ECMA
EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-101

OFFICE DOCUMENT ARCHITECTURE

September 1985

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

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-101

OFFICE DOCUMENT ARCHITECTURE

September 1985

BRIEF HISTORY

In 1981 ECMA, recognizing the inevitable convergence of computer systems and office equipment, decided upon the constitution of a Technical Committee, TC 29, with a scope to develop standards in the field of preparation, processing and interchange of text as used in office documents.

The present standard is the first ECMA publication in this domain.

In the course of its study ECMA intensively contributed to related activities in CCITT and in ISO. The interchange format adopted in October, 1984 (Recommendation T.73) by CCITT for its Telematic Services was strongly influenced by early ECMA proposals.

Conversely, the ECMA and ISO work were influenced by the CCITT work.

At the date of publication, ISO draft proposals for a multi-part standard (DP 8613), are, due to the active participation of a group of common members, nearly identical with the present ECMA Standard.

TABLE OF CONTENTS

FIGURES

TABLES

1. GENERAL

1.1 Scope

The objective of this ECMA Standard is to facilitate the interchange of office documents. For this purpose this Standard provides:

i) the definition of an abstract document model intended for presentation of office documents;

ii) the definition of character and photographic content architectures;

iii) the format of data streams used to interchange documents structured in accordance with this model;

iv) the conformance requirements for implementations of this Standard;

v) the specification of a document processing model used in developing the document architecture.

This Standard also includes the definitions of the terms used.

Appendices are included which specify:

- definitions of control functions;
- coded representations of control functions;
- examples of document structures.

1.2 Field of Application

This ECMA Standard applies to the interchange of office documents.

In the context of this ECMA Standard, office documents are considered to be items such as memoranda, letters, forms and reports, including pictures and tabular material.

The graphic elements used within the documents include graphic characters and photographic elements, potentially both within one document.

The abstract document model and the document processing model used to describe an office document are designed to be extensible to encompass other types of documents and further types of elements such as geometric elements and digitized sound.

2. DEFINITIONS AND REFERENCES

2.1 Definitions

For the purpose of this Standard the following definitions apply. The definitions for attributes can be found in section 6.

2.1.1 Attribute

A property of a document, of a component of a document, or of a content portion. It expresses a characteristic of the document, component or content portion concerned, or a relationship with one or more other documents, components or content portions.

2.1.2 Basic

Qualifier for objects which are not sub-divided into other objects.

2.1.3 Basic Component

A basic object or a basic object definition.

2.1.4 Basic Layout Object

A basic object in the layout structure.

2.1.5 Basic Logical Object

A basic object in the logical structure.

2.1.6 Basic Measurement Unit (BMU)

A unit of measurement used for positioning and dimensioning layout objects. The absolute size of the basic measurement unit is 1/1200 of 25,4 mm if the output medium is paper and the scaling factor is 1.

2.1.7 Basic Object

An object that is not subdivided into other objects.

2.1.8 Basic Object Definition

The definition of an object class which does not refer to definitions for subordinate objects.

2.1.9 Basic Page

A page that is a basic object in a layout structure and therefore has no subordinate objects.

2.1.10 Basic Value (of an attribute or of a control function parameter)

A value which may always be used at a particular conformance level.

2.1.11 Block

A basic layout object corresponding to a rectangular area within a page or within a frame with its sides parallel to the sides of the enclosing page or frame.

2.1.12 Character

A member of a set of elements used for the organization, control or representation of information.

2.1.13 Common Content

Any content associated with object definitions.

2.1.14 Character Box

A rectangular area on the presentation medium that can be used for the presentation of one graphic character.

2.1.15 Component

An object or an object definition.

2.1.16 Composite

Qualifier for objects which are sub-divided into other objects.

2.1.18 Composite Layout Object

A composite object in the layout structure.

2.1.19 Composite Logical Object

A composite object in the logical structure.

2.1.20 Composite Object

An object that is subdivided into other composite objects and/or basic objects.

2.1.21 Composite Page

A page which is a composite object in a layout structure and that must have subordinate frames or blocks.

2.1.22 Conformance Combination

A set of conformance requirements defined in terms of document architecture level, content architecture, interchange format and document profile level.

2.1.23 Content

The information conveyed by the document, independent of logical structure and layout structure.

2.1.24 Content Architecture

Rules for defining the internal structure and presentation of the content or the common content of basic components, in terms of a set of graphic elements, attributes and control functions, with their coded representation.

2.1.25 Content Portion

A part of the content of the document associated with at most one basic logical component and at most one basic layout component.

2.1.26 Control Function

A character that affects the recording, processing, transmission or interpretation of data and that has a coded representation consisting of one or more bit combinations.

2.1.27 Data Stream

An ordered sequence of data structures, i.e. descriptors and/or text units.

2.1.28 Data Structure

A set of data items, and the relationships among them, representing a document or part of a document.

2.1.29 Default Value (of an attribute or a control function parameter)

A value that is derived through the defaulting conventions when no value is explicitly specified for the attribute or parameter.

2.1.30 Descriptor

A data structure representing a component or the document profile.

2.1.31 Document

A structured amount of text, consisting of a document profile and a document body, that can be interchanged as a unit between an originator and a recipient.

2.1.32 Document Architecture Level

A standard combination of architectural features used in the specification of a conformance level.

2.1.35 Document Architecture

Rules for structuring documents.

2.1.36 Document Body

The content of a document, the logical and layout structures and the logical and layout definitions, excluding the document profile.

2.1.37 Document Class

A set of similar documents defined by a set of common characteristics, e.g. letter, memorandum, report, invoice.

2.1.38 Document Definition

The definition of a document class, consisting of object definitions for all logical objects and layout objects that can occur in documents of the class concerned.

2.1.39 Document Profile

A set of attributes associated with a document, for the purpose of handling the document as a whole, the profile being intended for comprehension by a human and by a machine.

2.1.40 Document Structure

The result of dividing and subdividing the content of a document into increasingly smaller parts.

2.1.41 Editing

The carrying out of operations associated with creation and amendment of the structure and/or the content of a document, e.g. replacement, insertion, deletion.

2.1.42 Frame

A composite layout object corresponding to a rectangular area within a page or within another frame with its sides parallel to the sides of the enclosing page or frame.

2.1.43 Geometric Element

A graphic element that is a member of a set of primitive geometric shapes such, as points, lines, arcs, used to construct pictures.

2.1.44 Graphic Character

A graphic element that is a character, other than a control function, and that has a visual representation normally handwritten, printed or displayed.

2.1.45 Graphic Element

A basic element of the content of a document.

2.1.46 Imaging

Presentation in visible form.

2.1.47 Interchange

The process of providing a duplicate of a document to a

receiving person or device.

2.1.48 Interchange Format

Rules for representing documents by data structures for the purpose of interchange.

2.1.49 Intersection

The common area of two or more layout objects that overlap each other partially or fully on the presentation medium.

2.1.50 Layout Category

A relationship between logical objects and one or more classes of frame specifying where a logical object should be laid out.

2.1.51 Layout Definition

An object definition of a class of layout objects.

2.1.52 Layout Directive

One of a set of attributes of a logical object that controls the allocation of its contents to basic layout objects and/or affects the positioning of those basic layout objects.

2.1.53 Layout Object

One component of a document pertaining to its layout structure, e.g. page, block.

2.1.54 Layout Process

The carrying out of operations to determine the layout of a document.

2.1.55 Layout Structure

The result of dividing and subdividing the content of a document into increasingly smaller parts, on the basis of the presentation, e.g. into pages, blocks.

2.1.56 Logical Definition

An object definition of a class of logical objects.

2.1.57 Logical Object

One component of a document pertaining to the logical structure, e.g. chapter, section, paragraph.

2.1.58 Logical Structure

The result of dividing and subdividing the content of a document into increasingly smaller parts, on the basis of the meaning of the content, e.g. into chapters, sections,

paragraphs.

2.1.59 Non-basic Value (of an attribute or a control function)

A value which may only be used if it is permitted as a non-basic value of the particular conformance combination and is declared in the document profile.

2.1.60 Object

A subdivision of a document structure (layout or logical).

2.1.61 Object Class

A set of similar objects defined by a set of common characteristics.

2.1.62 Object Definition

The definition of an object class consisting of rules to assign values to attributes and/or to determine content of objects of the class concerned.

2.1.63 Object Type

A property of every object, that specifies which attributes are applicable and indicates the role of the object in the document architecture model.

2.1.64 Overlay (to)

To cover a layout object with another layout object in the area of intersection such that if the overlaying object is opaque the content of the overlayed object is not visible in the area of intersection.

2.1.65 Page

A layout object corresponding to a rectangular area which is a unit of the presentation medium.

2.1.66 Page Set

A composite layout object corresponding to a sequence of other page sets and/or pages.

2.1.67 Photographic Element

A graphic element used to construct pictures out of dots, each of which has a specified shape, size, colour, intensity and position.

2.1.68 Presentation

The operation of making a document perceptible by a human being.

2.1.69 Presentation Attribute

An attribute which controls the format and appearance of content within a basic object.

2.1.70 Presentation Medium

The carrier of information in a form perceptible to a human, e.g. a sheet of paper, or a display screen.

2.1.71 Processing

The carrying out of operations on a document, including editing, layout, presentation, filing, and retrieval processes.

2.1.72 Reference Attributes

A category of document profile attributes which provides information for handling the document as a whole.

2.1.74 Rule

An attribute of an object definition which determines the valid values of the corresponding attribute of the objects of the class defined.

2.1.75 Sequential Order

A complete ordering defined between the objects of a structure.

2.1.76 Standard Default Value

A value of a defaultable attribute or a control function parameter which is defined in the conformance section of this Standard to be used when no value is specified in the document.

2.1.77 Text

A representation of information for human perception that can be presented in a two-dimensional form.

2.1.78 Text Unit

A data structure representing a content portion.

2.2 Abbreviations

The following abbreviations are used in this Standard. The acronyms for control functions are to be found in Appendix A.

ASN.1	Abstract Syntax Notation No. 1
BMU	Basic Measurement Unit

CCITT	Comité Consultatif International Télégraphique et Téléphonique
CF	Character Content Architecture for Formatted Processible Form
CI	Character Content Architecture for Image Form
CP	Character Content Architecture for Processible Form
DPM	Document Processing Model
FPDA	Formatted Processible Document Architecture
IDA	Imaging Document Architecture
ISO	International Organization for Standardization
MIF	Mixed Image Format
MPF	Mixed Processible Format
ODA	Office Document Architecture
ODIF	Office Document Interchange Format
PDA	Processible Document Architecture
PH	Photographic Content Architecture
PIF	Page Image Format
SFD	Simple Formattable Document
TIF	Text Image Format
TPF	Text Processible Format
UTC	Universal Coordinated Time

2.3 References

2.3.1 ECMA Standards

ECMA-6	7-bit coded character set
ECMA-35	Code extension techniques
ECMA-48	Additional control functions for character imaging I/O devices

2.3.2 International Standards

ISO 2014	Writing of calendar dates in all numeric form
ISO 3307	Representation of time of the day
ISO 6937	Coded character sets for text communication

ISO 7350	Text communication registration of graphic character sub-repertoires
ISO 8613	Text structure (in preparation)
ISO 8824	OSI specification of Abstract Syntax Notation 1
ISO 8825	OSI basic encoding rules for Abstract Syntax Notation 1

2.3.3 CCITT Recommendations

T.6	Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus
T.61	Basic Teletex character sets
T.73	Document interchange protocol for the Telematic Services
X.420	Message Handling Systems: Interpersonnal Messaging, User Agent Layer

3. OFFICE DOCUMENT ARCHITECTURE OVERVIEW

3.1 Architectural Concepts

For the purpose of this Standard, a document is a structured amount of text that can be interchanged as a unit between an originator and a recipient.

A document thus need not have any direct relationship to the term document as used for purposes outside the scope of this Standard. The document in this Standard may form part of, or be the whole of, a document defined as a unit by an application using this Standard.

A document can be interchanged for two major purposes:

- it may be interchanged in an image form allowing for printing and displaying as intended by the originator;
- it may be interchanged in a processible form allowing for editing and layout revision by the recipient.

Text is a representation of information for human perception that can be presented in a two-dimensional form, e.g. printed on paper or displayed on a screen.

Text consists of graphic elements such as graphic characters, geometric elements, photographic elements and combinations of these. The content of a document consists of text and some additional control information.

The content of the document can be physically and logically structured in order to:

- delimit layout objects such as pages;
- delimit logical objects such as paragraphs;
- use different types of graphic elements;
- allow processing after interchange.

The rules for structuring documents are collectively called the document architecture.

This Standard does not require all parts of the document architecture to be present in any particular document.

3.2 Document Structure

Within a document, two different but complementary views of the content are provided for:

- the logical view associates content with entities such as chapters, headings, paragraphs, footnotes, figures and appendices;

- the layout view associates content with entities such as pages, relating to presentation media.

These entities are called objects. Each view associates the same document content with a separate structure. Each structure is a hierarchy of objects, e.g. chapters consisting of sections, consisting of paragraphs. Thus:

- the logical structure associates the content of a document with a hierarchy of logical objects;

- the layout structure associates the content of a document with a hierarchy of layout objects.

The two structures are interrelated, since the layout structure has to express the logical structure on the presentation media. This interrelationship is established by means of relationships between the objects. The relationships between objects in the logical structure and objects in the layout structure are called layout directives.

The major concepts of the document structure are illustrated in Figure 1.

Relationships also occur between objects within a structure in other ways than is expressed by the hierarchical structure, such as cross references to figures or footnotes.

The objects have defined attributes, which are used to specify relationships and other properties of objects.

The structures and relationships are independent of the type of graphic elements in the document content.

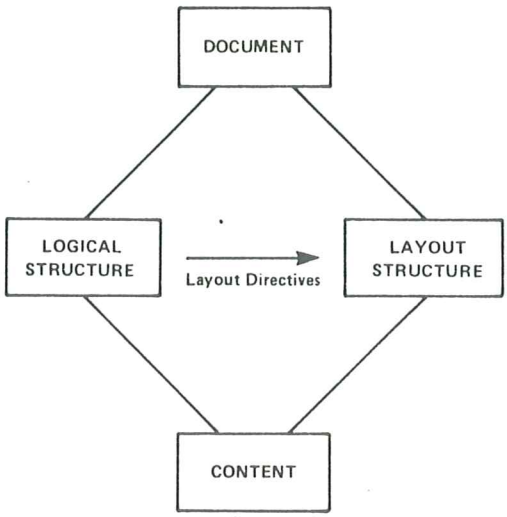


Figure 1: Document Structure

3.3 Document Content

The structures of a document partition the content into content portions associated with the objects at the lowest hierarchical levels in the structures.

Depending on their type of graphic elements the content portions are structured according to certain content architectures.

A content architecture consists of a set of rules for defining the internal structure and presentation of the content in terms of a set of graphic elements, attributes and control functions, with their coded representation.

Within one document different content portions may use different content architectures.

The document architecture supports the incorporation of the content architectures specified in section 7 of this Standard.

3.4 Object Definitions

In many documents, there may be a number of objects with common characteristics, for example:

- logical objects called sections, which are constructed from a sequence of subordinate objects called paragraphs, with the same style;
- pages with the same headers and footers.

Such a set of objects with some common attribute values is called an object class.

Object classes are not defined by this Standard, they are determined by the applications.

The specification of an object class is called an object definition.

Object definitions can be used for the following purposes:

- to improve transmission efficiency;
- to maintain the internal consistency of a document when it is modified;
- to facilitate the creation of objects and documents.

An object definition for a logical object is called a logical definition and an object definition for a layout object is called a layout definition.

An object definition has defined attributes which parallel the attributes defined for objects. An attribute of an object definition specifies a rule to determine the value of the corresponding attribute for an object of the object class concerned.

Any content associated with object definitions is called common content.

The overall document architecture is illustrated in Figure 2.

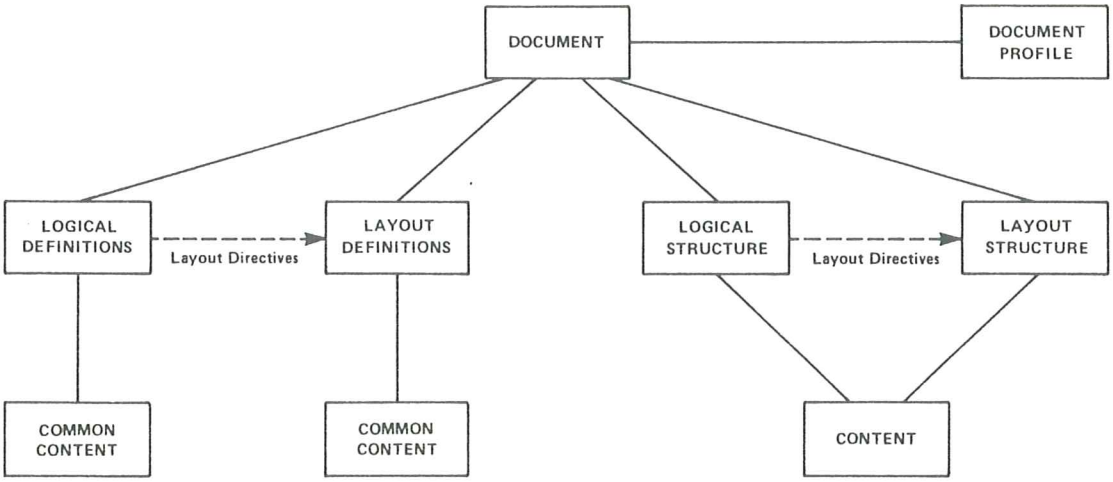


Figure 2: Document Architecture

3.5 Document Definitions

A set of documents with common characteristics is called a document class.

An example of a document class is a set of reports with common layout and common paragraphs.

The Standard does not specify document classes. These are to be determined by the applications.

The Standard allows for the definition of a document class by means of a document definition which includes all the object definitions of the objects that are common to the document class.

The common content associated with a document definition is the union of all common content associated with the object definitions participating in the document definition.

Note that the concept 'document definition' is different from 'object definition'. A document definition includes object definitions for objects of type 'document' as well as object definitions for any subordinate logical and layout objects.

3.6 Document Profile

The document profile contains information for handling the document as a whole; it consists of a set of attributes at the highest level in the document structure.

The document profile specifies the conformance combination used in the document and specifies which of the following are present in the document:

- layout structure;
- logical structure;
- layout definitions;
- logical definitions.

It may describe the document and its history, including information for editing, layout, presentation, indexing, filing and retrieval. For delivery control the document profile may duplicate information usually found in the document content (e.g. document name, date).

It also includes general information required for interchange (e.g. title, author) but excludes data specific to a particular mode of transmission, such as mail, message, or Teletex.

4. DOCUMENT ARCHITECTURE

4.1 Document Structure

4.1.1 Structures

The logical structure and the layout structure are hierarchies of objects with associated attributes.

The object at the highest hierarchical level of each structure is called the document object.

The minimum number of hierarchical levels below the level of document is one. The actual number of levels in a given document is variable.

The hierarchical relationship is expressed by the attribute "references to subordinate objects". This attribute identifies the immediately subordinate objects.

A complete sequential order is defined between the objects of a structure, as follows:

i) The first object is the unique object without superiors, the document object.

ii) The ordering between any set of objects that share the same immediate superior object is that defined by the order of appearance of numbers corresponding to them in the attribute "references to subordinate objects".

iii) Any object with subordinate objects precedes all of its subordinate objects.

iv) The ordering defined in ii) applies also between all subordinates of such objects.

The sequential order defined by the layout structure is termed the sequential layout order; that defined by the logical structure is termed the sequential logical order.

Any object that has inferiors in the structure is a composite object.

Basic objects are objects at the lowest hierarchical level and do not have the attribute "references to subordinate objects".

The content of a basic object is limited to a single content architecture.

4.1.2 Relationships

In addition to the hierarchical relationships given by the attribute "references to subordinate objects" other relationships can be defined between objects and between objects and definitions:

- relationships between objects of the logical structure are called logical-logical relationships;
- relationships between objects of the layout structure are called layout-layout relationships;
- relationships between objects of the logical structure and layout definitions are called logical - layout relationships.

A relationship is expressed by attributes of objects that reference one or more other objects or definitions.

A relationship can be bi-directional, in which case each object has a relationship attribute referring to the other object.

Relationships can be derived from relation rules specified in the object definitions.

4.1.3 Object Types and Object Classes

Each object has a certain type. This is expressed by the attribute "object type" associated with each object.

The attribute "object type" determines which attributes and attribute values are applicable to that object.

An object may belong to a certain object class. This is expressed by the attribute "reference to object definition" associated with the object. Which attributes are applicable to the objects of a certain class is determined by the object type specified in the object definition.

4.2 Layout Structure

The layout structure is illustrated in Figure 3.

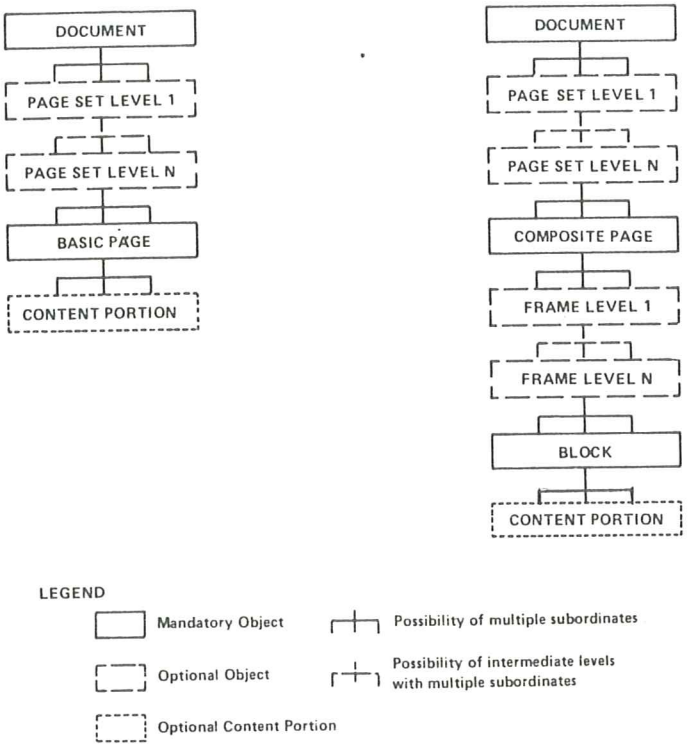


Figure 3: Layout Structure

4.2.1 Layout Objects

The objects at successively lower hierarchical levels of the layout structure are of object types:

- document;
- page set;
- page;
- frame;
- block.

A basic layout object is either a page or a block.

4.2.1.1 Document

The document is the highest-level object in the hierarchy of the layout structure.

