*

— *Part 2: Protocol specification*

— *Part 3: Management abstract service definitions and procedures*

Annexes A and B form a normative part of this part of ISO/IEC 10175.

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

Part 3: Management abstract service definitions and procedures

Section 1 - General

1 Scope

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

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

Part 1: Abstract service definitions and procedures

Part 2: Protocol specification

Part 3: Management abstract service definition and procedures

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

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

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

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 10175. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 10175 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 7498-1: 1994,	*Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model.*
ISO/IEC 7498-2: 1989,	*Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture.*
ISO/IEC 8824: 1990,	*Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).*
ISO/IEC 8825: 1990,	*Information technology - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*
ISO/IEC 10021-3: 1990,	*Information technology - Text Communication - Message-Oriented Text Interchange Systems (MOTIS) - Part 3: Abstract Service Definition Conventions.*
ISO/IEC 10175-1: 1996,	*Information technology - Text and office systems - Document Printing Application (DPA) - Part 1: Abstract service definition and procedures.*
IEEE 1387.4 / D9,	*Draft Standard for Information Technology - POSIX System Administration - Part 4: Printing Interfaces, IEEE, March 1998.*

3 Definitions

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

4 Abbreviations

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

5 Conventions

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

Section 2 - DP management abstract service definition

6 DP abstract model

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

7 Abstract association information

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

8 Abstract operations

This clause defines the following abstract-operations:

a) Clean
b) Control
c) Create
d) Delete
e) Disable
f) Enable
g) Pause
h) ResubmitJob
i) Resume
j) Set
k) Shutdown

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

The table below lists each of the above abstract-operations and all of the abstract-operations of ISO-IEC 10175-1, 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	

NOTES

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

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

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

8.1 Common datatypes and values used in DP management abstract operations

8.1.1 Imported datatypes

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

From ISO/IEC 10175-1.Annex B:

Attribute,
AttributeId,
CommonArguments,
JobAttrModification,
ModifyOperator
ObjectIdentification,

AccessError,
AttributeError,
PrinterError,
SecurityError,
SelectionError,
ServiceError,
UpdateError

From ISO/IEC 10175-1.Annex C:

DistinguishedNameString.
JobIdentifier,
Message

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

8.1.2 Datatypes common for most DPA abstract operations

8.1.2.1 Common-arguments

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

8.1.2.2 Message

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

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

If messages are specified in one locale and retrieved in another, a variety of results may occur depending on the message specification from the client locale and at the server locale. See the table below for different possible results 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

8.2 DP-User Port Abstract Operation Definitions

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

8.3 DP-Administration Port Abstract-Operation Definitions

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

8.3.1 Clean

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

a) If the object is enabled, reject the abstract-operation and report an **inappropriate-object-state** AccessError; otherwise,

b) If the object is a physical printer, pause the printer to stop the printer from producing output (see **Pause** operation),

c) Remove all jobs contained in or scheduled for that object,

d) Restore the state of the object to what it was when the execution of the **Clean** operation began.

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

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

This abstract-operation is formally defined as follows:

```
Clean ::= ABSTRACT-OPERATION
      ARGUMENT        CleanArgument
      RESULT          CleanResult
      ERRORS          {  AccessError,
                         SecurityError,
                         ServiceError }
```

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

8.3.1.1 Clean-argument

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

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

The following argument elements shall be supported by the server:

object-class

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

object-instance

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.**

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

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

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 **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. Use of the **Clean** operation for a logical printer is not supported. 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**.
id-oc-server	**simple-name**	Server name. 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. 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.

message

See 8.1.2.

common-arguments

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

8.3.1.2 Clean-result

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

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

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

8.3.1.3 Clean Abstract-errors

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.

8.3.2 Control

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

This abstract-operation is formally defined as follows:

```
Control ::=   ABSTRACT-OPERATION
      ARGUMENT        ControlArgument
      RESULT          ControlResult
      ERRORS          {  AccessError,
                         SecurityError,
                         ServiceError }
```

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

8.3.2.1 Control-argument

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

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

The following argument elements shall be supported by the server:

object-class

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

object-instance

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.**

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

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

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: • **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**. Use of the **Control** operation for a logical printer is not supported.
id-oc-server	**simple-name**	Server name. The behavior of the **Control** operation depends on the **operations** argument element, which may be one or more of the following attributes: • **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**.

message

See 8.1.2.

operations

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

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

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

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

common-arguments

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

8.3.2.2 Control-result

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

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

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

8.3.2.3 Control Abstract-errors

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.