4.2.1.2 Page Set

A page set is a composite layout object that contains one or more pages and/or one or more subordinate page sets that need to be identified as a group, e.g. the pages containing a section of a document.

There may be zero, one or more levels of page set between the levels of document and page.

4.2.1.3 Page

A page is a rectangular area that corresponds to a unit of the presentation medium. The page is the reference area used for positioning and imaging the content of the document. The size of a page may be smaller than, equal to or greater than the size of the corresponding unit of the presentation medium.

A page is immediately subordinate either to the document or to a page set. A page may be a basic layout object, in which case it is termed a basic page. If a page is a composite layout object, in which case it is termed a composite page, then below the level of page is the level of frame or block.

4.2.1.4 Frame

A frame is a rectangular area within a page or within a frame of a hierarchically higher level, with its sides parallel to the sides of the page. Frames define boundaries for the layout of content within a page and a frame may intersect partially or fully with other frames and blocks.

A frame is a composite layout object that contains one or more subordinate frames and/or one or more blocks.

There may be zero, one or more levels of frame below the level of a composite page.

4.2.1.5 Block

A block is a rectangular area with its sides parallel to the sides of the page. It is a container for portions of the document content.

A block is a basic layout object.

Blocks may intersect partially or fully with other blocks and frames, independent of the content architecture of the content portions to be imaged in the block.

4.2.2 Positioning Principles for Layout Objects

Each layout object below and including the level of page has an orthogonal coordinate system defined.

The origin of each object's coordinate system is at its top left-hand corner, and the horizontal axis coincides with its top edge. As a consequence, the vertical axis coincides with the left edge of the object.

Horizontal positions are measured positively from the vertical axis to the right, and vertical positions are measured positively from the horizontal axis downwards.

The position and dimensions of all objects subordinate to a page are specified in relationship to the coordinate system of their immediately superior object.

The reference point for positioning is the top left corner of each frame or block. All frames and blocks are positioned relative to the coordinate system of their immediately superior object and are contained entirely within the area of that object.

Thus, the objects immediately below the level of page are positioned in absolute page coordinates, while all objects subordinate to that level use relative positioning.

The coordinate system of each basic object is provided for use by its content architecture.

4.2.3 Intersection Principles

Within a page, frames and blocks may be positioned in such a way that they intersect partially or fully, i.e. they share common areas.

A page or a frame can be considered as a sheet which carries within its surface area other sheets representing its immediately subordinate objects which may be frames or blocks. Similarly a block can be considered as a sheet, which in case of character content carries small sheets representing character boxes.

The surface of a sheet has a certain texture. By analogy layout objects have a property called background texture. In this Standard, the background texture of frames and blocks is restricted to 'white opaque' or 'colourless transparent' , and the background texture of pages is always 'white opaque'.

The intersection is governed by the following rules:

- if frames and/or blocks intersect, their overlay sequence is given by the attribute "imaging order" of their common superior object, where layout objects later in the sequence are said to overlay objects earlier in the sequence;

- if a white opaque layout object overlays another layout object, any content or background texture of the

underlying objects and their inferiors become invisible in the area of intersection;

- if a colourless transparent layout object overlays another layout object their contents in the common areas are combined.

Overlay order is illustrated in Figure 4.

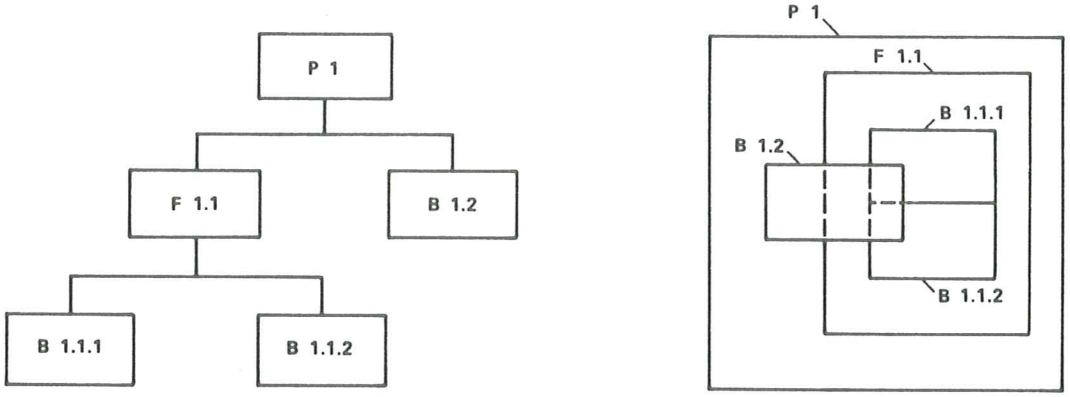


Figure 4: Overlay Order

4.3 Logical Structure

The logical structure is illustrated in Figure 5.

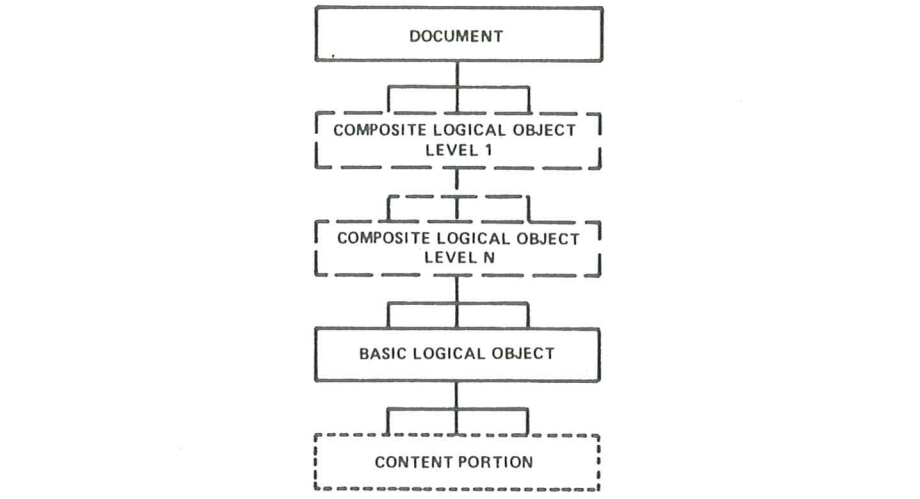


Figure 5: Logical Structure

The logical objects at successively lower levels of the logical hierarchy are of object types:

- document;
- composite logical object;
- basic logical object.

The Standard does not define standard object classes but rather provides means for allowing them to be determined by the applications.

The logical structure consists of the document, which is the highest object in the hierarchy, and at least one basic logical object.

The document contains as its immediate subordinates zero, one or more basic logical objects and/or zero, one or more composite logical objects.

A composite logical object contains as its immediate subordinates zero, one or more basic logical objects and/or zero, one or more composite logical objects.

A basic logical object contains no subordinate objects.

4.4 Object Definitions

4.4.1 Object Class

An object definition defines an object class.

The object class is identified by the attribute "definition identifier". All objects of the class concerned have an attribute "reference to object definition" which takes the value of this definition identifier.

The object type is specified by an attribute and is applicable to all objects of the class concerned.

Other attributes of an object definition specify rules to determine values for the corresponding attributes of the objects of the class defined.

Rules can be classified into:

- property rules;
- relation rules;
- content rules;
- construction rules.

These may be specified in a number of forms, including constant or variable expressions.

If an attribute value is specified explicitly for an object then this overrides any value that may be derived from a rule.

4.4.1.1 Property Rules

Property rules specify values of attributes of objects other than relationships.

4.4.1.2 Relation Rules

Relation rules specify how relationships between objects may be generated.

A relation rule refers to an object class and determines that the corresponding object attribute refers to an object of that class.

Relation rules are divided into:

- logical-logical relation rules which yield relationships between logical objects;
- layout-layout relation rules which yield relationships between layout objects;
- logical-layout relation rules which yield relationships between logical objects and layout objects.

Thus logical-layout relation rules control the placement of the content of logical objects into layout objects when a document is laid out and together with associated attributes specify how the layout is to express the logical structure on the presentation media.

4.4.1.3 Content Rules

The definition of a class of basic objects may specify content either by referring to one or more common content portions or by supplying a content generator, i.e. an expression from which content may be derived.

Examples of common content portions are logos and predefined paragraphs.

Examples of content generators are expressions to produce page numbers or section numbers.

4.4.1.4 Construction Rules

The definition of a class of composite objects may specify by means of a construction rule:

- the object classes to which the immediately subordinate objects belong, and
- how these subordinate objects are ordered.

A rule may define recursive constructions, repetitive constructions, alternative constructions and optional constructions.

4.4.2 Common Content

When the object definition of a class of basic layout objects specifies common content, each object of that class refers to the definition and does not refer to a content portion unless the common content specification is overriden by a content specification which is unique for that object.

When such a basic layout object is imaged, either the common content specified by the object definition or the object's unique content is used by the imaging process.

When the object definition of a class of basic logical objects specifies common content, each object of that class refers to the definition and does not refer to a content portion unless the common content specification is overriden by a content specification which is unique for that object.

When such a basic logical object is formatted, each resulting basic layout object refers to a unique copy of the common content. These copies will, in general, be different from each other because the different layout objects may have different dimensions and different presentation attributes. These content portions are referred to only by the basic layout objects, not by the basic logical objects or their definitions.

4.5 Document Definition

A document definition specifies a document class. It describes how to construct the logical and layout structures of documents that are members of that class.

In order to be able to use a document definition to control production of the whole structure of a document it has to contain the object definitions of all logical and layout object classes which are allowed to occur in a document of that class.

A document definition consists of a complete set of object definitions for objects from the document level down to the basic object levels.

4.6 Content Portions

The content of a document is divided into content portions in such a way that each content portion is associated with at most one basic layout object and with at most one basic logical object.

Consequently, the content associated with each object, basic or composite, comprises an integral number of content portions.

A single basic object may refer to several content portions. For example, a basic logical object representing a paragraph would need to contain at least two content portions in the case where the paragraph is split across page boundaries. This is illustrated in Figure 6.

If there is only one structure in a document, there need be only one content portion corresponding to each basic object.

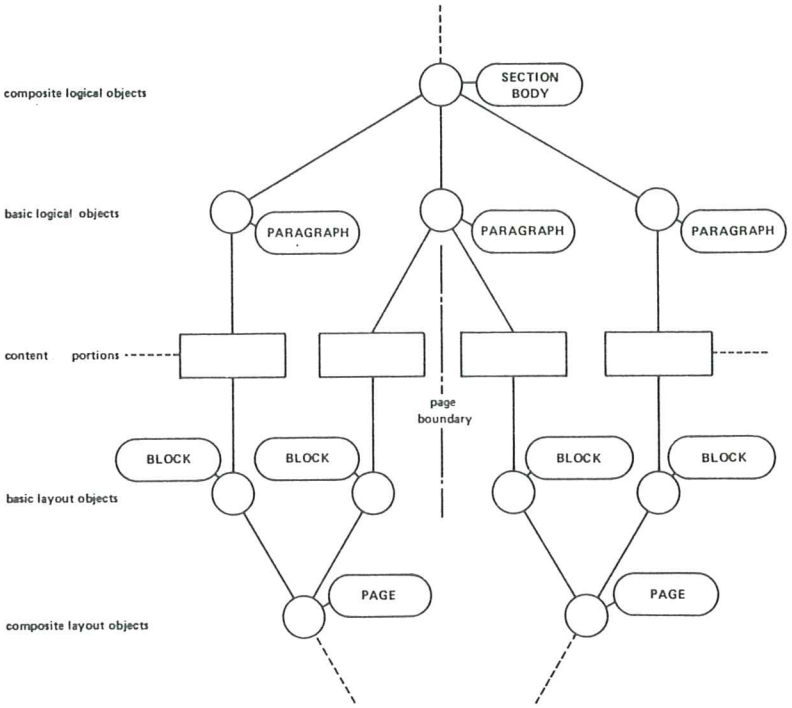


Figure 6: Correspondence between Logical Objects, Layout Objects and Content Portions

5. DOCUMENT PROCESSING MODEL

5.1 Introduction

This section describes a conceptual model for processing a document and describes the principal operations performed on a document as a basis for understanding the semantics of the attributes defined in this Standard.

This section is not intended to represent an actual implementation, nor to restrict in any way the processing that may be applied to an interchanged document.

The document processing model defines three processes:

- editing;
- layout;
- presentation.

The document processing model is illustrated in Figure 7. The order of processes in Figure 7 is not intended to imply that they are performed sequentially in an actual implementation.

The editing process consists of content editing which creates new content or modifies previous content, and logical structure editing which creates a logical structure or modifies a previous logical structure and allocates content to the basic logical

objects. Modifications to the logical structure are checked for comformity with the rules specified in the logical definitions.

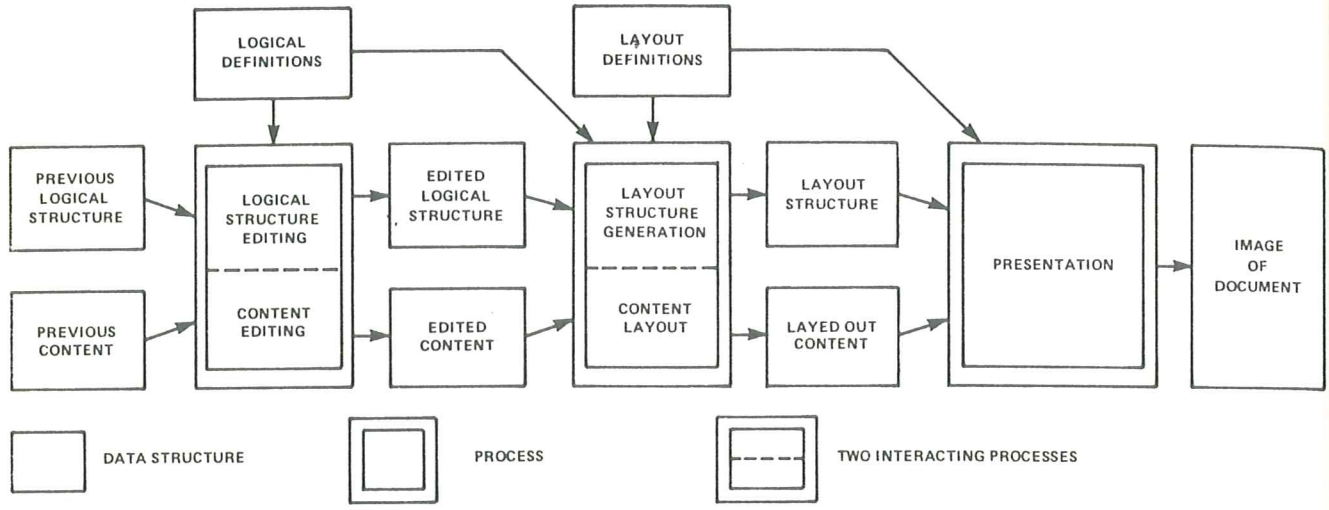


Figure 7: Document Processing Model

The layout process consists of creating a layout structure and allocating the content to the basic layout objects without changing its allocation to logical objects. These two activities are highly inter-related. The layout process is controlled by the information contained in the logical structure, the content, the layout definitions and the logical-layout relation rules.

The presentation process consists of using the layout structure, the layout definitions and the content to present an image of the document.

The model provides for manual intervention only in the editing processes on the logical structure and the content. Manual intervention can occur at many stages in the model, but is neither explicitly allowed nor prohibited by the Standard.

Manual intervention other than that provided for by the model may produce an inconsistent document.

It is also possible to edit the logical and layout definitions themselves, but this is not included in this model, since the Standard does not place any constraints on such editing.

5.2 Roles of the Document Architecture Components

This subsection describes the roles of the various parts of the document architecture during the editing, layout and presentation processes.

5.2.1 Logical Definitions

If a set of logical definitions is present in a document for the purpose of controlling editing then every logical object

must refer to a logical definition, either explicitly or by use of the defaulting rules.

In this case, changes to the logical structure are checked to ensure that they are consistent with the logical definitions, in particular with the construction rules. In addition, when a new logical object is created, a logical definition serves as a template for the logical object.

During the layout process, the logical definitions may provide values of property attributes, relation attributes and common content.

5.2.2 Logical Structure

The logical structure is the target of the structure editing process. The logical structure is changed by:

- creating or deleting a logical object in the structure;
- changing the position of a logical object in the structure;
- changing or deleting the value of a property or relation attribute that is defined for a logical object, either explicitly or in a default value list;
- adding an explicit value for a property or relation attribute;
- adding a default value for a property or relation attribute to a default value list for a logical object.

During the layout process, the sequential logical order of the objects in the logical structure determines the sequence in which the content of the document is processed. The layout directives and presentation attributes associated with each logical object are used to control the layout of the content associated with that object.

The logical structure plays no role in the presentation process.

5.2.3 Content

The content is the target of the content editing process. The content is changed by:

- adding or deleting one or more graphic elements;
- adding, modifying or deleting embedded control functions.

The editing algorithms used to change the content are not described in this model as they are outside the scope of this Standard.

During the layout process, the content is allocated to basic layout objects. The division of the content into content

portions may be modified so that it is consistent with both structures. In addition, the layout process may insert control functions into the content to facilitate the presentation process.

In the presentation process, the content is used to produce an image of the document.

5.2.4 Layout Definitions

The layout definitions play no role in the editing processes.

The construction rules associated with the layout definitions determine all permissible layout structures which may be created by the layout process. Which of these permissible structures is used is determined from the logical structure, the content, the layout directives, and the logical-layout relation rules in the logical definitions. This process is described in 5.3.

When a layout object is created using a layout definition, that definition serves as a template for the layout object. The property rules contained in the layout definitions may direct the layout of the content within basic layout objects.

During the presentation process, the layout definitions may provide, for the associated layout object, any combination of the following:

- attributes that direct the presentation of the content;
- the content.

5.2.5 Layout Structure

The layout structure does not play a role in the editing process.

The document processing model does not provide for direct modification of the layout structure. This Standard does not prohibit such editing by the user, but this may produce a layout structure which is inconsistent with the logical structure and the layout definitions.

When a layout structure is created, the layout process uses the information contained in the logical structure, the content, the layout definitions and the logical-layout relation rules. This process is described in 5.3.

5.3 The Layout Process

This section provides an informal description of a layout process compatible with this Standard and applicable to documents containing logical structure, layout definitions and, optionally, logical definitions.

The simplest case for layout is first described, and this is enhanced through the following sub-sections, which describe increasingly sophisticated facilities for layout.

Throughout this section, the term layout process is used to mean the generation of layout structure only, excluding the content layout process.

5.3.1 Layout References

The attribute "layout reference" represents a relationship between a logical object and the definition of a layout object of type document, page set or page. It indicates that the entire logical sub-structure rooted at this object must be laid out within a single instance of a layout object derived from the referenced definition, and that no other part of the document may be laid out within the same layout object.

5.3.2 The Simplest Case

Consider a document with the following restrictions imposed:

- the following attributes do not occur in the document:
  * required category,
  * permitted categories,
  * indivisible,
  * new layout object;
- the following attributes either do not occur in the document, or occur only with the standard default value:
  * object path,
  * separation;
- the attribute "concatenation" is used only with the value concatenate;
- the attribute "layout reference" is used exactly once, on the object which is the root of the logical structure;
- in all layout definitions, the attribute "generator for subordinate objects" defines a sequence and makes no use of choices, options or repetitions;
- one content architecture is used in the document.

For layout purposes, such a document conceptually reduces to a document with a single basic logical object referring to a single content portion, since there are no logical breaks in the content imposed by the document architecture. The document contains a single text stream. Note that, in reality, the logical structure may be subdivided for various reasons, such as differing presentation, but that this has no effect on the layout process.

In addition, the document has fixed layout, since it is only possible to derive a single layout structure from the layout definitions. Therefore, in theory, all that is required of the layout process is to generate this layout structure and then allocate the content to the basic layout objects in their sequential order.

In practice, this is achieved as follows. Starting with the layout definition referenced from the root of the logical structure, the corresponding layout object is created, and the first element of the generator for subordinate objects is evaluated. This process is repeated until a definition is reached which has no generator for subordinate objects. The object created from this definition must be a frame which is to contain blocks of content.

Since, in the case under consideration, there are no logical breaks in the content, there can only be a single block within each lowest level frame. The tentative position and dimensions of this block are determined from those of the frame, taking into account any constraints imposed by the attribute "offset". The content layout process then determines the actual position and dimensions and allocates content to the block according to the particular content architecture.

If there is still content which remains to be laid out at this point, the next block to be created must be determined. This is achieved by using the first un-evaluated element of a generator for subordinate objects encountered on following the layout structure towards its root, examining each definition, and then repeating the procedure described above.

This process is repeated until no content remains to be laid out, and the layout structure is fully constructed. It is considered an error if there is any content, or there are any unevaluated elements of a generator for subordinate objects, remaining in the document at this point.

From the above, it is clear that the layout structure is created in depth-first order. Figure 8 shows a simple layout structure (excluding blocks) which might be created by this process, indicating the order in which the objects were created.

If the end of a frame is reached inside a content portion of a basic logical object, the content portion is split into two content portions belonging to the basic logical object. The attribute "references to content portions" is modified so as to refer to the additional content portion in the correct sequential order. The new content portion receives values for appropriate attributes.

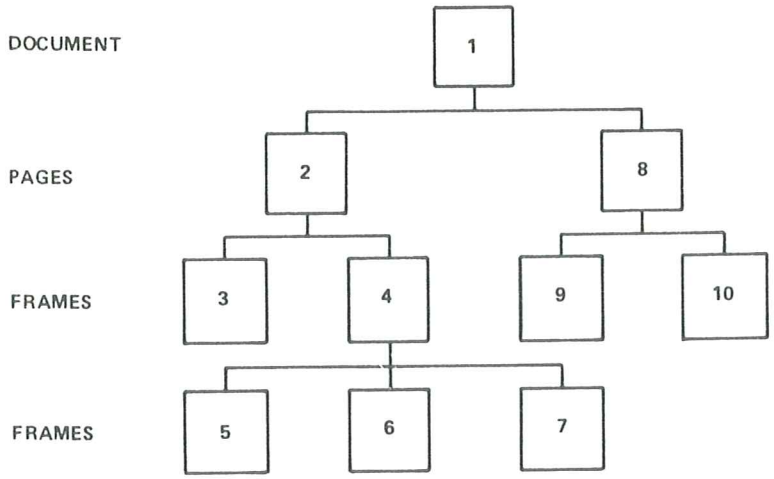


Figure 8: Order of Creation of Composite Layout Objects

5.3.3 Multiple Blocks per Frame

If a logical object specifies that it should not be concatenated with the preceding logical object, this causes the creation of a new block, which may occur within the same frame as the preceding block.

In this case the attributes "object path" and "separation" may specify additional constraints which must be satisfied in the determination of tentative position and dimensions for the block.

The separation between the preceding block and the new block is the maximum of the leading separation of the preceding logical object, or if it is the last in a concatentated sequence the leading separation specified in the first object of that sequence, and the trailing separation of the first logical object to be laid out in the new block.

5.3.4 Non-fixed Layout

If a layout definition has a "generator for subordinate objects" which defines an aggregate or makes use of choices, options or repetitions, then, potentially, several different layout structures could be created using this definition.

The evaluation of such an expression is performed according to the description in 5.3.7. If all alternatives in the expression fail, then a re-evaluation of an alternative construction expression at the nearest higher level is attempted.

All failed constructions are discarded.

5.3.5 Multiple Layout References

If the restriction on the usage of the attribute "layout reference" is removed, then three possible cases can arise,

depending upon the logical structure.

i) The existence of a single path from the root of the logical structure to a basic logical object, along which no occurrence of the attribute "layout reference" is found creates an error condition, since there is no information provided as to where and how this path should be laid out.

ii) If, for every path from the root of the logical structure to a basic logical object, there is exactly one occurrence of the attribute "layout reference", then the logical structure can conceptually be divided into a number of sub-structures, each of which is rooted at a logical object for which a layout reference is specified, and each of which can be laid out independently, as before.

iii) When a single path, from the root of the logical structure to a basic logical object, contains more than one occurrence of the attribute "layout reference", then each layout reference imposes successively greater constraints on the layout of the sub-structure rooted at the object on which it occurs.

A secondary layout reference (i.e. not the first on the path) indicates that, in addition to the logical object being laid out entirely within a layout object of the specified class, that layout object must be within the layout object created from the preceding layout reference on the same path. Furthermore, the layout sub-structure so created must be in accordance with the generators for subordinate objects along the path.

5.3.6 Layout Categories

A layout category is a name which may be associated with logical objects and layout objects in order to restrict the layout objects into which the logical objects may be placed. The required category specified on a logical object must match one of the permitted categories of the layout object in which it is laid out. A layout object of a class which does not specify a set of permitted categories may have associated with it logical objects of any layout category.

When the attribute "required category" is specified in the logical structure, there are two possible cases to consider:

i) In the simpler case, exactly one required category is specified in the path from the root of the logical sub-structure to a basic logical object. When the object specifying the required category is encountered during the depth-first processing of the logical sub-structure, then the current layout process is suspended, and a new layout process is initiated such that the logical sub-structure rooted at this object is laid out only within frames whose definitions permit the required category, and starting with the next

available such frame within the layout sub-structure corresponding to the lowest occurrence of layout reference.

In the event that the first such frame would appear on a new page, or that this layout process requires the creation of a new page, the process is suspended. Previously suspended layout processes which may further lay out the current page are resumed in the sequential logical order of their sub-structure, and this continues until all layout processes relating to this layout sub-structure either have terminated or require the creation of a new page.

If a new page is then required, it is created according to the relevant "generator for subordinate objects" attribute, and the layout process continues.

ii) In the remaining case, more than one required category is specified in the path through the logical sub-structure. Those logical objects specifying secondary required categories (i.e. not first on the path) can be considered as the roots of nested logical sub-structures. When the layout process encounters such an object, then this layout process is suspended pending either termination or resumption of the layout process for the nested sub-structure, in the same manner described above.

Secondary required categories are not cumulative. That is, at the point at which a secondary required category occurs, only that category is required for the sub-structure rooted there. Exactly one of the permitted categories of a layout object is satisfied.

5.3.7 Evaluation of Construction Expressions

The evaluation of a sequence construction expression occurs in left to right order, resulting in subsequent construction terms. If the layout process fails for any of these terms, then the corresponding partial layout is discarded and a new layout object condition is raised. This causes re-evaluation of a construction term at a higher level in order to create a new layout sequence.

For a choice constructor, one alternative is evaluated at a time, and subsequent choices are not evaluated unless required. The order of selection of choices is from left to right. If the layout process for a given choice fails, then the layout is retried using the next alternative term. If there are no remaining alternatives, then the layout fails at this point.

An aggregate construction expression may be evaluated in any order, since no restrictions are imposed.

For an optional construction factor, the option is included on first evaluation. If the subsequent layout in this layout

object fails, then that partial layout is discarded and the layout is retried as if the option was not present.

A repetitive construction factor causes, if the layout for a construction term succeeds, generation of a further identical construction term. If the layout fails, the last term in the repetition is discarded and the next construction term is tried.

5.3.8 Determination of Presentation Attributes

When a layout structure is generated by a layout process on the basis of a logical structure, logical definitions and layout definitions, and different values for the same presentation attributes of any layout object are derived from the layout definition and the logical objects and logical definitions, the attribute value from the logical objects and logical definitions should take precedence.

6. ATTRIBUTE DEFINITIONS

6.1 Introduction

The term component is used throughout section 6 to mean an object or an object definition.

The term may be prefixed with the following qualifying terms: basic, composite, page set, page, frame, block, logical and layout.

For example 'frame component' means an object of type frame, or an object definition for an object of type frame.

6.1.1 Attributes

Attributes are parameters of components, content portions or the document profile specifying their characteristics.

The value specification of an attribute may contain:

- a numeric constant;
- a character string;
- a reference to another component;
- an expression to calculate the value, which may depend on attributes of other components.

The conventions governing determination of attribute values are specified in 6.2 and 6.3.

6.1.2 Attribute Types

There are five types of attributes, namely:

- identification attributes, which identify components and content portions and specify the types of objects and content portions;

- property attributes, which specify the characteristics of objects and content portions;

- presentation attributes which are dependent on content type and specify how the content associated with objects is to be imaged;

- construction attributes, which specify:

  * the hierarchical relationships between objects,
  * the relationships between objects and object definitions,
  * the relationships between components and content portions;

- relation attributes, which specify:

  * logical-logical relationships,
  * layout-layout relationships,
  * logical-layout relationships.

6.1.3 Attribute Classification

In this section of the Standard attributes are classified as follows.

i) Common Attributes (see 6.5)

These consist of the identification and construction attributes applicable to components.

ii) Layout Attributes (see 6.6)

These consist of the property and relation attributes for layout components.

iii) Logical Attributes (see 6.7)

These consist of the property and relation attributes for logical components.

iv) Content Attributes (see 6.8)

These consist of the identification and property attributes for content portions.

v) Presentation Attributes (see 6.9)

These consist of the presentation attributes for basic components.

vi) Document Profile Attributes (see 6.10)

These consist of attributes specifying general characteristics of the document.

6.1.4 Summary of Attributes of Components

Table 1 is a summary of the defined attributes of components. These are listed under the classifications i) to v) given in 6.1.3.

6.2 Determination of Values of Attributes of Objects

The value of an attribute for an object is determined by the first of the following which is applicable:

i) the value may be specified explicitly for the object to which the attribute applies;

ii) if the object concerned refers to an object definition and a rule for the attribute concerned is specified explicitly in that object definition, then the attribute of the object derives its value from that rule;

iii) certain attributes, for example presentation attributes, can have values specified by the "default value lists" attribute of objects or their corresponding object definition, if any, at higher levels of the hierarchy;

if more than one default value list specifies a value for the same attribute, then the value derived from the lowest hierarchical level in the structure is used;

at each level, the attribute "default value lists" can only be taken from the definition if this attribute is not specified for the object;

iv) the default value defined in this Standard;

v) no value is determined - the attribute is not defined.

Attributes which must have a value specified by steps i) and ii) above are termed mandatory and it is an error if no value is specified.

Attributes which have an effect defined for the object to which they apply only when they have a value specified by steps i) to iii) above are termed non-mandatory.

Attributes that have a default value defined in this Standard and therefore always have a defined value derived by steps i) to iv) above are termed defaultable.

A part of the definition of each attribute of a component is a statement of the types of component for which the attribute may be specified, and whether the attribute is mandatory, non-mandatory or defaultable. If the attribute is defaultable, the default value specified by this Standard is indicated.

Section	Attribute	Qual.	May be specified for	Default
6.5	**COMMON ATTRIBUTES**			
6.5.1	Object Type	M	All Components	
6.5.2.1	Object Identifier	M	All Objects	
6.5.2.2	Definition Identifier	M	All Definitions	
6.5.3	User-Readable Comments	NM	All Components	
6.5.4	Reference to Object Definition	NM	All Objects	
6.5.5.1	References to Subordinate Objects	M	All Composite Objects	
6.5.5.2	Generator for Subordinate Objects	NM	All Composite Definitions	
6.5.6	References to Content Portions	NM	Basic Components	
6.5.7	Content Generator	NM	Basic Definitions	
6.5.8	Bindings	NM	All Components	
6.5.9	Default Value Lists	NM	All Composite Components	
6.6	**LAYOUT ATTRIBUTES**			
6.6.1	**Property Attributes**			
6.6.1.1	Position	D	Frame, Block Components	Zero Offsets
6.6.1.2	Dimensions	D	Page, Frame, Block Components	(see text)
6.6.1.3	Object path	D	Lowest Level Frame Components	0°
6.6.1.4	Background texture	D	Frame, Block Components	Colourless Transparent
6.6.2	**Relational Attributes**			
6.6.2.1	Permitted Categories	NM	Frame Definitions	
6.6.2.2	Imaging Order	NM	All Composite Layout Objects	
6.7	**LOGICAL ATTRIBUTES**			
6.7.1	**Layout Directives**			
6.7.1.1	Indivisible	D	All Logical Components	No
6.7.1.2	Separation	D	All Basic Logical Components	0,0
6.7.1.3	Offset	D	All Basic Logical Components	Zero Offsets
6.7.1.4	Concatenation	D	All Basic Logical Components	(see text)
6.7.1.5	New Layout Object	D	All Logical Components	No
6.7.1.6	Layout Reference	NM	All Logical Components	
6.7.1.7	Required Category	NM	All Logical Components	
6.7.2	**Property Attributes**			
6.7.2.1	Protected	D	All Logical Components	Unprotected
6.8	**ATTRIBUTES OF CONTENT PORTIONS**			
6.8.1	Content Portion Identifier	M	Content Portion	
6.8.2	Alternative Graphic Representation	NM	Content Portion	
6.8.3	Type of Coding	*	Content Portion	*
6.8.4	Coding Attributes	*	Content Portion	*
6.9	**PRESENTATION ATTRIBUTES**			
6.9.1	Content Type	D	All Basic Components	Character content
6.9.2	Content Architecture	D	All Basic Components	(see text)

The attributes are qualified as described in 6.2. This is indicated in the 'Qual.' column as follows:

M Mandatory
NM Non-Mandatory
D Defaultable (of objects) or Non-mandatory (of object definitions)

* Conformance combination and content architecture dependent

Table 1 : Summary of Attribute Definitions

6.3 Determination of Values of Attributes of Object Definitions

Generally the value of any attribute of an object definition expresses a rule to determine, for any object of the class defined, the valid values of the corresponding attribute. Where the value is a constant the rule is implicitly to copy the value into the attribute of the object.

Attributes of object definitions which must have a value specified are designated mandatory, other attributes are designated non-mandatory.

6.4 Expressions within Attribute Values

Within an object definition, an attribute value is in general specified by an expression. These expressions are defined in this section.

6.4.1 Object Identifier Expressions

Within an object definition, a reference to an object is represented by an object identifier expression. This expression identifies the referenced object, if any, by specifying a hierarchical position within the structure relative to the position of any object of the class.

An object identifier expression names an object selection function and provides one or more parameters.

Object selection functions are:

i) CURRENT-ID, producing the identifier of the object for which the attribute is being evaluated; this function has no parameters;

ii) SUPERIOR, producing the identifier of the object that is immediately superior to a selected object; this function has one parameter, consisting of an object identifier expression;

iii) PRECEDING-ID, producing the identifier that precedes the identifier of the specified object in the list of subordinate objects which is kept at its superior object. If the specified object is the first page in a page set, but not the very first page of the document, the identifier of the preceding page in the document is returned; this function has one parameter consisting of an object identifier expression;

iv) SUCCEEDING-ID, producing the identifier that follows the identifier of the specified object in the list of subordinate objects which is kept at its superior object; this function has one parameter consisting of an object identifier expression.

Any object selection function will produce a null identifier when an attempt is made to reference a non-existent object (for example, the page preceding the first page of the document).

6.4.2 References to Binding Values

Within the value specification of an attribute, it is possible to refer to the value of a "bindings" attribute of either the object being generated or a different object. Such a reference is represented by the combination of an object identifier expression (see 6.4.1) and a binding name (see 6.5.8).

6.4.3 String Expressions

A string expression within an attribute value specification consists of either an atomic string expression or the concatenation of two or more atomic string expressions.

An atomic string expression is one of the following:

- a string literal;
- a binding reference (see 6.4.2);
- a string function application(see below).

A string literal is an arbitrary character string.

A string function application is an application of one of the functions described below. Each of these functions has a parameter consisting of a cardinal expression (see 6.4.4).

i) MAKE-STRING
This function produces a character string consisting of the decimal representation of the numeric value of the parameter.

ii) UPPER-ALPHA
This function produces a character string consisting of one of the capital letters A to Z, corresponding to the numeric value (1 to 26) of the parameter.

iii) LOWER-ALPHA
This function produces a character string consisting of one of the small letters a to z, corresponding to the numeric value (1 to 26) of the parameter.

iv) UPPER-ROMAN
This function produces a character string consisting of the Roman numeral representing the numeric value of the parameter, using the capital letters:
C, D, I, L, M, V and X.

v) LOWER-ROMAN
This function produces a character string consisting of the Roman numeral representing the numeric value of the parameter, using the small letters:
c, d, i, l, m, v and x.

A string function produces an empty string when the value of the parameter is zero or negative.

6.4.4 Cardinal Expressions

A cardinal expression within an attribute value specification consists of one of the following:

- a cardinal literal;
- a reference to a binding value (see 6.4.2);
- a cardinal function application (see below).

A cardinal literal is any positive integer.

A cardinal function application is an application of one of the functions described below. Each of these functions has a parameter consisting of a cardinal expression.

i) INCREMENT
This function produces a numeric value which is one greater than the value of the parameter.

ii) DECREMENT
This function produces a numeric value which is one less than the value of the parameter.

6.5 Common Attributes

The common attributes defined in this section are applicable to both logical components and layout components. They are not applicable to content portions.

6.5.1 Object Type (ALL COMPONENTS)

Mandatory.

This attribute specifies the object type. The type determines the attributes that are applicable to the object. In the case of a layout object, it specifies whether the type is:

- document;
- page set;
- page;
- frame;
- block.

In the case of a logical object, it specifies whether it is:

- document;
- composite logical object;
- basic logical object.

In the case of an object definition, it specifies the type of the object being defined.

6.5.2 Object / Definition Identification

6.5.2.1 Object Identifier (ALL OBJECTS)

Mandatory (except in certain conformance combinations).

This attribute identifies an object uniquely within the context of the document.

An object identifier consists of a sequence of numbers. Each number in the sequence corresponds to a hierarchical level of the layout structure or logical structure and identifies one particular object at that level.

The first number in the sequence indicates whether the identifier pertains to a layout object or a logical object. The values assigned to this first number are:

- 1 layout object;
- 3 logical object.

6.5.2.2 Definition Identifier (ALL DEFINITIONS)

Mandatory.

This attribute identifies a definition uniquely within the context of the document.

A definition identifier consists of a sequence of numbers.

The first number in the sequence indicates whether the identifier pertains to a layout definition or a logical definition. The values assigned to this first number are:

- 0 layout definition;
- 2 logical definition.

The other numbers are allocated by the application.

6.5.3 User-Readable Comments (ALL COMPONENTS)

Non-Mandatory.

This attribute consists of a character string that is to be interpreted as comments relevant to that component or to the associated content portions. This character sequence is not part of the document content.

The character set to be used in this attribute is that specified in the Document Profile attribute "user-readable comments graphic character set"

The default character set is the minimum subrepertoire of ISO 6937, plus carriage return and line feed.

6.5.4 Reference to Object Definition (ALL OBJECTS)

Non-Mandatory.

This attribute is used to establish a relationship between an object and its object definition.

The value of this attribute is the identifier of the corresponding object definition (see 6.5.2).

6.5.5 References to Subordinates

6.5.5.1 References to Subordinate Objects (ALL COMPOSITE OBJECTS)

Mandatory (except in certain conformance combinations).

The value of this attribute is a sequence of one or more numbers. Each number corresponds to an immediately subordinate object and consists of the last number in the identifier of that subordinate object (see 6.5.2). A given number may not occur more than once in the sequence.

The order of appearance of the numbers in the sequence defines a sequential order among the subordinate objects.

In both logical and layout objects the sequential order is interpreted as determining the order in which the objects are handled by the layout process. For layout objects, sequential order represents the order in which objects are to be overlaid, unless overridden by the "imaging order" attribute.

6.5.5.2 Generator for Subordinate Objects (ALL COMPOSITE DEFINITIONS)

Non-Mandatory.

This attribute consists of an expression that guides and restricts the generation of the attribute "references to subordinate objects" of a composite object.

The value of this attribute is either a simple construction or a construction expression.

A simple construction is a sequence of one or more numbers. Each of these numbers refers to an object definition that is to be used to generate a subordinate object of the object being generated. The identifier of the object definition referred to consists of the identifier of the current object definition, extended with the number concerned.

A construction expression specifies either a sequence of definition identifiers, or two or more alternative sequences of definition identifiers of object definitions that are to be used to generate subordinate objects of the object being generated.

A construction expression is one of the following:

- a sequence construction;
- an aggregate construction;
- a choice construction.

A sequence construction consists of one or more construction terms (see below), which are to be evaluated in the order specified.

An aggregate construction consists of one or more construction terms (see below), which are to be evaluated in an arbitrary order.

A choice construction consists of one or more construction terms (see below), one of which is to be evaluated.

A construction term is one of the following:

- a required construction factor;
- an optional construction factor;
- a repetitive construction factor;
- an optional repetitive construction factor.

A construction factor is either a definition identifier or a construction expression. In the former case, the definition identifier is the value of the construction factor. In the latter case, evaluation of the construction factor produces either an empty sequence or a sequence of one or more definition identifiers and this sequence is the value of the construction factor.

A required construction factor is to be evaluated once when the containing construction term is evaluated.

An optional construction factor may be evaluated once or may not be evaluated, when the containing term is evaluated.

A repetitive construction factor is to be evaluated one or more times in succession when the containing construction term is evaluated.

An optional repetitive construction factor may be evaluated one or more times in succession, or may not be evaluated, when the containing construction term is evaluated.

6.5.6 References to Content Portions (BASIC COMPONENTS)

Non-Mandatory.

The value of this attribute is a sequence of one or more numbers. Each number corresponds to a content portion of the component concerned and consists of the last number in the identifier of that content portion (see 6.8.1). A given number may not occur more than once in the sequence.

The order of appearance of the numbers in the sequence defines a sequential order among the content portions.

The attribute must be specified for basic objects unless the basic object refers to an object definition which contains a content rule.

6.5.7 Content Generator (BASIC DEFINITIONS)

Non-Mandatory.

This attribute and the attribute "references to content portions" are mutually exclusive.

The value of this attribute is a string expression, which, when evaluated, results in a string constant, this being the content of the resultant object.

String expressions are defined in 6.4.3.

6.5.8 Bindings (ALL COMPONENTS)

Non-Mandatory.

This attribute specifies any application dependent attributes associated with an object or object definition, together with their values.

The value of this attribute is a sequence of pairs, each pair consisting of the name of an attribute together with an expression representing its value. The value of a binding may be of any type.

6.5.9 Default Value Lists (ALL COMPOSITE COMPONENTS)

Non-Mandatory.

This attribute specifies default attribute values for subordinate objects.

The attribute consists of one or more lists, each of which applies to a different subordinate object type.

Attributes that may be included in a default value list include:

- "reference to object definition";
- layout attributes other than "imaging order";
- logical attributes other than "layout reference";
- all presentation attributes for (one or more) content architectures.

Each attribute may only have a single entry in each default value list.

6.6 Layout Attributes

The attributes applicable to layout components are defined in this section.

Only property rules whose expression defines a constant value are provided in this version of the Standard.

The attributes apply to the specified types of components.

6.6.1 Property Attributes

6.6.1.1 Position (FRAME, BLOCK COMPONENTS)

Defaultable: Default: Zero Offsets.

This attribute consists of a pair of coordinates that specify the position of the object relative to the object at the next higher level in the hierarchy (i.e. either the containing page or frame).

The value of this attribute consists of two integers that specify the X and Y coordinate distances in Basic Measurement Units from the origin of the containing object's coordinate system to the reference point of the object to which this attribute applies.

6.6.1.2 Dimensions (PAGE, FRAME, BLOCK COMPONENTS)

Defaultable: Page Default: common image area for A4 and North American Letter;

Frame or Block Default: maximum size that can be achieved for its position within the containing object.

This attribute consists of a pair of dimensions that specify the size of the object.

The value of this attribute consists of two integers that specify the X and Y dimensions of the object in Basic Measurement Units.

6.6.1.3 Object Path (LOWEST LEVEL FRAME COMPONENTS)

Defaultable: Default: 0°.

This attribute specifies the direction of progression of the allocation of basic layout objects to the frame during the layout process, relative to the positive vertical direction of the frame coordinate system.

This attribute can have one of four values: 0°, 90°, 180°, or 270°.

6.6.1.4 Background Texture (FRAME and BLOCK COMPONENTS)

Defaultable: Default: colourless transparent.

This attribute specifies a frame or block to be white opaque or colourless transparent.

This attribute is effective only when blocks and/or frames actually intersect.

When two or more blocks and/or frames intersect then the effect of combination is determined from the sequential layout order (see 4.2.3).

6.6.2 Relational Attributes

6.6.2.1 Permitted Categories (FRAME DEFINITIONS)

Non-Mandatory.

This attribute specifies the layout categories permitted for logical objects which are to be laid out within a layout object derived from this definition.

The value of this attribute is a list of string constants, these being the names of the permitted layout categories.

If the attribute is not specified any categories are permitted.

6.6.2.2 Imaging Order (ALL COMPOSITE LAYOUT OBJECTS)

Non-Mandatory.

This attribute specifies the order in which the layout objects immediately subordinate to this layout object are to be imaged.

The value of this attribute is a sequence of one or more numbers, each of which is the last number in the identifier of an immediately subordinate object. The order in which the numbers occur determines the order in which the subordinates are imaged. Each of these subordinate objects must be referred to exactly once in the sequence.

In the absence of a value for this attribute the layout objects are imaged in the order specified in the attribute "references to subordinate objects".