8.3.3 Create

This abstract-operation allows an administrator to create a printing object and set its attributes to the values specified within the argument elements. The **object-class** argument element shall indicate the class of the object. The server shall create the specified object instance as an object of the specified **object-class**. When the Create operation is invoked by an administrator with appropriate privileges, the server shall perform the following actions:

a) The server shall initialize the newly created object with the attributes specified by the **object-attributes** argument element. This has the same effect as issuing an equivalent **Set** operation. Note however, that some attributes may be settable with the **Create** operation, but may be restricted from further modification with the **Set** operation. See the **Set** operation description and the restrictions on attribute modification in 8.3.10. If the **copy-from** argument element is specified, the server creates a copy of the object specified by the **copy-from** argument element, and the attributes specified by the **object-attributes** argument element are applied subsequently.

b) If the specified object already exists and the **force-creation** argument element is set to **FALSE**, the print server shall return the **object-already-exists SelectionError** for this operation. If the **force-creation** argument element is set to **TRUE**, the print server shall immediately delete the existing object and shall create the new object. If the printer is not able to immediately complete the deletion of the existing object due to the state of the printer, the server shall rename the existing object so that the creation of the new object can complete.

c) An **object-instance** created by the **Create** operation shall be permanent; the object is persistent, lasting across re-initialization of the print system, or any of its components.

d) When an **object-instance** of class **id-oc-printer** is created, the server shall place it in the **idle** state with the **enabled** attribute set to **FALSE** so print jobs cannot be accepted by the new printer. The administrator may then invoke the **Enable** operation to place the printer object in the **idle** state with the **enabled** attribute set to **TRUE**.

e) The server shall create an entry for the newly created **object-instance** in the name service, if such an entry is required by the server implementation.

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

This abstract-operation is formally defined as follows:

```
Create ::=ABSTRACT-OPERATION
    ARGUMENT        CreateArgument
    RESULT          CreateResult
    ERRORS          {  AccessError,
                       SecurityError,
                       SelectionError,
                       ServiceError }
```

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

8.3.3.1 Create-argument

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

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

The following argument elements shall be supported by the server:

object-class

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

object-instance

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**.

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

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

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**.

message

See 8.1.2.

copy-from

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

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

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

force-creation

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

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

When an administrator with appropriate privileges specifies the **force-creation** argument element with value **TRUE**, the server shall create the specified print object, whether or not an object with the specified name already exists. If a print object with the specified name already exists, the server shall delete the existing print object before

it creates the new print object. The server shall return no abstract-error or warning in this case. If the specified object does not exist, this argument element shall have no effect.

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

object-attributes

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

common-arguments

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

8.3.3.2 Create-result

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

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

The server shall return the supported subset of the following **object-status** attributes of the created object depending 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**

8.3.3.3 Create Abstract-errors

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.

8.3.4 Delete

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

a) If the specified object is a printer the server shall.first determine whether the printer has been disabled and is without jobs: if the printer is not in this required state, the server shall reject the abstract-operation and report an **inappropriate-object-state** AccessError; otherwise:

b) The server shall delete the specified print object(s) from the print system;

c) The server shall delete the print object entry in the name service, if an entry exists and is required by the server implementation;

d) The server shall delete a currently printing job, as soon as the printer technology allows it to be deleted, in an implementation-defined manner.

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

This abstract-operation is formally defined as follows:

```
Delete ::=ABSTRACT-OPERATION
      ARGUMENT        DeleteArgument
      RESULT          DeleteResult
      ERRORS          {  AccessError,
                         SecurityError,
                         ServiceError,
                         UpdateError }
```

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

8.3.4.1 Delete-argument

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

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

The following argument elements shall be supported by the server:

object-class

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

object-instance

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**.