6.7 Logical Attributes

6.7.1 Layout Directives

Layout directives are attributes of logical components which guide the generation of layout objects and their placement.

Layout directives are characterised as follows:

- they apply to the object as a whole and cannot be changed within the content;
- they are content architecture independent;
- during layout, they affect only the creation of layout objects (see 5.3).

Layout directives are applied subject to their conforming to the layout definitions.

6.7.1.1 Indivisible (ALL LOGICAL COMPONENTS)

Defaultable : Default : No.

This attribute specifies that the content of the logical object should appear within a single layout object of the specified class or category.

The value of this attribute can either be the identifier of a layout definition for a page set or page or the name of a layout category.

If this attribute is specified and the document is such that the required effect cannot be achieved consistently with the attribute "generator for subordinate objects" then the layout process proceeds ignoring the specification of the attribute.

6.7.1.2 Separation (BASIC LOGICAL COMPONENTS)

Defaultable: Default: 0,0.

This attribute specifies minimum amounts of separation between the basic layout object used to present the contents of this logical object and the nearest adjacent basic layout objects. Separation may be specified for the leading edge (i.e. the edge at right angles to the object path and furthest in the direction of the object path) and/or the trailing edge (i.e. the edge opposite to the leading edge).

It is specified in terms of Basic Measurement Units.

6.7.1.3 Offset (BASIC LOGICAL COMPONENTS)

Defaultable: Default: Zero Offsets.

This attribute specifies minimum amounts of offset between the basic layout object used to present the contents of this logical object and the boundary of the containing layout object. Offset may be specified for the top, bottom, left and right edges of the corresponding basic layout object relative to the coordinate system of the containing layout object. It is specified in terms of Basic Measurement Units.

6.7.1.4 Concatenation (BASIC LOGICAL COMPONENTS)

Defaultable: Default: 'Concatenate'.

This attribute specifies how the content portions of the basic logical object are to be laid out in relationship to the content portions of the preceding basic logical object.

The value of this attibute can be either 'concatenate' or 'non-concatenate'.

The value 'concatenate' specifies that the layout of the content portions associated with the object may be continued in the same basic layout object as used with the content portions of the preceding basic logical object.

The value 'concatenate' is only effective in a basic logical object if the preceding basic logical object has the same values of the attributes "content type" and "content architecture".

The value 'concatenate' excludes specification of other attributes of the basic logical object that affect the allocation or placement of basic layout objects, such as all other layout directives.

The value 'non-concatenated' specifies that the logical object should be laid out in a new basic layout object.

6.7.1.5 New Layout Object (ALL LOGICAL COMPONENTS)

Defaultable : Default : No.

This attribute indicates that the logical object is to be placed starting within a new layout object of the specified class or category.

The value of this attribute can either be the identifier of a layout definition for a page set or page or the name of a layout category.

6.7.1.6 Layout Reference (ALL LOGICAL COMPONENTS)

Non-mandatory.

For objects, this attribute specifies the definition of a layout object of type document, page set or page into an instance of which the content of this logical object and all of its subordinates is to be laid out. It indicates that this content is to be laid out within a single instance of a document, a page set or page derived from the referenced object definition, and that no other part of the document may be laid out within the same page, page set or document.

For definitions, the above specification applies to any logical object of the class defined.

The value of the attribute is the identifier of a layout definition.

6.7.1.7 Required Category (ALL LOGICAL COMPONENTS)

Non-Mandatory.

This attribute specifies the layout category of the logical object in order to constrain its placement during the layout process.

The value of this attribute is a string constant, this being the name of the layout category.

6.7.2 Property Attributes

6.7.2.1 Protected (ALL LOGICAL COMPONENTS)

Defaultable: Default: 'unprotected'.

This attribute indicates whether or not the logical object is intended to be protected from being modified by the recipient.

6.8 Attributes of Content Portions

The attributes applicable to content portions are defined in this section.

6.8.1 Content Portion Identifier (CONTENT PORTION)

Mandatory (except in certain conformance combinations).

This attribute identifies a content portion uniquely within the context of the document and is used to refer to that content portion.

This attribute is used in the context of references to content portions (see 6.5.6).

Either one or two instances of this attribute are associated with a content portion. Each instance consists of a sequence of numbers, as defined in 6.5.2, for an identifier of a basic component, with an additional number to identify the content portion uniquely among any other content portions that are referred to by the relevant basic component.

If a content portion has two instances of this attribute, one is derived from the identifier of a logical component and one is derived from the identifier of a layout component. References to the content portion from the logical structure use the identifier derived from a logical component, while references from the layout structure use the identifier derived from a layout component.

6.8.2 Alternative Graphic Representation (CONTENT PORTION)

Non-Mandatory.

This attribute specifies a sequence of alpha-numeric characters that may be imaged in lieu of the actual content when a receiver of the document is not capable of decoding and/or imaging the content portion.

The character set to be used in this attribute is that specified in the document profile attribute "user-readable comments graphic character set".

The default character set is the minimum subrepertoire of ISO 6937, plus carriage return and line feed.

6.8.3 Type of Coding (CONTENT PORTION)

Mandatory/Optional status and default value (if any) are conformance combination and content architecture dependent.

The values this attribute can take are defined in each content architecture, to specify the coding used to represent the content, and any additional coding attributes applicable to the content portion concerned.

6.8.4 Coding Attributes (CONTENT PORTION)

Mandatory/Optional status and default value (if any) are conformance combination and content architecture dependent.

The values these attributes can take are defined in each content architecture. They are associated with the type of coding of the content portion and provide additional parametric information used in encoding/decoding the content portion.

6.9 Presentation Attributes

Presentation attributes are applicable to basic logical components and to basic layout components.

There are several sets of presentation attributes. Which set applies to a given basic component depends on the content architecture used in the component concerned. This content architecture is specified by a pair of presentation attributes that apply to all content architectures and that are therefore called "common presentation attributes". They are defined below.

6.9.1 Content Type (ALL BASIC COMPONENTS)

Defaultable (except at certain conformance combinations):
Default : Character Content Type.

This attribute specifies the type of content architecture that is associated with a basic component, for example, whether it is character or photographic content architecture.

6.9.2 Content Architecture (ALL BASIC COMPONENTS)

Defaultable: Default: depends on conformance combination.

This attribute indicates the specific content architecture that is applicable.

The default value is the maximum level content architecture of the content type concerned that is permitted at the document architecture level being used (see section 9).

6.10 Document Profile Attributes

The document profile attributes are classified in the following categories.

i) References to document structures (see 6.10.1)

These consist of references to the document structures and sets of object definitions which exist in the document.

ii) Presentation capabilities (see 6.10.2)

These consist of attributes which specify the conformance combination and content types, any non-basic attribute values and other non-basic features used in the document.

iii) Reference attributes (see 6.10.3)

These consist of attributes providing information for handling the document as a whole, in particular for storage and retrieval of the document.

6.10.1 References to Document Structures

6.10.1.1 Reference to Layout Definitions

Non-Mandatory.

This document profile attribute is present if and only if the document contains any layout definitions. The format of this attribute is that of a definition identifier (see 6.5.2) consisting of one number; its value is zero.

6.10.1.2 Reference to Layout Structure

Non-Mandatory.

This document profile attribute is present if and only if the document contains any layout structure. The format of this attribute is that of an object identifier (see 6.5.2) consisting of one number; its value is one.

6.10.1.3 Reference to Logical Definitions

Non-Mandatory.

This document profile attribute is present if and only if the document contains any logical definitions. The format of this attribute is that of a definition identifier (see 6.5.2) consisting of one number; its value is two.

6.10.1.4 Reference to Logical Structure

Non-Mandatory.

This document profile attribute is present if and only if the document contains any logical structure. The format of this attribute is that of an object identifier (see 6.5.2) consisting of one number; its value is three.

6.10.2 Presentation Capabilities

6.10.2.1 Conformance Combination

Mandatory.

The value of this attribute consists of a code specifying one of the conformance combinations PIF0, PIF1, TIF0, MIF1, MIF2, TPF0, TPF1, MPF1 and MFF2 defined in section 9.

6.10.2.2 Content Types

Mandatory.

The value of this attribute consists of a code specifying one of the following content types and combinations of content types:

- character content;
- photographic content;
- mixed, i.e. character and photographic content.

6.10.2.3 Non-basic Presentation Capabilities

6.10.2.3.1 Graphic Character Sets

Non-Mandatory.

This attribute specifies the graphic character sets, other than those of ISO 6937/2, used in the document. The value of this attribute consists of the escape sequences used to designate the sets.

6.10.2.3.2 Page Dimensions

Non-Mandatory.

This attribute specifies the page sizes, other than A4, used in the document. Each page size is represented by

a pair of numbers specifying the width and the height of the page in Basic Measurement Units.

6.10.2.3.3 Coding Attributes

Non-Mandatory.

This attribute specifies the non-basic coding attribute values used in the document.

6.10.2.3.4 Presentation Attributes

Non-Mandatory.

This attribute specifies the non-basic presentation attribute values used in the document.

A list of basic values and permitted non-basic values for each conformance combination is given in section 9.

6.10.2.3.5 Code Extension Announcers

Non-Mandatory.

This attribute specifies any locking and non-locking shift functions used in the document. The value of this attribute consists of the escape sequences defined in ECMA-35 for the purpose of announcing the use of the shift functions concerned.

6.10.2.3.6 Document Profile Graphic Character Set

Non-Mandatory.

This attribute specifies the graphic character set(s), other than the minimum sub-repertoire of ISO 6937/2, used in the document profile reference attributes. The value of this attribute consists of the escape sequence(s) used to designate the set(s) in accordance with ECMA-35 and the ISO register.

6.10.2.3.7 User-Readable Comments Graphic Character Set

Non-Mandatory.

This attribute specifies the graphic character set(s), other than the minimum sub-repertoire of ISO 6937/2, used in the user readable comments. The value of this attribute consists of the escape sequence(s) used to designate the set(s) in accordance with ECMA-35 and the ISO register.

6.10.2.4 Non-basic Structure Capabilities

6.10.2.4.1 Number of Objects per Page

Non-Mandatory.

This attribute specifies an upper limit for the number of layout objects per page used in the document. This attribute is required only if this upper limit exceeds 31.

6.10.3 Reference Attributes

This category of document profile attributes is divided into:

- date and time attributes;
- originator attributes;
- external references;
- filing and retrieval attributes;
- physical properties attributes;
- content attributes;
- document security attributes.

6.10.3.2 Date and Time Attributes

6.10.3.2.1 Document Date

Non-Mandatory.

This attribute specifies the date associated with the document by its author(s) or originating organization(s). Its value consists of the date expressed in accordance with ISO 2014.

6.10.3.2.2 Last Change Date and Time

Non-Mandatory.

This attribute specifies the date and time when this document was last changed electronically. Its value consists of the date and time expressed in accordance with ISO 2014 and ISO 3307.

6.10.3.2.3 Local Filing Date and Time

Non-Mandatory.

This attribute specifies the date and time when this document was filed. Its value consists of the date and time expressed in accordance with ISO 2014 and ISO 3307.

6.10.3.2.4 Expiry Date

Non-Mandatory.

This attribute specifies the date after which the originator considers the document to be invalid. Its

value consists of the date expressed in accordance with ISO 2014.

6.10.3.3 Originator Attributes

6.10.3.3.1 Organizations

Non-Mandatory.

This attribute specifies the originating organization(s) associated with the document. Its value consists of a set of character strings.

6.10.3.3.2 Preparer

Non-Mandatory.

This attribute specifies the name of the person responsible for the preparation/encoding of the document. Its value consists of a character string.

6.10.3.3.3 Owners

Non-Mandatory.

This attribute specifies the current administrator(s) of the document. Its value consists of a set of character strings.

6.10.3.3.4 Authors

Non-Mandatory.

This attribute specifies the author(s) of the document. Its value consists of a set of character strings.

6.10.3.3.5 Copyright

Non-Mandatory.

This attribute specifies the name(s) of the legal party/parties in whom the copyright of the document is vested. Its value consists of a set of character strings.

6.10.3.3.6 Status

Non-Mandatory.

This attribute specifies the status of the document, e.g. working draft, approved, issued, superseded, withdrawn. Its value consists of a character string.

6.10.3.3.7 User-specific Codes

Non-Mandatory.

This attribute specifies additional user-specified

code(s) of the document, e.g. contract number, project number, budget code. Its value consists of a set of character strings.

6.10.3.3.8 Circulation List

Non-Mandatory.

This attribute specifies the originator's intended recipients. Its value consists of a list of names or references in a set of character strings.

6.10.3.4 External References

None in this version of the Standard.

6.10.3.5 Filing and Retrieval Attributes

6.10.3.5.1 Title

Non-Mandatory.

This attribute specifies a word or phrase assigned by the author(s) or originating organization(s) to identify the document. Its value consists of a character string.

6.10.3.5.2 Version Number

Non-Mandatory.

This attribute specifies a code associated with the document to distinguish it from other documents bearing the same reference name. Its value consists of a character string.

6.10.3.5.3 Keywords

Non-Mandatory.

This attribute specifies logical associations to the content of the document. Its value consists of a set of character strings.

6.10.3.5.4 Reference

Non-Mandatory.

This attribute specifies a label assigned by the owner(s) for the identification of the document within a document library by man or machine for further information. Its value consists of a character string.

6.10.3.6 Physical Properties Attributes

None in this version of the Standard.

6.10.3.7 Content Attributes

6.10.3.7.1 ODA Version Date

Non-Mandatory.

This attribute specifies the date of issue of the version of this Standard to which the document conforms. Its value consists of a character string.

6.10.3.7.2 Languages

Non-Mandatory.

This attribute specifies the primary language(s) in which the main body of the document is written. Its value consists of a set of character strings.

6.10.3.7.3 Document Class

Non-Mandatory.

This attribute specifies the type of document, e.g. memorandum, letter, report. Its value consists of a character string.

6.10.3.8 Document Security Attributes

None in this version of the Standard.

7. CONTENT ARCHITECTURES

7.1 General Principles

A content architecture consists of a set of rules describing the internal structure of basic objects.

The internal structure of a basic object is specified by:

- attributes associated with the basic object;
- control functions embedded in the content portions of the basic object.

The attributes associated with the basic object are called presentation attributes. The set of presentation attributes that can be associated with a basic object and the set of control functions that can be embedded in its content portions depend on the content architecture.

The content architecture that applies to a basic object is specified by the common presentation attributes "content type" and "content architecture", which are defined in 6.9.

The definition of a content architecture consists of:

i) a general description of the possible internal structure or structures of a basic object to which the content architecture applies;

ii) an explanation of the concepts associated with the internal structure or structures that are specific to the content architecture concerned;

iii) a specification of how multiple content portions of a basic object are interpreted (e.g. combination by concatenation or fragmentation by pagination);

iv) a definition of the set of graphic elements that may be used in content portions of basic objects to which the content architecture applies;

v) a definition of the set of control functions that may be embedded in content portions of basic objects to which the content architecture applies;

vi) a definition of the set of presentation attributes that may be associated with basic objects to which the content architecture applies;

vii) definitions of the effects of embedded control functions and associated presentation attributes on the presentation, formatting, etc. of the content, including any interactions between control functions and presentation attributes;

viii) definitions of the coded representations of graphic elements, control functions and presentation attributes;

ix) a definition of the set of coding attributes that may be associated with content portions of basic objects to which the content architecture applies;

x) identification of any non-basic attributes, attribute values, graphic elements, control functions and control function parameter values, the use of which is to be announced in the document profile of the document concerned.

Two types of content architecture are defined in the current version of this Standard i.e. character content architectures (see 7.2) and photographic content architectures (see 7.3).

Additional information concerning the aspects of the content architecture definitions mentioned in items iv), v), vi) and x), above, is provided in section 9.

7.2 Character Content Architectures

7.2.1 Introduction

7.2.1.1 Content

The content of a basic object that is structured according to a character content architecture is a character string. This character string is formed by concatenating the content character strings of all content portions of the basic object.

A content character string consists of a combination of space characters (SP), graphic characters and control functions.

7.2.1.2 Graphic Characters

Any set or sets of graphic characters may be used in the content of basic objects, subject to the restrictions associated with the particular content architecture in use and with the conformance requirements in section 9, and subject to proper designation and invocation in accordance with ISO 2022.

Graphic character sets may include fixed-width space characters such as DIGIT SPACE, EM SPACE and EN SPACE.

7.2.1.3 Control Functions

Control functions consist of:

i) logical control functions which apply only to the formatting process; these control functions specify logical properties of the text, e.g. potential hyphenation; they are ignored by the imaging process;

ii) shared control functions which apply both to the

formatting process and to the imaging process; these control functions specify mainly:

- the rendition of the text, e.g. font, style, subscript, superscript,

- code extension control functions to represent additional graphic characters;

iii) layout control functions which apply only to the imaging process; these control functions specify the positioning of text within a basic layout object; they are normally generated by a formatting process and intended for the imaging process.

The definitions of control functions to be used in this Standard are to be found in Appendix A. Throughout section 7 they are referred to by their acronyms.

The scope of all control functions is local to a basic object.

7.2.1.4 Space Character

The character SP is considered both as a control character and as a graphic character.

As a graphic character, it has a graphical representation consisting of the absence of a graphic symbol.

As a control character, it acts as an inter-word space and indicates a potential line break point.

7.2.1.5 Presentation Attributes

The presentation attributes that may be associated with basic objects are classified in a similar way:

i) logical presentation attributes which apply only to basic logical objects; these attributes are intended for the formatting process but not for the imaging process;

ii) shared presentation attributes which apply both to basic logical objects and to basic layout objects;

iii) layout presentation attributes which apply only to basic layout objects; these attributes are normally generated by a formatting process, and intended for the imaging process.

The values specified by the presentation attributes should be regarded as initial values for the basic object concerned. They may be overridden by control functions embedded in the content.

7.2.1.6 Classes of Character Content Architecture

Three classes of character content architecture are distinguished:

i) character content architecture for image form which contains layout and shared control functions, but no logical control functions;

ii) character content architecture for processible form which contains logical and shared control functions, but no layout control functions;

iii) character content architecture for formatted processible form which contains logical, shared and layout control functions.

A character content architecture for image form is used in basic objects of a document that has a layout structure but no logical structure.

A character content architecture for processible form is used in basic objects of a document that has a logical structure but no layout structure.

A character content architecture for formatted processible form is used in basic objects of a document that has both a logical and a layout structure.

Different conformance levels and conformance classes specify the permitted sets of graphic characters, control functions and presentation attributes. They also specify the permitted and default values of the presentation attributes and the permitted values of the control function parameters. For a detailed description see section 9.

7.2.1.7 Internal Structure

7.2.1.7.1 Image Form

The content of a basic layout object structured according to a character content architecture for image form consists of one or more lines of characters. Each pair of successive lines is separated by a hard line terminator control function. The last (or only) line may or may not be terminated by a line terminator control function; the end of the content of the basic layout object implicitly terminates the last line.

7.2.1.7.2 Processible form

The content of a basic logical object structured according to a character content architecture for processible form consists of sequences of characters Each pair of successive character sequences is separated by a hard line terminator control function. The last (or only) character sequence may or may not be

terminated by a line terminator control function.

If the hard line terminator at the end of the basic logical object is omitted and the next basic logical object has the same content architecture as the current one and the "concatenation" attribute specifies 'concatenate' (see 6.7.1.4), then the last character sequence continues into the next basic logical object. In all other cases, the end of the basic logical object implicitly terminates the last character sequence.

7.2.1.7.3 Formatted Processible Form

The content of a basic layout object structured according to a character content architecture for formatted processible form consists of one or more lines of characters. Each pair of successive lines is separated by either a hard or soft line terminator control function. The last (or only) line may or may not be terminated by a line terminator control function; the end of the content of a basic layout object implicitly terminates the last line.

The content of a basic logical object structured according to a character content architecture for formatted processible form consists of sequences of characters. Each pair of successive character sequences is separated by a hard line terminator control function. The last (or only) character sequence may or may not be terminated by a line terminator control function.

If the hard line terminator at the end of the basic logical object is omitted and the next basic logical object has the same content architecture as the current one and the "concatenation" attribute specifies 'concatenate' (see 6.7.1.4), then the last character sequence continues into the next basic logical object. In all other cases, the end of the basic logical object implicitly terminates the last character sequence.

Soft line terminators are used as separators between lines within a character sequence.

The division into character sequences represents the internal structure of a basic logical object. Each character sequence is anonymous, in that no name or identifier is associated with it, and no relationship exists among character sequences except that of sequence.

7.2.2 Positioning and Imaging

7.2.2.1 Text Positioning

7.2.2.1.1 Basic Concepts

The initial point is the origin of the internal coordinate system specified by the attribute "initial offset" (see Figure 9).

The active position is the point at which the action specified by the next character is to be effected. If the next character is a graphic character, it is imaged with its reference point at that position. If the next character is a control function, it is performed relative to that position.

Angles are to be interpreted as counter-clockwise rotations.

7.2.2.1.2 Line Boxes

A line box is not a layout object. It is used to clarify the positioning of character boxes.

A line box is a rectangular area characterized by:

- reference point;
- height;
- nominal length;
- actual length;
- base line.

Line boxes are illustrated in Figure 9 and Figure 10.

The reference point of a line box is the origin from which character boxes are positioned within this line box. It is situated on the line through the initial point and parallel to the direction specified by the attribute "line progression".

The base line of a line box is the line through its reference point and in the direction specified by the attribute "character path".

The nominal length of the line box is the distance along the direction specified by the attribute "character path" from the reference point of the line box to the border of the basic layout object.

The actual length of a line box is the minimum length such that all character boxes of the line will be contained in it. The actual length may be longer than the nominal length, when character boxes extend beyond the reference point in the direction opposite to the character path.

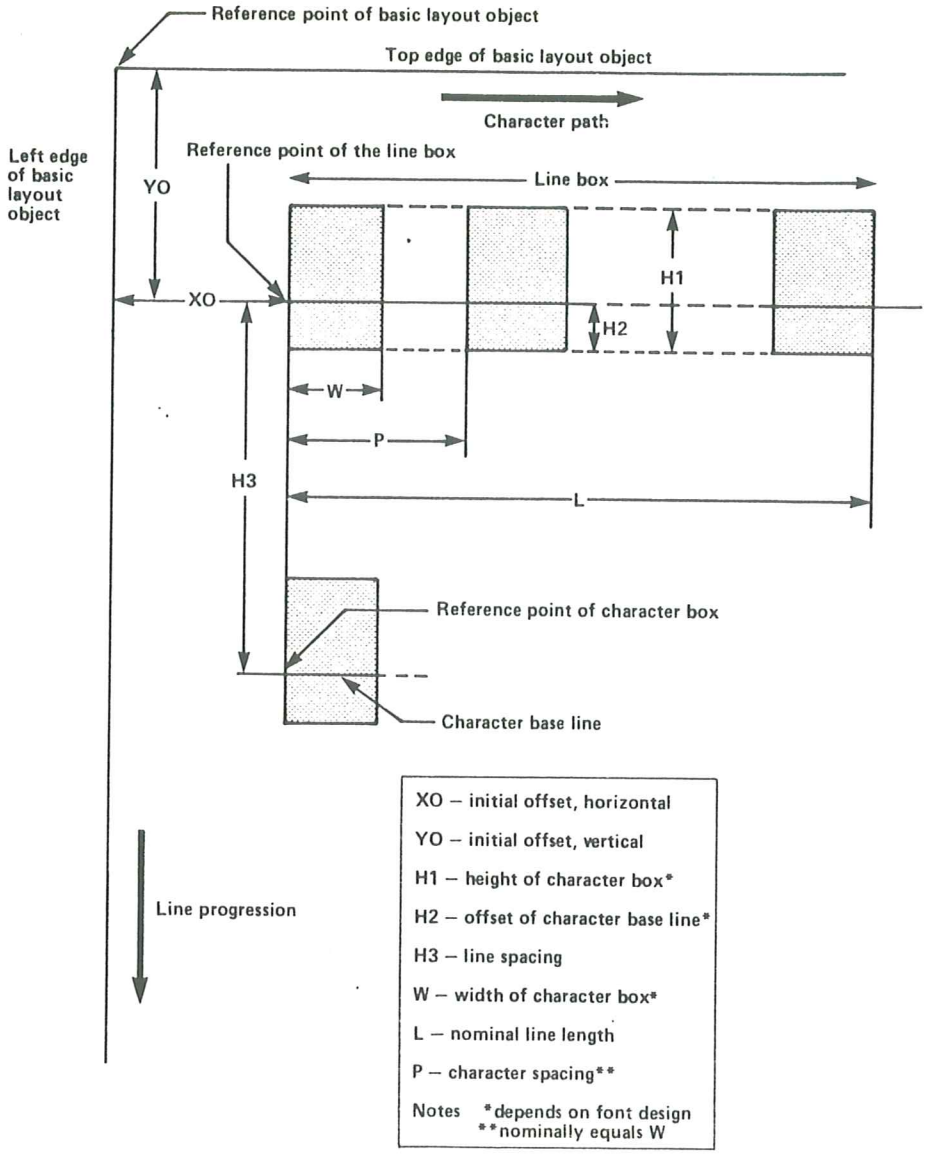
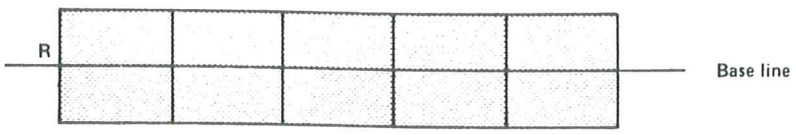
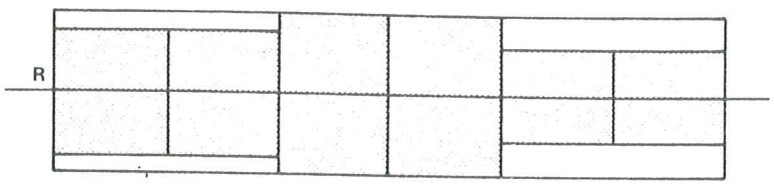


Figure 9: Example of Positioning Line Boxes and Character Boxes within a Basic Layout Object



Line box



Line box with multiple font sizes

R indicates the reference point

Figure 10: Examples of Character Boxes within Line Boxes

The height of a line box is the minimum distance perpendicular to its length such that all character boxes of the line (actual length) are completely contained within the line box.

The orientation of a line box is determined by the attributes "character path" and "line progression".

7.2.2.1.3 Distances between Line Boxes

The line spacing is the distance between the reference points of two successive line boxes.

The unit line spacing is the line spacing defined by the attribute "line spacing" or the control function SVS or SPI.

7.2.2.1.4 Character and Space Boxes

A character box is a rectangular area containing the image of a graphic character, and is characterized by:

- reference point;
- height;
- width;
- offset position;
- base line.

Character boxes are illustrated in Figure 9 and Figure 10.

The orientation of character boxes are determined by the attributes "character path" and "character orientation". For rotation of character boxes, the axis of rotation is through the nominal reference point of the character box (see Figure 11).

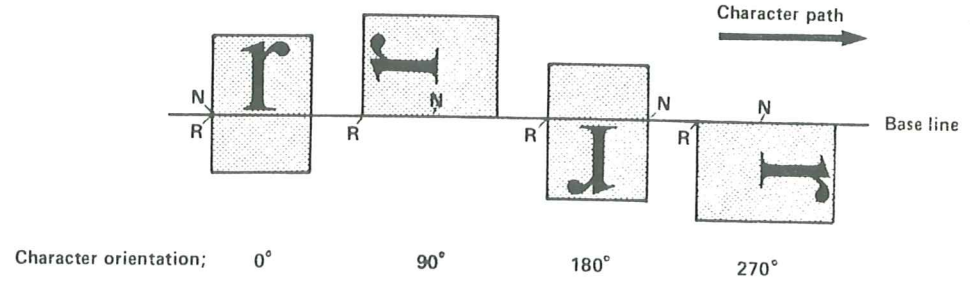


Figure 11: Orientation of Character Box or Space Box

The nominal orientation of a character box is achieved when the attribute "character orientation" has the value zero. The direction of the base line of the character box from the reference point of the character box is then the same as the direction defined by the attribute "character path".

The width of a character box is the dimension of the character box in the direction specified by the attribute "character path", when the attribute "character orientation" has the value zero. The width of a character box depends on the font design, the font size, the graphic rendition and the character that is to be imaged; in the case of a font that is designed for constant spacing, the character box width is the same for all characters.

The height of a character box is the length of the side of the character box perpendicular to its width. The height of a character box is determined by the font size and includes provision for various graphic renditions, e.g. underlining, bold.

The character base line is a positioning reference line across the width of a character box, for aligning character boxes along a line and for the placement of a symbol image within the character box.

The nominal reference point of a character box is the intersection of the character base line and the trailing edge of the character box in the direction of the character path when the character is in its nominal orientation (see Figure 11).

The actual reference point of a character box is the intersection of a line containing the nominal reference point and parallel to the character path, and the trailing edge of the character box in the direction of the character path (see Figure 11).

The reference points of the character boxes are positioned on the base line of the line box if none of the control functions PLD or PLU is active.

The character base line offset is the vertical distance between the bottom of the character box and the reference point (nominal orientation and "character path" equal to zero).

A space box is a rectangular area, similar to a character box, within which no character is to be imaged. The width of a space box depends on the font design, on the font size, on the type of space that is to be imaged and on the line justification process. Renditions, e.g. underlining, apply to space boxes.

7.2.2.1.5 Distances between Character Boxes

These spacings are illustrated in Figure 12.

The character spacing is the distance between the actual reference points of two successive character boxes.

The unit character spacing is the character spacing determined by the attribute "character spacing", or by the control function SHS or SPI.

The intercharacter spacing is the width of the spacing between successive character boxes.

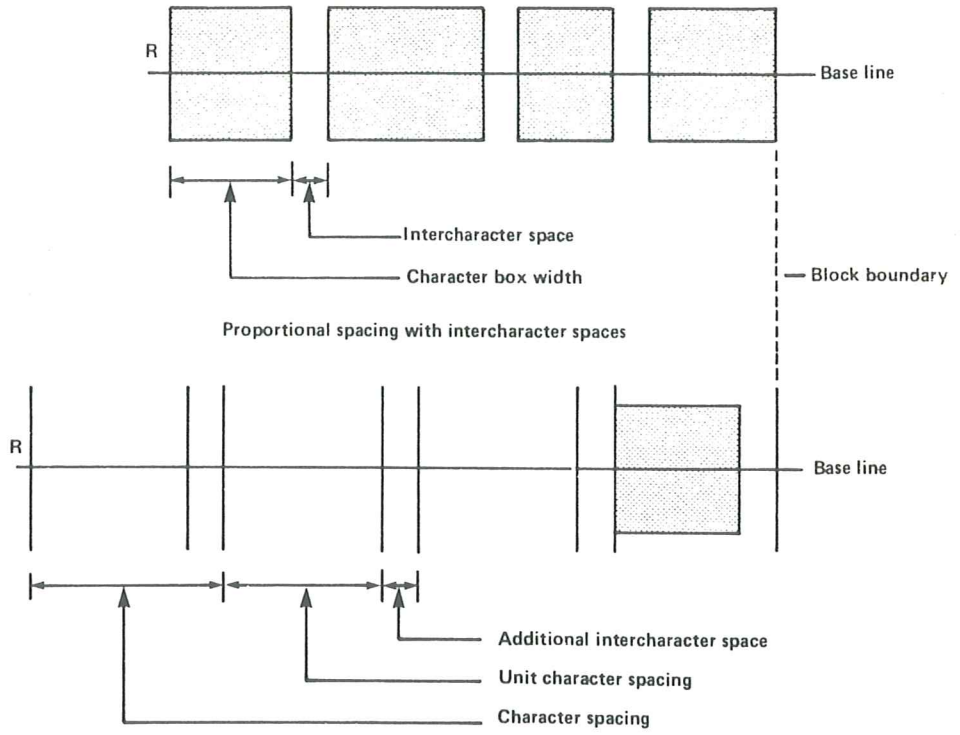


Figure 12: Examples of Character Spacing and Intercharacter Spacing

7.2.2.1.6 Coordinate Systems

The positioning of text within a basic layout object is defined in relation to two orthogonal coordinate systems: an external coordinate system and an internal coordinate system.

The origin of the external coordinate system is at the top left-hand corner of the layout object. The X-axis and the Y-axis are parallel to, respectively, the X-axis and the Y-axis of the page.

X coordinates are measured positively from the Y-axis to the right and Y coordinates are measured positively from the X-axis downward.

The origin of the internal coordinate system is specified by the presentation attribute "initial offset". The direction of the axes of the internal coordinate system are specified by the presentation attributes "character path" and "line progression".

The initial offset is specified in relation to the external coordinate system of the basic layout object.

The character path is specified relative to the direction of the X-axis of the external coordinate system.

7.2.2.1.7 Positioning of Text within a Basic Layout Object

For presentation of text in a basic layout object, the area of that object is independent of any adjoining areas. No part of the text image is permitted to extend beyond the boundaries of the basic layout object. Thus, all line boxes are contained within the basic layout object.

The definitions of control functions in content architectures are typically based on the presentation of text on a page. When such page-oriented content architectures are used within a basic layout object, all control functions operate within the area of the basic layout object as if this were the area of a page. For example, CR would move the active position to the initial character position of a line box defined within the basic layout object. If an adequate initial offset is specified in the basic layout object, it is possible to backspace along an extended projection of a line box to character positions ahead of the initial character position of that line.

7.2.2.1.8 Character Box and Character Base Line

Each alphanumeric or pictorial character is imaged within a rectangular area called a character box. The character base line is a positioning reference line across the width of a character box, for aligning character boxes within a line box and for the placement of a symbol image within the character box. The character base line may be offset from the bottom of the character box. For example, in Latin alphabets, capital letters and small letters without descenders are positioned on the character base line. A base line offset is necessary if the font design uses descenders or if underlining is to be used. For non-Latin alphabets such as Kanji, the base line offset may be zero.

A pictorial character may extend over the entire character box. In this case, the purpose of any offset of the character base line from the base of the character box may be to allow alignment of pictorial characters with alphanumeric characters.

The height and width of a character box and the offset of the character base line depend on the font design, the font size, the rendition and the character being imaged. Adjacent character boxes are nominally positioned contiguously in a line box, so that the

nominal spacing between characters is to be included within the character box.

The term 'font design' should be interpreted in a general sense so that it applies to pictorial characters as well as alphanumeric characters.

For Latin alphabets, the height of the character box and the position of the character base line are the same for all characters in a given font. The height of the character box is equal to the nominal unit line spacing for that font. Depending on the font design, the width of the character boxes may either be the same for all characters or variable from one character to another to allow for proportional spacing.

The actual reference point of a character box is used for positioning purposes.

7.2.2.1.9 Character Box Orientation

The default orientation of character boxes is based on Latin languages. Other orientations may be specified by indicating an angle of rotation. These different orientations are illustrated in Figure 11.

For rotation of character boxes, the axis of rotation is through the nominal reference point of the character box. The character base line maintains its relative position within the character box. The default rotation is 0° with respect to the direction of character path, with 90°, 180° and 270° as selectable alternatives.

A basic layout object may have only one character orientation specified.

7.2.2.1.10 Positioning of Character Boxes within a Line Box

Character path is the direction of progression of successive character boxes within a line box. The default direction is 0°, with 90°, 180° and 270° as selectable alternatives. For all character orientations, the character box reference points are aligned on the base line of the line box.

When the character base line is parallel to the character path, there are two methods for positioning the character boxes in the line box.

i) Constant Spacing

The distance from the reference point of a character box to the reference point of the next character box is constant.

This distance is nominally equal to the width of the character box for fonts of fixed character width.

ii) Proportional Spacing

The distance from the reference point of a character box to the reference point of the next character box varies within the line box. This provides for proportional spacing. This distance is nominally equal to the width of the character box for fonts with character dependent widths.

When the character box orientation is such that the character base lines are perpendicular to the character path, the distance along the base line of the line box between the reference points of successive character boxes is nominally equal to the height of the character boxes.

7.2.2.1.11 Alignment

The content of a basic object can be specified to be left aligned, right aligned, centred or justified, as described below.

i) Left Aligned

If the content is left aligned, then the character string on each line is positioned in such a way that the reference point of the first character box containing a graphic character excluding a space character is at the reference point of the line box.

ii) Right Aligned

If the content is right aligned, then the character string on each line is positioned in such a way that the leading edge of the last character box containing a graphic character excluding a space character, i.e. the edge in the direction of the character path, is at the edge of the line box.

iii) Centred

If the content is centred, then the character string on each line is positioned in such a way that the reference point of the first character box containing a graphic character excluding a space character and the leading edge of the last character box containing a graphic character excluding a space character are at approximately equal distances from the mid-point of the line box as measured from the reference point to the leading edge of the line box.

iv) Justified

If the content is justified, then the inter-word and/or inter-character spacings are varied such that the character string on each line is both left aligned and right aligned.

7.2.2.1.12 Tabulation

The positions of characters on each line in a basic layout object can be controlled by means of a set of tabulation stops. Each tabulation stop specifies a point on each line relative to the reference point of the line box.

A string of one or more characters can be specified to be positioned at the next tabulation stop by means of a control function embedded in the content. The character string may be left aligned, right aligned, centred or aligned around one or more specified characters within the string.

i) Left Aligned

If the string is left aligned, then the reference point of the first character box is positioned at the tabulation stop.

ii) Right Aligned

If the string is right aligned, then the leading edge of the last character box, i.e. the edge in the direction of the character path, is positioned at the tabulation stop.

iii) Centred

If the string is centred, then the string is positioned in such a way that the reference point of the first character box and the leading edge of the last character box are at approximately equal distance from the tabulation stop.

iv) Aligned Around

If the string is aligned around, then the reference point of the first character box of the specified group of characters within the string is positioned on the tabulation stop. If the specified group of characters does not occur within the string, then the string will be right aligned.

7.2.2.1.13 Positioning of Lines within a Basic Layout Object

The position of the first line box is established relative to the origin of the internal coordinate system specified by the "initial offset" attribute.

Line progression is the direction of progression of successive line box reference points. The default is 270°, with 90° as the selectable alternative.

The nominal length of all line boxes in the basic layout object is the same. The nominal line length is specified as the distance along the character path from the reference point to the last point of the line box at which any part of the last character box in that line box may be positioned.

Line spacing is the distance between the reference points of two adjacent line boxes. Line spacing may be equal to, greater than, or less than the dimension of the character boxes, on the side that is parallel to the direction of line progression.

7.2.2.1.14 Character Sequencing

The sequence of characters in text interchange is always the character sequence of the language being used. This is necessary since the control functions in character content architectures are defined to operate sequentially according to their position in the text stream.

7.2.2.2 Text Imaging

Text imaging for character content architectures is based on fonts and different renditions on these fonts defined by presentation attributes and control functions.

Different renditions for characters may be selected by means of presentation attributes and control functions.

7.2.2.2.1 Selectable Renditions

This subsection defines renditions selectable by the "graphic rendition" attribute or the SGR control function.

i) Underlining

This provides for underlining character boxes with either of three modes of underline: single, double or no underline.

Only one of these modes may be in effect at any one point in the text stream i.e. invoking one mode cancels any previous mode.

ii) Intensity

This provides emphasis or de-emphasis by means of the contrast of intensity levels of the foreground colour of the graphic symbol in a character box. Three modes are provided, namely: faint (reduced intensity), normal and bold (increased intensity).

Only one of these modes may be in effect at any one point in the text stream i.e. invoking one mode cancels any previous mode.

There is interaction between this parameter and certain font attributes in the option for font selection; see (v) below.

iii) Italicized

This causes a change to an italicized font, of the same font size and font design.

Two modes are provided, namely normal (not italicized) and italicized.

iv) Crossed-out Characters

This allows characters to be marked for deletion. The method of crossing out is not defined but the characters should be legible. Two modes are defined, namely normal (not crossed out) and crossed out.

v) Font Selection

This capability is based on the use of an extended repertoire of parameter values for the control function SGR. SGR is used to invoke the use of any one of a set of up to ten fonts.

The designation of this selectable repertoire of up to ten fonts can be done either by the attribute "character fonts" of the basic layout object or by the value for the attribute "character fonts" specified in the attribute "default value lists" that is applicable to some or all basic layout objects in the document.

This presentation attribute provides the selection of font size and the intended font design. Note that there is interaction with other parameter values of SGR, for use of italics and for the control of intensity of the image rendition.

A default font is implementation dependent. The default font applies to the primary font and all of the nine alternative fonts.

7.2.2.2.2 Subscript and Superscript

The control functions PLD and PLU provide for characters to be imaged as subscript and superscript, respectively.

7.2.3 Character Presentation Attributes

Presentation attributes are applicable to basic logical and layout objects. They specify the initial conditions at the start of the presentation of the content of that basic object. The content architecture associated with each basic object may have the means for changing certain of these attributes by means of control functions embedded in the content. Such capabilities are indicated in the definition of each presentation attribute given below.

Logical presentation attributes
First line indentation Orphan size Widow size
Shared presentation attributes
Alignment Character fonts Character orientation Character path Character spacing Graphic character subrepertoire Graphic rendition Line layout table Line progression Line spacing
Layout presentation attributes
Initial offset

Table 2: Character Presentation Attributes

7.2.3.1 Logical Presentation Attributes

These attributes apply only to basic logical objects; they are intended for the formatting process but not for the imaging process.

7.2.3.1.1 First Line Indentation

This attribute specifies the position of the first character on the first line of the formatted text,

relative to the position of the first character on subsequent lines. This value is measured along the character path in BMUs.

If the value is positive, it specifies the amount of indentation; if it is negative, it specifies the amount of overhang. The default value is zero. See Figure 13.

This attribute cannot be altered within a basic object.

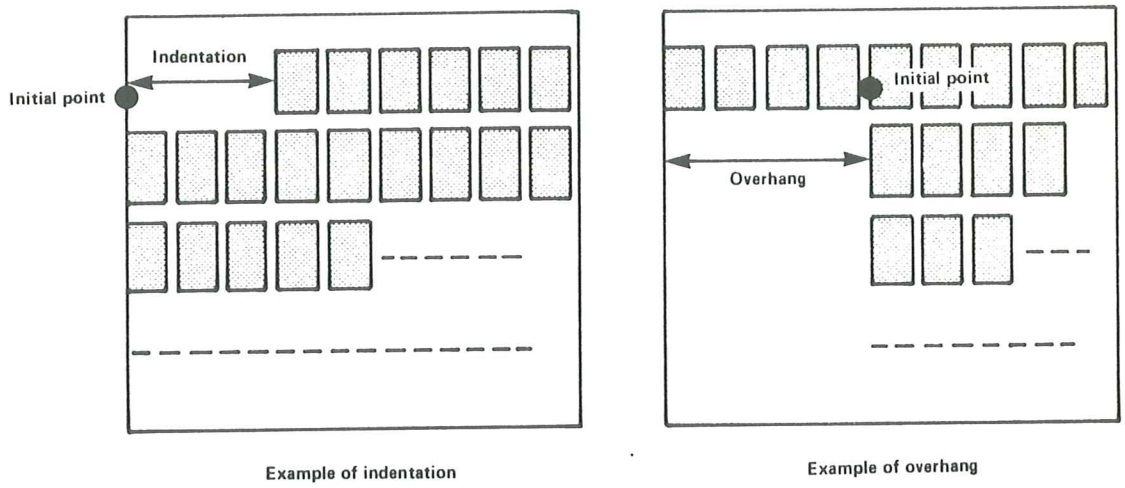


Figure 13: Examples of attribute "First Line Indentation"

7.2.3.1.2 Orphan Size

This logical attribute applies when the logical object is distributed over two or more non-contiguous basic layout objects during formatting, e.g. at a page boundary. It specifies the minimum number of text lines that must be allocated to the last basic layout object containing content of the logical object. The default value is one.

This attribute cannot be altered within a basic object.

7.2.3.1.3 Widow Size

This logical attribute applies when the logical object is distributed over two or more non-contiguous basic layout objects during formatting, e.g. at a page boundary. It specifies the minimum number of text lines that must be allocated to the first basic layout object containing content of the logical object. If during the formatting process it is discovered that the minimum number of text lines cannot be accomodated in the first basic layout object, then no text is placed within that basic layout object. The default value is one.

This attribute cannot be altered within a basic object.

7.2.3.2 Shared Presentation Attributes

These attributes apply to both basic logical objects and basic layout objects.

7.2.3.2.1 Alignment

This attribute specifies the method of positioning characters along a line.

This attribute can have one of four values: left-aligned, right-aligned, centred or justified.

The default value is left-aligned.

This attribute cannot be altered within a basic object.

7.2.3.2.2 Character Fonts

This is a group of ten attributes designating up to ten fonts which may be used within a basic object. Invocation of a particular font within a basic object is effected by the SGR control function.

The invocation of a particular font implicitly specifies the character box width and the base line offset.

Each attribute is made up of two parameters, as follows:

i) The first parameter specifies the size of the font. The value of this parameter is an integer representing the size of the font in BMU (i.e. the nominal height of the character box to be used with the font).

ii) The second parameter identifies the font design. This parameter identifies the font design according to a register. Whether the font is designed for constant spacing or proportional spacing can thereby be deduced directly from the value of the identifier. In the absence of font registration only constant spacing fonts can be supported.