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

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

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 resubmitted to another printer. 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. 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**.
id-oc-server	**simple-name**	Server name. The server shall immediately delete the server object in an unspecified manner. The ability to delete a server object is implementation-defined. 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.
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**.

message

See 8.1.2.

common-arguments

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

8.3.4.2 Delete-result

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

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

No specific set of attributes are required to be returned.

8.3.4.3 Delete Abstract-errors

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.

8.3.5 Disable

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.

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.

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

This abstract-operation is formally defined as follows:

```
Disable ::=   ABSTRACT-OPERATION
      ARGUMENT        DisableArgument
      RESULT          DisableResult
      ERRORS          {  AccessError,
                         SecurityError,
                         ServiceError }
```

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

8.3.5.1 Disable-argument

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

```
DisableArgument ::= SEQUENCE {
   object-class     [0] OBJECT IDENTIFIER,
                        -- id-oc-server, -id-oc-printer
   object-instance  [1] ObjectIdentification,
                        -- choice depends on class
   message          [2] Message OPTIONAL,
                        -- value of id-att-server-message,
                        -- id-att-printer-message
   common-arguments [3] CommonArguments OPTIONAL }
```

The following argument elements shall be supported by the server:

object-class

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**.

object-instance

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**.

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

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

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**.

message

See 8.1.2.

common-arguments

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

8.3.5.2 Disable-result

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

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

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

8.3.5.3 Disable Abstract-errors

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.

8.3.6 Enable

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.

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

This abstract-operation is formally defined as follows:

```
Enable ::=ABSTRACT-OPERATION
      ARGUMENT        EnableArgument
      RESULT          EnableResult
      ERRORS          {  AccessError,
                         SecurityError,
                         ServiceError }
```

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.

8.3.6.1 Enable-argument

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

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

The following argument elements shall be supported by the server:

object-class

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**.

object-instance

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**.

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

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

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.

message

See 8.1.2.

common-arguments

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

8.3.6.2 Enable-result

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

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

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

8.3.6.3 Enable Abstract-errors

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.

8.3.7 Pause

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

This abstract-operation is formally defined as follows:

```
Pause ::= ABSTRACT-OPERATION
      ARGUMENT        PauseArgument
      RESULT          PauseResult
      ERRORS          {  AccessError,
                         AttributeError,
                         SecurityError,
                         SelectionError,
                         ServiceError }
```

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

8.3.7.1 Pause-argument

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

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

The following argument elements shall be supported by the server:

object-class

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

object-identification

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

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

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

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' **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. Use of the **Pause** and **Resume** operations for a logical printer is not supported. 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.
id-oc-server	**simple-name**	Server name. When pausing a server, the server shall: a) 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 b) change the server-state to **on-request**, All other server operations shall be unaffected.

message

See 8.1.2.

common-arguments

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

8.3.7.2 Pause-result

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

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

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

8.3.7.3 Pause Abstract-errors

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.

8.3.8 Resubmit-job

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.

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

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

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

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

Single job resubmission

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

- pending
- paused
- held
- retained

The server shall reject the ResubmitJob request and return an **inappropriate-object-state AccessError** to the print 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

If the server determines that the job is in an acceptable state and condition, the server begins preparation of the job for resubmission to the new printer/server. The old server shall initialize the new job from the old job; that is, the

new job shall contain all of the original job and document attributes and any attributes that the old print server had defaulted. This allows the new print job to be as similar to the original print job as possible. The ResubmitJob argument may also supply a new value for any job or document attribute that can be modified by a ModifyJob operation. In addition, the requester may supply a new value for the **accounting-information** attribute if the new job is to be charged to a different account.

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

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

When the new print server accepts the initial **create-job** print request from the old server, the new server shall assign a new job identifier and shall return the new identifier to the old print server (which may or may not be the same as the new print server). The old print server shall maintain the old instance of the print job in the state it was in when the ResubmitJob request was submitted.

When the new print server accepts the **close-job** print request, this signals that the new job has been transferred to 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.

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

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 identifer 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.

Resubmission of all jobs on a printer or server

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

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

- pending
- paused
- held

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

Processing of each of the jobs in a printer/server resubmission shall proceed as specified for the single job resubmission, with the exception of the return to the ResubmitJob client. ResubmitJob is a synchronous operation; 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 either been resubmitted successfully to the new printer/server, or has failed to be resubmitted for some reason.