The default value of each of the ten font selection attributes is implementation-dependent. No default values are defined for the individual parameters since both must be specified when a font selection attribute is used.

7.2.3.2.3 Character Orientation

This attribute specifies the orientation of the character box relative to the character path. The value of this attribute specifies the angle of rotation

from the nominal orientation around the nominal reference point of the character box.

This attribute can have one of four values: 0°, 90°, 180° or 270°. The default value is 0°.

This attribute cannot be altered within a basic object.

7.2.3.2.4 Character Path

This attribute specifies the direction of progression of successive character boxes in a line box, relative to the X-axis.

This attribute can have one of four values: 0°, 90°, 180° or 270°. The default value is 0°.

This attribute cannot be altered within a basic object.

7.2.3.2.5 Character Spacing

For constant spacing this attribute specifies the unit character spacing.

For proportional spacing this attribute is ignored.

The value of this attribute is an integer specifying the distance in BMUs. The default value is 120 BMU.

This attribute may be altered within a basic object by means of the control function SHS or SPI.

7.2.3.2.6 Graphic Character Subrepertoire

This attribute specifies the subrepertoire of the graphic character repertoire ISO 6937 used in the basic object.

The value of this attribute is the numeric identifier of the subrepertoire assigned in accordance with the registration procedure of ISO 7350.

The default value depends on the individual character content architecture being used.

This attribute may be altered within a basic object by means of the control function IGS.

7.2.3.2.7 Graphic Rendition

This attribute specifies the rendition parameters which apply to the basic object.

The value of this attribute is a sequence of integers each of which equals any of the SGR parameter values applicable to the individual character content architecture being used.

The default value is zero.

This attribute may be altered within a basic object by means of the control function SGR.

7.2.3.2.8 Line Layout Table

This attribute specifies a sequence of tabulation stops along the character path. Each stop is specified by a group of 1, 2 or 3 subordinate parameters, as follows:

i) a tabulation position parameter that specifies the distance in BMUs of the tabulation stop from the reference point of the line box;

ii) an optional alignment parameter that specifies whether the characters at that tabulation position are to be left aligned, right aligned, centred or aligned around one or more specified characters. The default alignment is left aligned;

iii) an optional alignment character string. This parameter is required only when the alignment parameter specifies alignment around a group of one or more characters.

The default is that no tabulation stops are defined.

This attribute cannot be altered within a basic object.

7.2.3.2.9 Line Progression

This attribute specifies the direction of progression of successive line boxes, relative to the character path.

This attribute can have one of two values: 270° or 90°.

The default value is 270°.

This attribute cannot be altered within a basic object.

7.2.3.2.10 Line Spacing

This attribute specifies the unit line spacing. The value of this attribute is an integer specifying the distance in BMUs.

The default value is 200 BMU.

This attribute may be altered within a basic object by means of the control function SVS or SPI.

7.2.3.3 Layout Presentation Attributes

These attributes apply only to basic layout objects; they are normally generated by a formatting process, and

intended for the imaging process.

7.2.3.3.1 Initial Offset

This attribute specifies the origin of the internal coordinate system of the basic layout object.

The value of this attribute consists of two integers that specify the X and Y coordinates of the internal origin, in relation to the external coordinate system of the basic layout object.

The default value of this attribute, assuming default character orientation and no first line indentation, depends on character path, line progression and line spacing, as defined in Table 3.

Character path	Line progression	X	Y
0°	270° / 90°	0 / 0	S / H-S
90°	270° / 90°	S / W-S	H / H
180°	270° / 90°	W / W	H-S / S
270°	270° / 90°	W-S / S	0 / 0

W = X dimension of basic layout object
H = Y dimension of basic layout object
S = unit line spacing

Table 3 - Default Values of "Initial Offset"

The eight cases of first character positions given by these combinations of the attributes "character path" and "line progression" are illustrated in Figure 14.

Figure 14 assumes the default values of the attributes "character orientation" and "first line indentation".

Xo = default X coordinate of initial offset
Yo = default Y coordinate of initial offset
CP = direction of character path (relative to the X-axis)
LP = direction of line progression (relative to the character path)

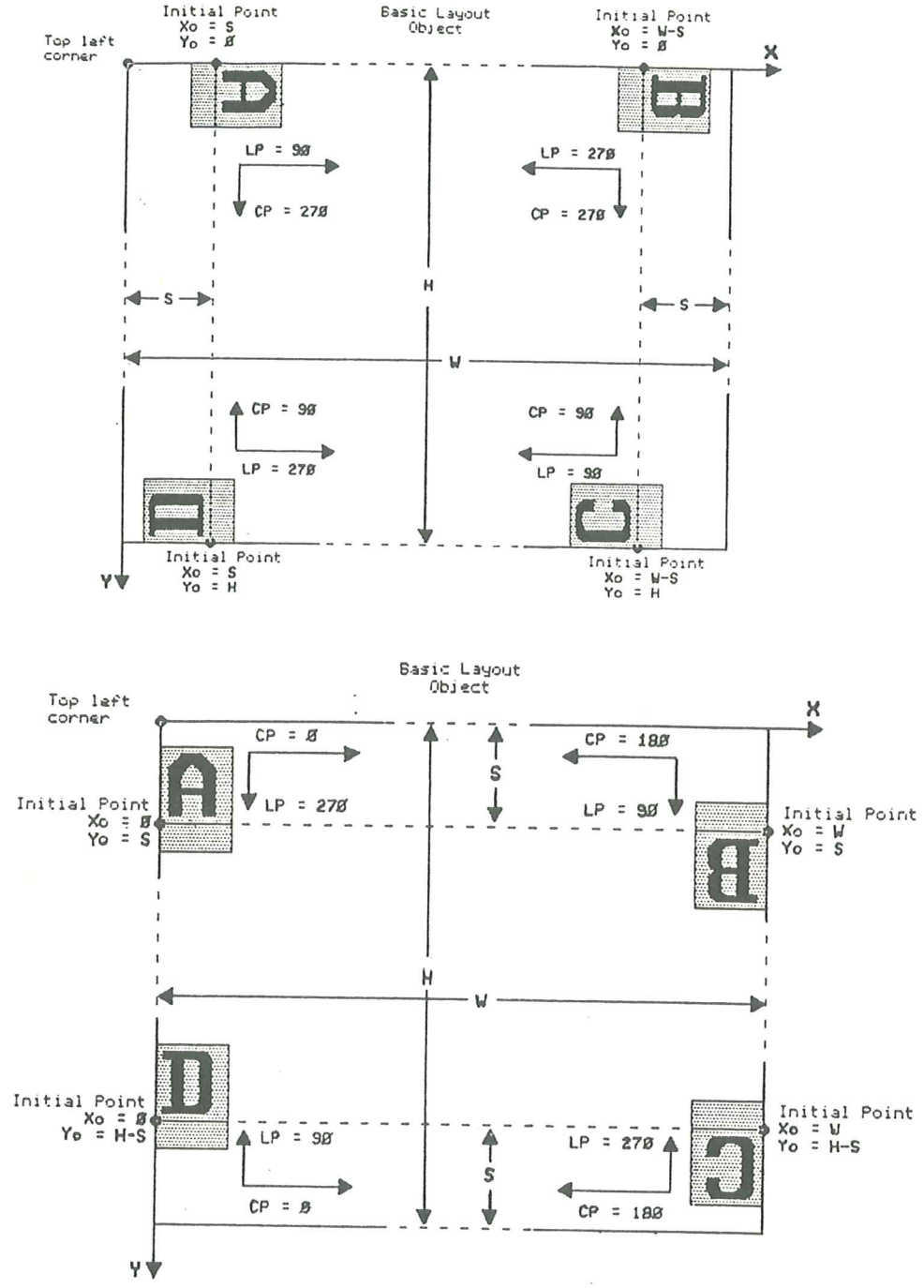


Figure 14: First Character Positions

7.2.4 Graphic Characters

When the graphic character sets of ISO 6937/2 are used, the character repertoire may be restricted by the presentation attribute "Graphic Character Subrepertoire" or by the control function IGS.

Graphic characters never indicate a potential line break point. Potential line breaks are indicated only by the character SP or by the control function BPH.

7.2.5 Control Functions

Logical control functions	
BPH	BREAK PERMITTED HERE
NBH	NO BREAK HERE
Shared control functions	
CR	CARRIAGE RETURN
HT	HORIZONTAL TABULATION
IGS	IDENTIFY GRAPHIC SUBREPERTOIRE
LF	LINE FEED
PLD	PARTIAL LINE DOWN
PLU	PARTIAL LINE UP
SGR	SELECT GRAPHIC RENDITION
SHS	SELECT CHARACTER SPACING
SPI	SPACING INCREMENT
SUB	SUBSTITUTE
SVS	SELECT LINE SPACING
Layout control functions	
BS	BACKSPACE
EM	END OF MEDIUM
HPB	CHARACTER POSITION BACKWARD
HPR	CHARACTER POSITION RELATIVE
Delimiters	
DCS	DEVICE CONTROL STRING
ST	STRING TERMINATOR

Table 4 - Control Functions

The definitions of the control functions are specified in Appendix A.

7.2.5.1 Logical Control Functions

Logical control functions apply only to the formatting process and are ignored by the imaging process.

Logical control functions for the character content architectures defined in this Standard are BPH (BREAK PERMITTED HERE) and NBH (NO BREAK HERE).

BPH specifies a potential line break point.

NBH specifies a point where a line break should not occur. This control function is used at a point immediately preceding a potential line break point, e.g. immediately preceding an interword SP.

7.2.5.2 Shared Control Functions

These control functions apply to both the formatting process and the imaging process.

A hard line terminator is represented by a LF (LINE FEED) that immediately follows either a CR (CARRIAGE RETURN) or another LF.

SUB (SUBSTITUTE) is used in the place of a character that has been found invalid or in error.

Any code extension control function defined in ISO 2022 can be used in a basic object. However, shift functions used in a document shall be announced in the document profile.

IGS (IDENTIFY GRAPHIC SUBREPERTOIRE) specifies the subrepertoire of the graphic character repertoire of ISO 6937 used.

The parameter value is the numeric identifier of the subrepertoire assigned in accordance with the registration procedure of ISO 7350.

HT (HORIZONTAL TABULATION) is a control function which specifies the positioning and alignment of subsequent text, until either the next occurrence of another HT or the end of the current line.

The text string is to be positioned at the next tabulation stop along the line, and aligned in accordance with the properties of that tabulation stop.

The rendition and positioning of graphic characters is indicated by the control functions SGR, PLD, PLU, SHS, SVS and SPI.

SGR (SELECT GRAPHIC RENDITION) can be used with the parameter values 0, 1, 2, 3, 4, 9, 10 to 19, 21, 22, 23, 24 and 29. The set of permitted values depends on the conformance level.

In the context of this Standard, SGR operates in the cumulative mode, i.e. the following graphic rendition aspects can be controlled independently from each other:

- intensity;
- italic;
- underlining;
- crossed-out;
- font.

In the context of this Standard, there is no limit to the number of parameter values in an SGR sequence.

PLD (PARTIAL LINE DOWN) indicates the beginning of subscript rendition or the end of superscript rendition.

PLU (PARTIAL LINE UP) indicates the beginning of superscript rendition or the end of subscript rendition.

Neither subscript nor superscript is permitted to be active at a line terminator.

SHS (SELECT CHARACTER SPACING) specifies the unit character spacing for fonts with constant spacing.

SVS (SELECT LINE SPACING) specifies the unit line spacing.

SPI (SPACING INCREMENT) specifies the unit character spacing and the unit line spacing in BMUs.

7.2.5.3 Layout Control Functions

These control functions apply to the imaging process; they specify the internal structure of a basic layout object and the positioning of text within a basic layout object; they are normally generated by a formatting process and intended for the imaging process.

A soft line terminator is represented by a LF that immediately follows either a CR or another LF, where the entire sequence of CR(s) and LF(s) is enclosed between a DCS control function and an ST control function.

Positioning control functions which may be embedded at the beginning of a line are BS, HPB, and HPR.

HPB and HPR may be embedded within the lines.

BS (BACKSPACE) defines a positioning in the direction opposite to the character path with an amount equal to the unit character spacing for constant spacing, and equal to the width of the character SP as defined by the current font design and font size for proportional spacing.

EM (END OF MEDIUM) is generated by the formatting process to indicate that the current line must not be justified.

HPB (CHARACTER POSITION BACKWARD) represents positioning in the direction opposite to the character path from where the next character box would have been positioned, with an amount specified by its parameter.

HPR (CHARACTER POSITION RELATIVE) represents positioning in the direction of the character path from where the next character box would have been positioned, with an amount specified by its parameter.

In the context of this Standard, the parameters of HPB and HPR are expressed in BMUs (i.e. the POSITIONING UNIT MODE is implicitly set to SIZE, and the size unit is the BMU, see ECMA-48).

7.2.5.4 Delimiters

DCS acts as the opening delimiter of a control string which is closed by the terminating delimiter ST.

A control string may contain occurrences of graphic characters and shared control functions, in particular CR, LF and Hyphen, introduced as a result of an interim formatting process.

These delimiters are used in a character content architecture for formatted processible text to identify a sequence of graphic characters and control functions that are significant for the imaging process only.

7.2.6 Space Character

To the formatting process, SP acts as an inter-word space and indicates a potential line break point.

To the imaging process, SP acts as a graphic character whose graphical representation consists of the absence of a graphic symbol.

Any SPs that follow a soft line terminator and precede the first graphic character of a line, or that precede a soft line terminator and follow the last graphic character of a line, are ignored by the imaging process, i.e. are considered to have a width equal to zero.

For any other SP, the width is determined by the imaging process on the basis of the font design and for proportional spacing the font size, or for constant spacing the unit character spacing.

All consecutive SP following NBH are not potential line break points.

7.2.7 Interactions Between Attributes and Control Functions

All attributes and control functions are not completely independent. This section describes constraints on their use.

LF is restricted to be used in the following cases:

- at the beginning of a basic layout object;
- immediately following a CR;
- immediately following another LF.

A normal SP is not permitted while PLU or PLD is active unless it is immediately preceded by NBH.

BPH is not permitted while PLU or PLD is active.

Rendition aspects defined by some presentation attributes can be redefined by control functions embedded in content portions. These are specified in Table 5.

Presentation Attribute	Control Function
Character spacing	SHS, SPI
Line spacing	SVS, SPI
Graphic rendition	SGR
Graphic character subrepertoire	IGS

Table 5: Interaction of Presentation Attributes and Control Functions

When the attribute "concatentation" has the value 'concatenate' for a basic logical object of character content architecture the following constraints are imposed:

- the basic logical object must not have the attribute "first line indentation";
- both this and the preceding basic logical objects must have identical values for the attributes "alignment", "character orientation", "character path", "content type", "content architecture", "line layout table" and "line progression".

The attribute "line progression" can interact with the document layout attribute "object path". For example, if the attributes "line progression" and "object path" are in opposite directions, then further blocks cannot be created within the frame once the first block has been created.

The attributes "alignment" and "line layout table" cannot be used on the same basic object.

7.2.8 The Character Content Formatting Process

This section presents an abstract model of the character content formatting process. Any functionally equivalent implementation conforms to this Standard.

Within the character content architecture, the formatting process consists of determining the content of, and creating, a basic layout object by determining the position of each character box. It also attaches the relevant attributes to the basic layout object. During this process, the dimension in the direction of the character path of a basic layout object is assumed to be fixed, having been previously determined elsewhere.

The first step of this process involves the removal of all DCS control strings and BS, EM, HPB and HPR control functions which should result only from the formatting process.

The second step groups the content portions of each basic logical object together such that each basic logical object contains only one content portion.

The next step involves the determination of the initial point. It depends on the attributes "character path", "line progression" and "first line indentation". The formatting process determines the initial point such that the first line box, as implied by the presentation attributes, would touch the boundary of the basic layout object in the direction opposite to the line progression if the first line box were positioned at the initial point.

The next step involves the determination of line breaks and the determination of the position of each character within the line box. As a general rule the basic layout object is utilised as much as possible in the direction of the character path.

Positioning by means of the "first line indentation" attribute will create a HPR or HPB control function.

A potential line break is indicated by the space character SP or by an occurrence of BPH (BREAK PERMITTED HERE). In addition, line breaks may be determined according to an implementation-dependent hyphenation algorithm.

When the output of the formatting process is in image form, then:

- a hard line terminator is inserted at each line break;
- all occurrences of SP that immediately precede or follow a line break are deleted;
- all occurrences of BPH and NBH are deleted.

When the output of the formatting process is in formatted processible form, then a soft line terminator is inserted at each line break; no characters are deleted from the content.

If the alignment process as specified by the attribute "alignment" is performed by the formatting process, the four different values 'left aligned' (default), 'right aligned', 'centred' and 'justified' are treated as follows:

- 'left aligned' will not result in any additional control functions;
- 'right aligned' and 'centred' will result in an additional HPR control function at the beginning of the line;
- 'justified' will result in the insertion of control functions such as HPR, SHS and SPI. Any shared control functions inserted must be enclosed in a pair of delimiters DCS and ST if the resulting content architecture is of the formatted processible form. If the resulting content architecture is of the image form, the delimiters DCS and ST are not permitted.

When 'right aligned' or 'justified' has been specified for a font with constant spacing, the distance from the actual reference point of the last character box of the line, to the end of the basic layout object in the direction of the character path equals the unit character spacing.

When 'right aligned' or 'justified' has been specified for a font with proportional spacing, the leading edge, in the direction of the character path, of the last character box of the line touches the border of the basic layout object.

When the alignment is performed by the formatting process, the attribute "alignment" is not transferred to the layout structure.

If the alignment process as specified by the attribute "alignment" is to be performed by the imaging process and 'justified' is specified, the formatting process inserts the control function EM in the last line of a character sequence. This prevents the last line from being justified.

If tabulation as specified by the attribute "line layout table" and the control function HT is performed by the formatting process, the four different values of the alignment parameter of the tabulation stop, 'left aligned' (default), 'right aligned', 'centred' and 'aligned around' are treated as follows:

- when the tabulation stop following the control function HT has the alignment parameter 'left aligned' the control function HPR will be added immediately following HT, such that the actual reference point of the next character box will be positioned at the tabulation stop;

- when the tabulation stop following the control function HT has the alignment parameter 'right aligned', a control function HPB or HPR will be added immediately following HT, such that the last character box before the next HT or the end of the line will be positioned at the tabulation stop. For fonts with proportional spacing the leading edge, in the direction of the character path, of the last character box will be positioned at the tabulation stop, and for fonts with constant spacing the actual reference point of the last character box will be positioned a unit character spacing before the tabulation stop;

- when the tabulation stop following the control function HT has the alignment parameter 'centred' a control function HPB or HPR will be added immediately following HT, such that the midpoint of the string of characters before the next HT or the end of the line will be positioned at the tabulation stop;

- when the tabulation stop following the control function HT has the alignment parameter 'aligned around', a control function HPB or HPR will be added immediately following HT, such that the actual reference point of

the first character in the specified string of characters contained in the text before the next HT or the end of the line will be positioned at the tabulation stop.

If there are no more tabulation stops in the direction of the character path from the position of the control function HT, the control function will be ignored.

When tabulation is performed by the formatting process, the attribute "line layout table" is not transferred to the layout structure.

Alignment and tabulation should be performed by the formatting process only if it is known that the imaging process will use a character font that is similar to that assumed by the formatting process, i.e. a font that has the same width for each character as the specified font.

If the basic layout object corresponds to multiple basic logical objects, control functions may have to be inserted at the beginning of the first content portion of the concatenated logical object. When the output of the formatting process is in formatted processible form, these control functions are enclosed in a pair of DCS and ST control functions.

A content portion of a basic logical object may be split into several content portions in order to place them in different layout objects. The attributes of these layout objects are given the values corresponding to the current status of the rendition.

The dimension of the basic layout object in the direction of line progression is determined from the position of the last line box of a sequence of concatenated logical objects. Finally the value of the attribute "initial offset" is determined.

A detailed description of the positioning of character boxes and line boxes is found in the imaging process below.

7.2.9 The Character Content Imaging Process

This section presents an abstract model of the character content imaging process. Any functionally equivalent implementation conforms to this Standard.

The imaging process consists of the construction of the image of a basic layout object from a set of line boxes, each of which is built from a set of character boxes and space boxes.

The content of the basic layout object that is to be imaged consists of the concatenation of the relevant content portions which have been prepared by a formatting process.

All logical control functions, if any, are ignored by the character content imaging process.

The position of each line box is determined relative to a base position, taking into account any LF control functions occurring at the beginning of the line.

The base position of the first line box of the basic layout object is at the origin of the internal coordinate system specified by the attribute "initial offset".

The base position of any line box which is not the first line box of the basic layout object is such that the distance between the base line of the line box and the base line of the preceding line box is equal to the current value of the unit line spacing.

The actual position of a line box is such that the distance from its base position is equal to the unit line spacing multiplied by the number of LF control functions occurring at the beginning of the line (excluding the LF which is part of the line terminator of the preceding line).

When the attribute "alignment" is present, the imaging process performs the alignment on a line-by-line basis in a way similar to that described for the formatting process.

Any line that contains the control function EM is excepted from being justified.

The character SP defines a space box. The width of this space box is determined as follows:

- for any SP, that follows a soft line terminator and precedes the first graphic character of a line, or precedes a soft line terminator and follows the last character of a line, the width is equal to zero;

- in a font with constant spacing, the width of SP is defined by the last preceding control function SHS or SPI in that basic layout object, if any. Otherwise it is defined by the attribute "character spacing";

- in a font with proportional spacing, the width of SP is implicitly defined by the font design and font size.

When the attribute "line layout table" is not present, the control functions HT are ignored.

When control functions HT and the attribute "line layout table" are present, the imaging process performs the tabulation in a way similar to that described for the formatting process.

A graphic character defines a character box. The width of a character box depends on the current font design and font size, on the current graphic rendition, and on the current graphic character being imaged.

The current graphic rendition is determined by the last preceding control function SGR in that basic layout object, if any. Otherwise it is defined by the attribute "graphic rendition".

The image of a graphic character in its character box depends on the current font design and font size. This image is modified by the application of the current graphic rendition attributes and by superscript or subscript situations. The image and the effect of modification is part of the font design. Graphic rendition also applies to SP.

The reference points of the space and character boxes are aligned on the base line of the line box.

Positioning of a space box or of a character box takes into account any control functions immediately preceding it.

For proportional spacing the positioning is determined as follows:

- two successive character boxes are positioned adjacent to each other;
- two successive space boxes are positioned adjacent to each other;
- space boxes and character boxes are positioned adjacent to each other;
- if the control function HPR is specified between two space or character boxes, the distance (in the direction of the character path) between the leading edge of the preceding box and the trailing edge of the succeeding box is that specified by HPR.

For constant spacing the positioning is determined as follows:

- two successive character boxes are positioned such that the distance between their actual reference points equals the character spacing;
- two successive space boxes are positioned such that the distance between their actual reference points equals the character spacing, i.e. adjacent to each other;
- space and character boxes are positioned such that the distance between their actual reference points equals the character spacing;
- if the control function HPR is specified between two space or character boxes, the distance between their actual reference points is the sum of the character spacing and the value specified by HPR.

When a change of font is made, the constant or proportional spacing characteristic of the succeeding character determines which of the above rules apply.

When the control function HPB or HPR is at the beginning of a line, it specifies the position relative to the reference point of the line box where the first character box should be positioned.

A line terminator causes the acquisition of content for the current line box to be terminated.

7.2.10 Coding

Coded representations of graphic characters, control functions and SP are in accordance with ECMA-6, ECMA-35, ECMA-48 and ISO 6937/3.

A summary of these coded representations is given in Appendix B.

These standards define the coded representations of control functions and SP.

8-bit character coding is used in content portions.

Coded representations of graphic characters are defined in the applicable standards, e.g. ISO 6937/2, or in the International Register of Character Sets.

Coded representations of presentation attributes are defined in 8.3.6.

There are no coding attributes for content portions defined in the character content architectures.

7.3 Photographic Content Architectures

7.3.1 Introduction

The content of a basic object of type photographic is a sequence of individual photographic elements (pels). These elements in conjunction with rendition attributes are used to render the content as an image.

A pictorial image is made up of a sequence of lines formed into a rectangular array. A line in turn consists of a string of pels, each pel has a specific shape, size, colour, intensity and position. The position of the pels within the image is determined using the rendition attribute, which specifies the pel direction i.e. the attribute, "pel path".

Within a given block all line lengths are the same. Each line follows on from the preceding one. The new line begins directly underneath the initial pel of the previous line. Thus the line progresion is perpendicular to the pel path. The image itself is made up from the coded content and the

image resolution is determined by the density of the image rectangular array.

In the photographic content architecture basic objects can only have a single content portion.

In a basic layout object of type photographic content the origin is identical to the reference point.

In the context of photographic content architecture the term 'horizontal' denotes the direction parallel to the pel path and 'vertical' denotes the direction perpendicular to the pel path, i.e. parallel to line progression.

Angles are to be interpreted as counter clockwise rotations.

7.3.2 Coding Schemes

This Standard permits photographic content to be encoded according to several coding schemes.

The coding scheme to be used is that of CCITT Recommendation T.6. This coding scheme has three components:

- information code words;
- control functions;
- pad bits.

There are two photographic content architectures. the differences between them consist of the classification or the default values of the following attributes:

- resolution;
- number of pels per line;
- number of pels discarded.

The two photographic architectures are designated PH0 and PH1.

7.3.2.1 Information Code Word

The information code word contains the actual image data and together with the control functions and pad bits create the desired image.

7.3.2.2 Control Functions

The following control functions are defined.

i) End of Facsimile Block
The end of a facsimile block code is added to the end of every coded facsimile data block.

ii) Extension
Extension code is used to indicate the change from the current mode to another mode, e.g. another coding scheme;

Uncompressed mode is an optional coding scheme associated with the CCITT Recommendation T.6 coding scheme and is used to transmit the image information without data compression technique.

7.3.2.3 Pad Bits

Pad bits may be used after the end of facsimile blockcode if it is necessary to align on octet boundaries or to a fixed data block size.

7.3.3 Attributes

Two types of attributes are used, rendition attributes which are associated with the basic layout object and coding attributes which are associated with the content portions.

7.3.3.1 Rendition Attributes

7.3.3.1.1 Initial Offset (BASIC OBJECTS)

Defaultable : Default 0,0.

This attribute specifies the reference position of the first photographic element (pel) within the basic layout object. The value of this attribute is a pair of integral numbers that specify the X and Y coordinates of the first pel relative to the origin of the basic layout object expressed in BMUs.

7.3.3.1.2 Pel Path (BASIC OBJECTS)

Defaultable : Default 0°.

This attribute specifies the direction of progression of successive pels along a line (relative to the X-axis of the basic layout object).

This attribute can have one of four values, 0°, 90°, 180°, or 270°.

7.3.3.1.3 Line Progression (BASIC OBJECTS)

Defaultable : Default 270°.

This attribute specifies the direction of progression of successive lines relative to the Pel Path.

This attribute can have one of two values, 90° or 270°.

7.3.3.1.4 Resolution (Pel Transmission Density) (BASIC OBJECTS)

Defaultable: Default: depends on conformance combination.

This attribute specifies the number of pels in the unit length (of 25,4 mm), which can be given uniformly or independently for the horizontal and vertical

directions.

The value of this attribute is a code in the uniform case or a pair of integers in the non-uniform case. In the latter case, the first stands for the resolution in the direction of the pel path and the second stands for the resolution in the direction of the line progression.

7.3.3.2 Coding Attributes

7.3.3.2.1 Number of Pels per Line

This is specified as an integer.

The default value depends on the conformance combination.

7.3.3.2.2 Number of Pels Discarded

This attribute specifies the number of pels, within the number of pels per line, that are to be discarded at the beginning of each interchanged image line of pels. Imaging is started from the next pel, at the initial pel position (specified by the combination of layout object boundary plus initial offset, if any).

The default value depends on the conformance combination.

7.3.3.2.3 Number of Lines

This is specified as an integer.

The default value depends on the conformance combination.

7.3.3.2.4 Compression

The basic coding technique specified uses a compression algorithm.

In an alternative mode, the image information may contain uncompressed parts.

The default value depends on the conformance combination.

7.3.4 Imaging Process

This section represents an abstract model of the photographic content imaging process. Any functionally equivalent implementation conforms to the Standard.

The imaging process makes use of a photographic element (pel) which is an individual picture element used in arrays to construct images. Each pel has a specific shape, size, colour, intensity and position. A pictorial image is formed

by a rectangular array consisting of sequences of pels in lines advancing at right angles to the direction of these lines.

The first pel to be imaged on each line is positoned at the initial position as established by the initial offset attribute (if any) from the origin of the layout object. Imaging of pels does not extend beyond the end of the line which is at the edge of the basic layout object. The data is converted from long strings of pels which are then divided into lines by the attribute "number of pels per line".

Provision is made within the attribute "number of pels discarded" to accommodate the 'overscan' practice of certain facsimile equipment. This attribute may be used either explicitly or depending on the conformance level (in the case of TIF0, see section 9), implicitly using a default value. If this attribute is used, the first pel to be imaged is the first one following the number of pels specified to be discarded on each line. Any pels that would extend beyond the end of the line are also discarded.

Photographic elements in a line are arranged sequentially in the pel path of the line.

The pels on a line are nominally positioned at constant intervals along the line length.

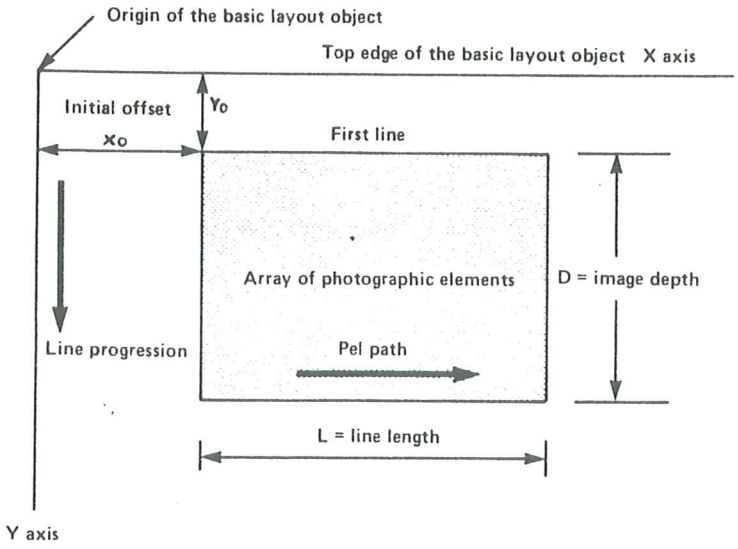
Image resolution is determined by combining horizontal resolution, given by the density of pels along the lines, and vertical resolution, given by the line density. By default these two image resolution factors are equal, but they could be specified independently.

The default path direction is horizontal from left to right, with the page in the vertical orientation, irrespective of whether the page image is to be viewed in the vertical or horizontal (i.e. 'Landscape') page orientation. Alternatively, the pel path direction may be specified with respect to the page coordinate system, as either 90°, 180° or 270°.

Each subsequent line is positioned relative to the preceding line at a distance specified by the vertical resolution attribute. The lines are nominally positioned at constant intervals along the image depth. Line progression is the path direction from the initial pel of the first line to the initial pel of subsequent lines. It is perpendicular to the pel path; the default value of this attribute is 270°, with 90°as the alternative.

In this Standard individual pels are of sufficiently small size that they are, for purposes of positioning, treated as points.

The resulting array of photographic elements is illustrated in Figure 15.



L = line length implemented by attribute number of pels per line and the image resolution

D = image depth implementedby attribute number of lines and the image resolution

X_o = X component of initial offset. Y_o = Y component of initial offset.

Figure 15: Positioning of Photographic Elements

8. INTERCHANGE FORMAT

8.1 General Description of the Data Stream

The data stream consists of a data structure representing a document structured according to the Office Document Architecture. Data structures subordinate to the data stream correspond to the components of the document. A data structure corresponding to the document profile, to an object definition or to a logical object or layout object is called a descriptor. A data structure corresponding to a content portion is called a text unit.

The descriptors and text units are ordered in the data stream so that the document profile descriptor comes first, followed by descriptors and text units representing any object definitions and common content portions, and finally descriptors and text units representing any logical objects and layout objects and associated content portions. This is illustrated in Figure 16.

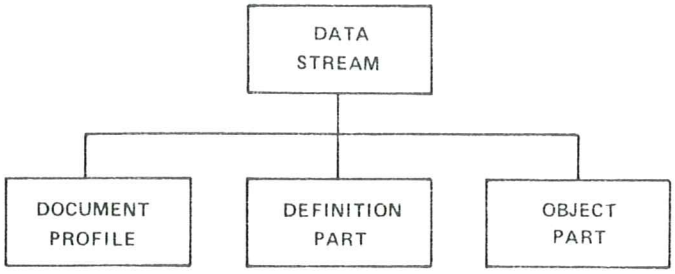


Figure 16: Structure of the Data Stream

The order of the descriptors representing the objects corresponds to the order of the nodes in each tree structure, i.e. starting at the root, then going down to the first leaf and then traversing to the next leaf. This is illustrated in Figure 17.

The order of the descriptors representing the logical objects corresponds to the sequential order of the objects in the logical structure, and the order of the descriptors representing the layout objects corresponds to the sequential order of the objects in the layout structure, as defined in 4.1.1. The order of the descriptors representing logical or layout object definitions is such, that descriptors are contiguous, when the leading elements of the corresponding definition identifiers are the same.

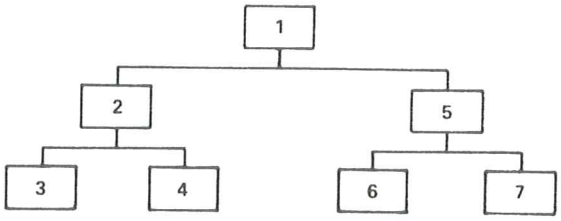


Figure 17: Example of the Sequential Order of Descriptors Representing Logical or Layout Objects

The text units are distributed over the data stream in one of two ways, resulting in two data stream formats called Data Stream A and Data Stream B which are defined below.

8.1.1 Data Stream A

The part of the data stream representing the object definitions and common content portions consists of descriptors representing any layout definitions, descriptors representing any logical definitions, and text units representing any common content portions, in that order (see Figure 18).

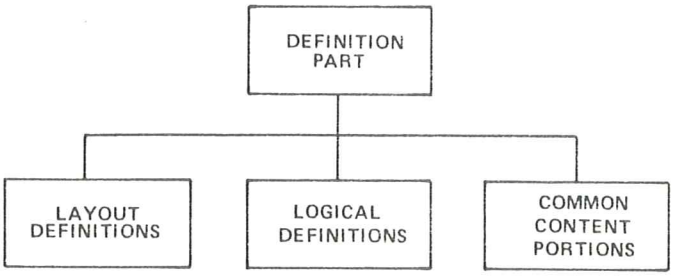


Figure 18: Structure of the Definition Part in Data Stream A

The part of the data stream representing the objects and their content portions consists of descriptors representing any layout objects, descriptors representing any logical objects, and text units representing any associated content portions, in that order (see Figure 19).

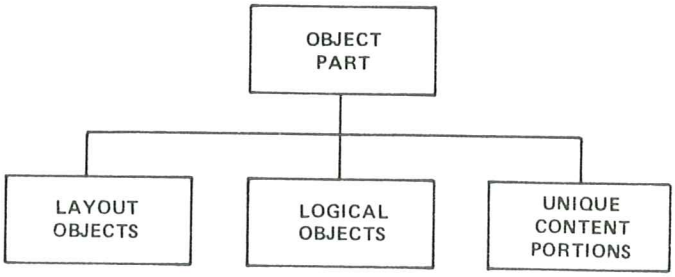


Figure 19: Structure of the Object Part in Data Stream A

The order of the text units within the definition part of the data stream and the order of the text units within the object part are arbitrary.

8.1.2 Data Stream B

Each descriptor of a basic layout object or basic logical object, or of a definition for a basic layout or logical object, is followed immediately by the text units (if any) representing the content of the basic object concerned.

Data stream B can be used when the data stream consists of either logical descriptors and text units or layout descriptors and text units.

Figure 20 shows an example of data stream B.

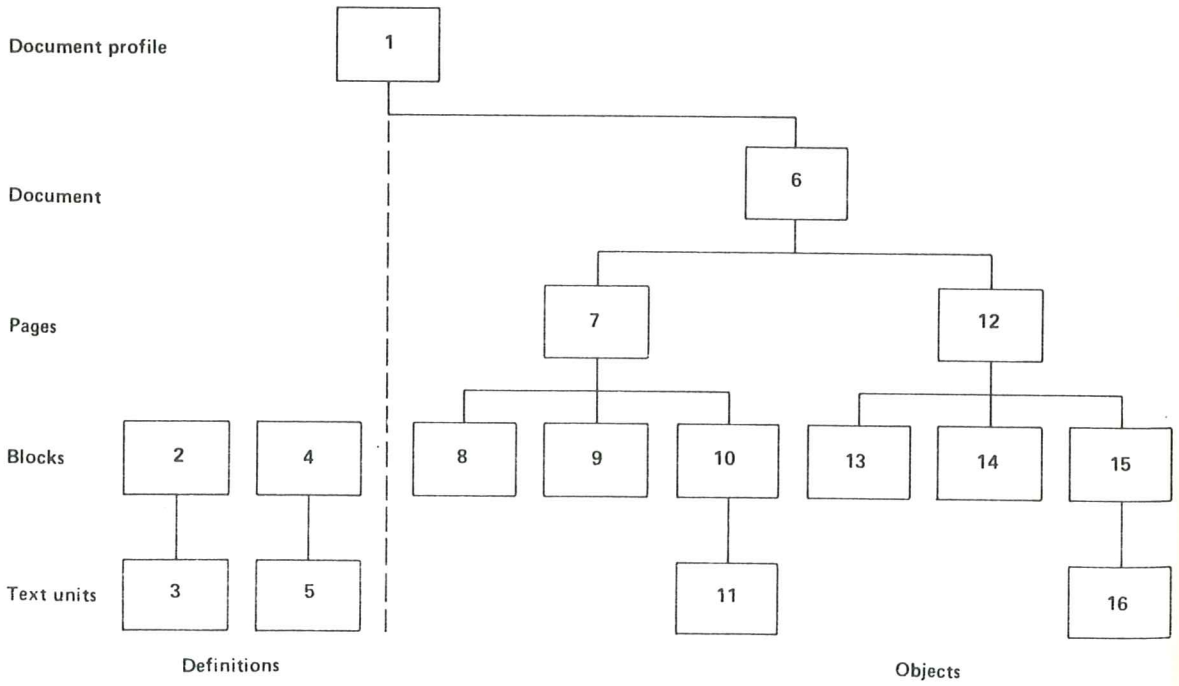


Figure 20: Example of Data Stream B

8.1.3 Descriptors and Text Units

A descriptor consists of subordinate data structures and data items representing the attributes of the document profile, the object definition or the object concerned.

The document profile, each object definition and each object is represented by one descriptor. The document body is represented by up to four descriptors since it acts as the source for the layout definitions, the logical definitions, the layout structure and the logical structure.

A text unit consists of two main parts:

i) an attribute field, i.e. a data structure consisting of subordinate data structures and data items representing the attributes of the content portion concerned;

ii) an information field, i.e. data structure that is either a data item or a set of data items representing the graphic elements making up the content portion concerned.

Each content portion is represented by one text unit.

The data formats of the data stream and its subordinate data structures are specified in 8.3, using the abstract syntax notation ASN.1 defined in ISO 8824. This notation is a formal description method that allows data types relevant to an application to be specified in terms of other data types, including basic data types such as 'integer' and 'octet string' which are defined in ISO 8824 itself.

The coded representation of each data structure or data item is implied by its data type definition when ASN.1 is used. The encoding rules are summarised in 8.2.

8.2 Coded Representation

This is a summary of the encoding rules for the abstract syntax notation ASN.1 defined in ISO 8825.

The coded representation of each data structure or data item that constitutes, or constitutes part of, a descriptor or a text unit consists of a type field, a length field and a value field.

If the data item concerned is an elementary data item, then the type field specifies the elementary data type, the length field specifies the length of the value field, and the value field represents the value of the data item.

If the data structure or data item concerned is not elementary, then the type field identifies the attribute or group of attributes corresponding to the data structure or data item, the length field specifies the length of the value field, and the value field consists of one or more triplets, each of which is composed of a type field, a length field and a value field, representing the subordinate data structures and data items.

The type field (called 'identifier octets' in ISO 8825) consists of one or more octets. The bits of the first octet are used as follows:

bits 8 and 7: tag class	(00: universal, 01: application, 10: context-specific, 11: private);
bit 6: contents encoding form	(0: simple, 1: structured);
bits 5 to 1: tag number, or multi-octet type field indicator	(00000 to 11110), (11111).

The following tag numbers for universal tags have been assigned in ISO 8824 and ISO 8825:

Built-in Data Types	Defined Data Types
0: End-of-contents	18: Numeric String
1: Boolean	19: Printable String
2: Integer	20: Teletex String
3: Bit String	21: Videotex String
4: Octet String	22: ISO 646 String
5: Null	23: UTC Time
16: Sequence	24: Generalized Time
17: Set	

Data items of type End-of-contents, Boolean, Integer or Null are simple (elementary data items). Sequences and Sets are structured (data structures with subordinate data items). Data items of type Bit String, Octet String or any of the defined data types can be either simple or structured.

The length field consists of one or more octets. It takes one of three forms: short, long and indefinite. The bits of the first octet are used as follows:

Bit 8:	length field form (0: short, 1: long or indefinite);
Bits 7 to 1:	if bit 8=0: number of octets of the value field, if bit 8=1: number of octets of the length field following the first octet, 0000000 indicates the indefinite form of the length field.

A data structure or data item with an indefinite length field must be structured and must be terminated by a delimiter consisting of an End-of-contents (EOC) item. An EOC item consists of two octets: a type field of one octet and a length field of one octet. Both are equal to zero. An EOC item has no value field.