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

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

This abstract-operation is formally defined as follows:

```
ResubmitJob ::=  ABSTRACT-OPERATION
      ARGUMENT         ResubmitJobArgument
      RESULT           ResubmitJobResult
      ERRORS           {  AccessError,
                          AttributeError,
                          PrinterError,
                          SecurityError,
                          SelectionError,
                          ServiceError,
                          UpdateError }
```

8.3.8.1 Resubmit-job-argument

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

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

The following argument elements shall be supported by the server:

object-class

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

object-identification

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

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

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.

printer

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

job-attr-modification

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

message

See 8.1.2.2.

common-arguments

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

8.3.8.2 Resubmit-job-result

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

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

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

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

object-status

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

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

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

8.3.8.3 Resubmit-job Abstract-errors

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.

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

8.3.9 Resume

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

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

This abstract-operation is formally defined as follows:

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

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

8.3.9.1 Resume-argument

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

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

The following argument elements shall be supported by the server:

object-class

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

object-identification

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

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

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

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.

message

See 8.1.2.

common-arguments

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

8.3.9.2 Resume-result

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

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

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

8.3.9.3 Resume Abstract-errors

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.

8.3.10 Set

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

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

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

The server shall not modify the following Job Object attributes: Access and Accounting attributes, Job Security 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

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

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

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

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

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

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

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

However, for printer and server object classes, the ability of a server to successfully set a printer attribute or a server attribute, and the resulting effect, depends upon the attribute involved and upon the presence of the attribute-type object identifier in the **printer-non-compulsory-attributes** attribute or the **server-non-compulsory-attributes** attribute. If the server cannot perform the specified modification for any compulsory attribute (e.g., due to

conflicting requirements, unavailability of a feature, or current status of the object), the server shall reject the **Set** abstract-operation.

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

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

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

This abstract-operation is formally defined as follows:

```
Set ::= ABSTRACT-OPERATION
      ARGUMENT          SetArgument
      RESULT            SetResult
      ERRORS            {  AccessError,
                           SecurityError,
                           ServiceError }
```

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

8.3.10.1 Set-argument

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

```
SetArgument ::= SEQUENCE {
   object-class           [0] OBJECT IDENTIFIER, -- id-oc-xxx
   object-identification  [1] ObjectIdentification,
   message                [2] Message OPTIONAL,
   attr-modification      [3] SEQUENCE OF AttrModification OPTIONAL,
   common-arguments       [4] CommonArguments OPTIONAL }

AttrModification ::= SEQUENCE {
   attribute-id     [0] AttributeId,
   attribute-values [1] SET OF ANY -- DEFINED BY attribute-id -- OPTIONAL,
   modify-operator  [2] ModifyOperator DEFAULT replace }
```

The following argument elements shall be supported by the server:

object-class

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

object-identification

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

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

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

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**.

message

See 8.1.2.

attr-modification

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

a) **attribute-id**: identification of attribute to be modified

b) **attribute-values**: value(s) to be assigned to the attribute (if appropriate)

c) **modify-operator**: type of modification to be performed. Possible modification types are described in the following table:

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

common-arguments

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

8.3.10.2 Set-result

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

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

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

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**

8.3.10.3 Set Abstract-errors

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.

8.3.11 Shutdown

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

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

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

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

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

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

The ability to shutdown physical printers is an implementation option.

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

This abstract-operation is formally defined as follows:

```
Shutdown ::=  ABSTRACT-OPERATION
      ARGUMENT        ShutdownArgument
      RESULT          ShutdownResult
      ERRORS          {  AccessError,
                         SecurityError,
                         ServiceError }
```

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

8.3.11.1 Shutdown-argument

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

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

The following argument elements shall be supported by the server:

object-class

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**.

object-instance

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**.

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

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.

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: 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: 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 **after-all**, place the print server in the **on-request** state. This prevents print jobs from being assigned to physical printers. 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**). d) Place each physical printer in the **paused** state. 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. f) Place the print server in the **terminating** state. g) Retain (save) any jobs not yet printed to be re-scheduled on server start-up.

message

See 8.1.2.

when

This argument element shall specify how much processing must occur before the shutdown can take place. The 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