8.3 Data Formats

8.3.1 Data Stream

```
Data-Stream                     ::= CHOICE {
    data-stream-A                       Data-Stream-A,
    data-stream-B                       Data-Stream-B }

Data-Stream-A                   ::= SEQUENCE {
    document-profile                    Document-Profile-Descriptor,
    definition-part                     [0] IMPLICIT Definition-Part OPTIONAL,
    object-part                         [1] IMPLICIT Object-Part OPTIONAL }

Definition-Part                 ::= SEQUENCE {
    layout-definitions                  [0] IMPLICIT SEQUENCEOF Layout-Definition-Descriptor OPTIONAL,
    logical-definitions                 [1] IMPLICIT SEQUENCEOF Logical-Definition-Descriptor OPTIONAL,
    common-content-portions             [2] IMPLICIT SETOF Text-Unit OPTIONAL }

Object-Part                     ::= SEQUENCE {
    layout-objects                      [0] IMPLICIT SEQUENCEOF Layout-Object-Descriptor OPTIONAL,
    logical-objects                     [1] IMPLICIT SEQUENCEOF Logical-Object-Descriptor OPTIONAL,
    unique-content-portions             [2] IMPLICIT SETOF Text-Unit OPTIONAL }

Data-Stream-B                   ::= CHOICE {
    layout-stream-B                     [0] IMPLICIT Layout-Stream-B,
    logical-stream-B                    [1] IMPLICIT Logical-Stream-B }

Layout-Stream-B                 ::= SEQUENCEOF Layout-Stream-Data-Unit

Layout-Stream-Data-Unit         ::= CHOICE {
    document profile                    [0] IMPLICIT Document-Profile-Descriptor,
    layout-definition                   [1] IMPLICIT Layout-Definition-Descriptor,
    layout-object                       [2] IMPLICIT Layout-Object-Descriptor,
    content-portion                     [3] IMPLICIT Text-Unit }

Logical-Stream-B                ::= SEQUENCEOF Logical-Stream-Data-Unit

Logical-Stream-Data-Unit        ::= CHOICE {
    document-profile                    [0] IMPLICIT Document-Profile-Descriptor,
    logical-definition                  [4] IMPLICIT Logical-Definition-Descriptor,
    logical-object                      [5] IMPLICIT Logical-Object-Descriptor,
    content-portion                     [3] IMPLICIT Text-Unit }
```

.3.2 Document Profile Descriptor

```
Document-Profile-Descriptor     ::= SET {
    ref-to-layout-definitions           [0] IMPLICIT Object-or-Definition-Identifier OPTIONAL,
    ref-to-layout-structure             [1] IMPLICIT Object-or-Definition-Identifier OPTIONAL,
    ref-to-logical-definitions          [4] IMPLICIT Object-or-Definition-Identifier OPTIONAL,
    ref-to-logical-structure            [5] IMPLICIT Object-or-Definition-Identifier OPTIONAL,
    presentation-capabilities           [2] IMPLICIT Presentation-Capabilities,
    reference-attributes                [3] IMPLICIT Reference-Attributes OPTIONAL }

Presentation-Capabilities       ::= SET {
    content-types                       [0] IMPLICIT Content-Types,
    conformance-combination             [1] IMPLICIT Conformance-Combination,
    non-basic-presentation-capabs       [2] IMPLICIT Non-Basic-Presentation-Capabs OPTIONAL,
    non-basic-structure-capabs          [3] IMPLICIT Non-Basic-Structure-Capabs OPTIONAL }

Content-Types                   ::= INTEGER {character-content (1),
                                             photographic-content (2),
                                             mixed-content (3) }

Conformance-Combination         ::= INTEGER {tif0 (0), mif1 (1), pif0 (2), pif1 (3), mif2(4),
                                             tpf0 (16), tpf1 (17), mpf1 (18), mff2 (32) }
```

```
Non-Basic-Presentation-Capabs ::= SET {
    graphic-character-sets          [0] IMPLICIT OCTETSTRING OPTIONAL,
                                        --- string of escape sequences
    code-extension-announcers       [5] IMPLICIT OCTETSTRING OPTIONAL,
                                        --- string of escape sequences
    doc-prof-graphic-char-sets      [6] IMPLICIT OCTETSTRING OPTIONAL,
                                        --- string of escape sequences
    usr-comm-graphic-char-sets      [7] IMPLICIT OCTETSTRING OPTIONAL,
                                        --- string of escape sequences
    page-dimensions                 [2] IMPLICIT SETOF Measure-Pair OPTIONAL,
    coding-attributes               [3] IMPLICIT SETOF Coding-Attribute OPTIONAL,
    presentation-attributes         [4] IMPLICIT SETOF Presentation-Attribute OPTIONAL }

Coding-Attribute                ::= CHOICE {
    compression                     [0] IMPLICIT Compression }
                                        --- see 8.3.8

Presentation-Attribute          ::= CHOICE {
    character-path                  [0] IMPLICIT One-of-Four-Angles,
    character-line-progression      [1] IMPLICIT One-of-Two-Angles,
    character-orientation           [2] IMPLICIT One-of-Four-Angles,
    character-spacing               [5] IMPLICIT INTEGER,
    line-spacing                    [6] IMPLICIT INTEGER,
    alignment                       [7] IMPLICIT Alignment,
    graphic-rendition               [8] IMPLICIT Graphic-Rendition,
    character-fonts                [13] IMPLICIT Character-Fonts,
    character-subrepertoire        [15] IMPLICIT Character-Subrepertoire,
    pel-path                        [9] IMPLICIT One-of-Four-Angles,
    photographic-line-progression  [10] IMPLICIT One-of-Two-Angles,
    uniform-resolution             [11] IMPLICIT Uniform-Resolution,
    non-uniform-resolution         [12] IMPLICIT Non-Uniform-Resolution }
                                        --- see 8.3.6

Non-Basic-Structure-Capabs      ::= SET {
    number-of-objects-per-page      [0] IMPLICIT INTEGER OPTIONAL }

Reference-Attributes            ::= SET {
    dates-and-times                 [0] IMPLICIT Dates-and-Times OPTIONAL,
    originators                     [1] IMPLICIT Originators OPTIONAL,
    external-references             [2] IMPLICIT External-References OPTIONAL,
    filing-and-retrieval            [3] IMPLICIT Filing-and-Retrieval OPTIONAL,
    physical-properties             [4] IMPLICIT Physical-Properties OPTIONAL,
    content-attributes              [5] IMPLICIT Content-Attributes OPTIONAL,
    security-attributes             [6] IMPLICIT Security-Attributes OPTIONAL }

Dates-and-Times                 ::= SET {
    document-date                   [0] IMPLICIT Date OPTIONAL,
    last-change-date-and-time       [1] IMPLICIT Time OPTIONAL,
    local-filing-date-and-time      [2] IMPLICIT Time OPTIONAL,
    expiry-date                     [3] IMPLICIT Date OPTIONAL }

Originators                     ::= SET {
    organizations                   [0] IMPLICIT SETOF Character-Data OPTIONAL,
    preparer                        [1] IMPLICIT Character-Data OPTIONAL,
    owners                          [2] IMPLICIT SETOF Character-Data OPTIONAL,
    authors                         [3] IMPLICIT SETOF Character-Data OPTIONAL,
    copyright                       [4] IMPLICIT Character-Data OPTIONAL,
    status                          [5] IMPLICIT Character-Data OPTIONAL,
    user-specific-codes             [6] IMPLICIT SETOF Character-Data OPTIONAL,
    circulation-list                [7] IMPLICIT SETOF Character-Data OPTIONAL }

External-References             ::= SET {}
                                    -- None in this version of the Standard

Filing-and-Retrieval            ::= SET {
    title                           [0] IMPLICIT Character-Data OPTIONAL,
    version-number                  [1] IMPLICIT Character-Data OPTIONAL,
    keywords                        [3] IMPLICIT SETOF Character-Data OPTIONAL,
    reference                       [4] IMPLICIT Character-Data OPTIONAL }

Physical-Properties             ::= SET {}
                                    -- None in this version of the Standard
```

```
Content-Attributes              ::= SET {
    oda-version-date                [0] IMPLICIT Date OPTIONAL,
    languages                       [1] IMPLICIT SETOF Character-Data OPTIONAL,
    document-class                  [2] IMPLICIT Character-Data OPTIONAL }

Security-Attributes             ::= SET {}
                                    -- None in this version of the Standard

Character-Data                  ::= [APPLICATION 3] IMPLICIT OCTET STRING
                                    -- string of characters according to the attribute
                                    -- "document profile graphic character sets"

Date                            ::= [APPLICATION 4] IMPLICIT PrintableString
                                    -- string of characters representing
                                    -- a date in accordance with ISO 2014

Time                            ::= [APPLICATION 5] IMPLICIT PrintableString
                                    -- string of characters representing
                                    -- a date and a time in accordance
                                    -- with ISO 2014 and ISO 3307
```

8.3.3 Identifiers and Expressions

An object/definition identifier or a content portion identifier consists of a sequence of numbers (see 6.5.2). Such an identifier is represented by a string of decimal-coded numerals with a SPACE character as a separator between each pair of successive numerals.

```
Content-Portion-Identifier      ::= [APPLICATION 0] IMPLICIT PrintableString

Object-or-Definition-Identifier ::= [APPLICATION 1] IMPLICIT PrintableString

Layout-Category-Name            ::= PrintableString

Construction-Expression         ::= CHOICE {
    sequence-construction           [0] IMPLICIT Term-Sequence,
    aggregate-construction          [1] IMPLICIT Term-Sequence,
    choice-construction             [2] IMPLICIT Term-Sequence }

Term-Sequence                   ::= SEQUENCEOF Construction-Term

Construction-Term               ::= CHOICE {
    construction-factor             [0] Construction-Factor,
    optional-construction-factor    [1] Construction-Factor,
    repetitive-construction-factor  [2] Construction-Factor,
    optional-repetitive-factor      [3] Construction-Factor }

Construction-Factor             ::= CHOICE {
    definition-identifier           Object-or-Definition-Identifier,
    construction-expression         Construction-Expression }

Object-Id-Expression            ::= CHOICE {
    current-id-function             [0] IMPLICIT NULL,
    preceding-id-function           [1] Object-Id-Expression,
    succeeding-id-function          [2] Object-Id-Expression,
    superior-function               [3] Object-Id-Expression }

Cardinal-Expression             ::= CHOICE {
    cardinal-literal                [0] IMPLICIT NumericString,
    increment-application           [1] Cardinal-Expression,
    decrement-application           [2] Cardinal-Expression,
    binding-reference               [4] IMPLICIT Binding-Reference }

Binding-Reference               ::= SET {
    object-reference                Object-Id-Expression,
    binding-name                    PrintableString }

String-Expression               ::= SEQUENCEOF Atomic-String-Expression
```

```
Atomic-String-Expression          ::= CHOICE {
    string-literal                        [0] IMPLICIT OCTETSTRING,
    binding-reference                     [2] IMPLICIT Binding-Reference,
    make-string-application               [3] Cardinal-Expression,
    upper-alpha-application               [4] Cardinal-Expression,
    lower-alpha-application               [5] Cardinal-Expression,
    upper-roman-application               [6] Cardinal-Expression,
    lower-roman-application               [7] Cardinal-Expression }
```

8.3.4 Layout Descriptors

```
Layout-Object-Descriptor          ::= SEQUENCE {
    object-type                           Layout-Object-Type,
    descriptor-body                       Layout-Object-Descr-Body OPTIONAL }

Layout-Object-Type                ::= INTEGER {document (0), page-set (1),
                                              page (2), frame (3), block (4) }

Layout-Object-Descr-Body          ::= SET {
    object-identifier                     Object-or-Definition-Identifier OPTIONAL,
    ref-to-subordinate-objects            [0] IMPLICIT SEQUENCEOF NumericString OPTIONAL,
    ref-to-content-portions               [1] IMPLICIT SEQUENCEOF NumericString OPTIONAL,
    ref-to-object-definition              [2] IMPLICIT Object-or-Definition-Identifier OPTIONAL,
    position                              [3] IMPLICIT Measure-Pair OPTIONAL,
    dimensions                            [4] IMPLICIT Measure-Pair OPTIONAL,
    background-texture                    [5] IMPLICIT Background-Texture OPTIONAL,
    presentation-attributes               [6] IMPLICIT Presentation-Attributes OPTIONAL,
    default-value-lists                   [7] IMPLICIT SETOF Default-Value-List OPTIONAL,
    user-readable-comments                [8] IMPLICIT Comment-String OPTIONAL,
    bindings                              [9] IMPLICIT SETOF Binding-Pair OPTIONAL,
    object-path                          [11] IMPLICIT One-Of-Four-Angles OPTIONAL,
    imaging-order                        [12] IMPLICIT SEQUENCEOF NumericString OPTIONAL }

Measure-Pair                      ::= SEQUENCE {
    x-measure                             Measure,
    y-measure                             Measure }

Measure                           ::= CHOICE {
    fixed-measure                         [0] IMPLICIT INTEGER }

Background-Texture                ::= INTEGER {colourless transparent (0), white opaque (1) }

Comment-String                    ::= OCTETSTRING
                                      -- string of characters from the sets designated by
                                      -- the document profile attribute
                                      -- "user comment graphic character sets"

Binding-Pair                      ::= SET {
    binding-name                          [0] IMPLICIT PrintableString,
    binding-value                         CHOICE {
                                             [1] IMPLICIT Object-Identifier-Expression,
                                             [2] IMPLICIT Cardinal-Expression,
                                             [3] IMPLICIT String-Expression } }

Layout-Definition-Descriptor      ::= SEQUENCE {
    object-type                           Layout-Object-Type,
    descriptor-body                       Layout-Definition-Descr-Body }

Layout-Definition-Descr-Body      ::= SET {
    definition-identifier                 Object-or-Definition-Identifier,
    generator-for-subordinates            Generator OPTIONAL,
    ref-to-content-portions               [1] IMPLICIT SEQUENCEOF NumericString OPTIONAL,
    position                              [3] IMPLICIT Measure-Pair OPTIONAL,
    dimensions                            [4] IMPLICIT Measure-Pair OPTIONAL,
    background-texture                    [5] IMPLICIT Background-Texture OPTIONAL,
    presentation-attributes               [6] IMPLICIT Presentation-Attributes OPTIONAL,
    default-value-lists                   [7] IMPLICIT SETOF Default-Value-List OPTIONAL,
    user-readable-comments                [8] IMPLICIT Comment-String OPTIONAL,
    bindings                              [9] IMPLICIT SETOF Binding-Pair OPTIONAL,
    content-generator                    [10] IMPLICIT String-Expression OPTIONAL,
    object-path                          [11] IMPLICIT One-Of-Four-Angles OPTIONAL,
    permitted-categories                 [13] IMPLICIT SETOF Layout-Category-Name OPTIONAL }
```

```
Generator                       ::= CHOICE {
                                      [0] IMPLICIT Simple-Construction,
                                      [2] Construction-Expression }

Simple-Construction             ::= SEQUENCEOF NumericString
```

8.3.5 Logical Descriptors

```
Logical-Object-Descriptor       ::= SEQUENCE {
    object-type                       Logical-Object-Type,
    descriptor-body                   Logical-Object-Descr-Body OPTIONAL }

Logical-Object-Type             ::= INTEGER {document (0),
                                             composite-logical-object (1),
                                             basic-logical-object (2) }

Logical-Object-Descr-Body       ::= SET {
    object-identifier                 Object-or-Definition-Identifier OPTIONAL,
    ref-to-subordinate-objects        [0] IMPLICIT SEQUENCEOF NumericString OPTIONAL,
    ref-to-content-portions           [1] IMPLICIT SEQUENCEOF NumericString OPTIONAL,
    ref-to-object-definition          [2] IMPLICIT Object-or-Definition-Identifier OPTIONAL,
    layout-directives                 [4] IMPLICIT Layout-Directives OPTIONAL,
    presentation-attributes           [5] IMPLICIT Presentation-Attributes OPTIONAL,
    default-value-lists               [6] IMPLICIT SETOF Default-Value-List OPTIONAL,
    user-readable-comments            [7] IMPLICIT Comment-String OPTIONAL,
    property-attributes               [8] IMPLICIT Property-Attributes OPTIONAL,
    bindings                          [9] IMPLICIT SETOF Binding-Pair OPTIONAL }

Layout-Directives               ::= SET {
    indivisible                       CHOICE {
                                         [0] IMPLICIT Object-or-Definition-Identifier,
                                         [1] IMPLICIT Layout-Category-Name } OPTIONAL,
    leading-separation                [2] IMPLICIT INTEGER OPTIONAL,
    trailing-separation               [3] IMPLICIT INTEGER OPTIONAL,
    left-offset                       [4] IMPLICIT INTEGER OPTIONAL,
    right-offset                      [5] IMPLICIT INTEGER OPTIONAL,
    top-offset                        [6] IMPLICIT INTEGER OPTIONAL,
    bottom-offset                     [7] IMPLICIT INTEGER OPTIONAL,
    concatenation                     [8] IMPLICIT Concatenation OPTIONAL,
    new-layout-object                 CHOICE {
                                         [9] IMPLICIT Object-or-Definition-Identifier,
                                        [10] IMPLICIT Layout-Category-Name } OPTIONAL,
    layout-reference                 [11] IMPLICIT Object-or-Definition-Identifier OPTIONAL,
    required-category                [12] IMPLICIT Layout-Category-Name OPTIONAL }

Concatenation                   ::= INTEGER {concatenated (0), non-concatenated (1) }

Property-Attributes             ::= SET {
    protection-flag                   [0] IMPLICIT BOOLEAN OPTIONAL }
                                      -- other property attributes to be defined

Logical-Definition-Descriptor   ::= SEQUENCE {
    object-type                       Logical-Object-Type,
    descriptor-body                   Logical-Definition-Descr-Body }

Logical-Definition-Descr-Body   ::= SET {
    definition-identifier             Object-or-Definition-Identifier,
    generator-for-subordinates        Generator OPTIONAL,
    ref-to-content-portions           [1] IMPLICIT SEQUENCEOF NumericString OPTIONAL,
    layout-directives                 [4] IMPLICIT Layout-Directives OPTIONAL,
    presentation-attributes           [5] IMPLICIT Presentation-Attributes OPTIONAL,
    default-value-lists               [6] IMPLICIT SETOF Default-Value-List OPTIONAL,
    user-readable-comments            [7] IMPLICIT Comment-String OPTIONAL,
    property-attributes               [8] IMPLICIT Property-Attributes OPTIONAL,
    bindings                          [9] IMPLICIT SETOF Binding-Pair OPTIONAL,
    content-generator                [10] IMPLICIT String-Expression OPTIONAL }
```

8.3.6 Presentation Attributes

```
Presentation-Attributes          ::= SET {
    content-type                     Content-Type OPTIONAL,
    content-architecture             Content-Architecture OPTIONAL,
    character-attributes             [0] IMPLICIT Character-Attributes OPTIONAL,
    photographic-attributes          [1] IMPLICIT Photographic-Attributes OPTIONAL }
                                     -- other sets of rendition attributes to be specified

Content-Type                     ::= [APPLICATION 2] IMPLICIT INTEGER {characters (0),
                                                                       photographic (1)}
                                     -- other values to be specified

Content-Architecture             ::= SET {
    character-content                [0] IMPLICIT INTEGER {ci1 (1), ci2 (2), ci3 (3),
                                                           cp0 (4), cp3 (5), cf3 (6) }

    photographic-content             [1] IMPLICIT INTEGER {ph0 (0), ph1 (1) } }

Character-Attributes             ::= SET {
    character-path                   [0] IMPLICIT One-Of-Four-Angles OPTIONAL,
    line-progression                 [1] IMPLICIT One-Of-Two-Angles OPTIONAL,
    character-orientation            [2] IMPLICIT One-Of-Four-Angles OPTIONAL,
    initial-offset                   [3] IMPLICIT Measure-Pair OPTIONAL,
    character-spacing                [6] IMPLICIT INTEGER OPTIONAL,
    line-spacing                     [7] IMPLICIT INTEGER OPTIONAL,
    alignment                        [8] IMPLICIT Alignment OPTIONAL,
    line-layout-table                [9] IMPLICIT Line-Layout-Table OPTIONAL,
    graphic-rendition               [10] IMPLICIT Graphic-Rendition OPTIONAL,
    character-fonts                 [12] IMPLICIT Character-Fonts OPTIONAL,
    graphic-char-subrepertoire      [13] IMPLICIT Character-Subrepertoire OPTIONAL,
    first-line-indentation          [14] IMPLICIT INTEGER OPTIONAL,
    widow-size                      [15] IMPLICIT INTEGER OPTIONAL,
    orphan-size                     [16] IMPLICIT INTEGER OPTIONAL }

One-Of-Four-Angles               ::= INTEGER {d0 (0), d90 (1), d180 (2), d270 (3) }

One-Of-Two-Angles                ::= INTEGER {d90 (1), d270 (3) }

Alignment                        ::= INTEGER {left-aligned (0), right-aligned (1),
                                              centred (2), justified (3) }

Line-Layout-Table                ::= SETOF Tabulation-Stop

Tabulation-Stop                  ::= SET {
    tabulation-position              [0] IMPLICIT INTEGER,
    alignment-type                   [1] IMPLICIT INTEGER
                                                  {left-aligned (0), right-aligned (1),
                                                   centred (2), aligned-around (3) } OPTIONAL,
    alignment-character-string       [2] IMPLICIT OCTETSTRING OPTIONAL }

Character-Subrepertoire          ::= INTEGER

Graphic-Rendition                ::= SETOF Graphic-Rendition-Aspect

Graphic-Rendition-Aspect         ::= INTEGER {reset (0),
                                              increased-intensity (1), decreased-intensity (2),
                                              italicized (3), underlined (4), crossed-out (9),
                                              primary-default-font (10),
                                              first-alternative-font (11),
                                              second-alternative-font (12),
                                              third-alternative-font (13),
                                              fourth-alternative-font (14),
                                              fifth-alternative-font (15),
                                              sixth-alternative-font (16),
                                              seventh-alternative-font (17),
                                              eight-alternative-font (18),
                                              ninth-alternative-font (19), doubly-underlined (21),
                                              normal-intensity (22), not-italicized (23),
                                              not-underlined (24), not-crossed-out (29) }
```

```
Character-Fonts                 ::= SET {
    primary-default-font                [0] IMPLICIT Font OPTIONAL,
    first-alternative-font              [1] IMPLICIT Font OPTIONAL,
    second-alternative-font             [2] IMPLICIT Font OPTIONAL,
    third-alternative-font              [3] IMPLICIT Font OPTIONAL,
    fourth-alternative-font             [4] IMPLICIT Font OPTIONAL,
    fifth-alternative-font              [5] IMPLICIT Font OPTIONAL,
    sixth-alternative-font              [6] IMPLICIT Font OPTIONAL,
    seventh-alternative-font            [7] IMPLICIT Font OPTIONAL,
    eight-alternative-font              [8] IMPLICIT Font OPTIONAL,
    ninth-alternative-font              [9] IMPLICIT Font OPTIONAL }


Font                            ::= SET {
    font-size                           [0] IMPLICIT INTEGER,
    font-design                         [1] IMPLICIT INTEGER }

Photographic-Attributes         ::= SET {
    pel-path                            [0] IMPLICIT One-Of-Four-Angles OPTIONAL,
    line-progression                    [1] IMPLICIT One-Of-Two-Angles OPTIONAL,
    resolution                          CHOICE {
       uniform-resolution                  [2] IMPLICIT Uniform-Resolution,
       non-uniform-resolution              [4] IMPLICIT Non-Uniform-Resolution} OPTIONAL,
    initial-offset                      [3] IMPLICIT Measure-Pair OPTIONAL }

Uniform-Resolution              ::= INTEGER {p180 (0), p200 (1), p240 (2), p300 (3),
                                             p400 (4), p600 (5), p1200(6)}

Non-Uniform-Resolution          ::= SEQUENCE {
    horizontal-resolution               INTEGER,
    vertical-resolution                 INTEGER }
```

8.3.7 Default Value Lists

```
Default-Value-List              ::= CHOICE {
    page-set-attributes                 [1] IMPLICIT Page-Set-Attributes,
    page-attributes                     [2] IMPLICIT Page-Attributes,
    frame-attributes                    [3] IMPLICIT Frame-Attributes,
    block-attributes                    [4] IMPLICIT Block-Attributes,
    composite-logical-attributes        [5] IMPLICIT Composite-Logical-Attributes,
    basic-logical-attributes            [6] IMPLICIT Basic-Logical-Attributes}

Page-Set-Attributes             ::= SET {
    reference-to-object-definition      < Attribute OPTIONAL }

Page-Attributes                 ::= SET {
    reference-to-object-definition      < Attribute OPTIONAL,
    dimensions                          < Attribute OPTIONAL,
    presentation-attributes             < Attribute OPTIONAL }

Frame-Attributes                ::= SET {
    reference-to-object-definition      < Attribute OPTIONAL,
    position                            < Attribute OPTIONAL,
    dimensions                          < Attribute OPTIONAL,
    background-texture                  < Attribute OPTIONAL,
    object-path                         < Attribute OPTIONAL }

Block-Attributes                ::= SET {
    reference-to-object-definition      < Attribute OPTIONAL,
    position                            < Attribute OPTIONAL,
    dimensions                          < Attribute OPTIONAL,
    background-texture                  < Attribute OPTIONAL,
    presentation-attributes             < Attribute OPTIONAL }

Composite-Logical-Attributes    ::= SET {
    reference-to-object-definition      < Attribute OPTIONAL,
    indivisible                         < Attribute OPTIONAL,
    new-layout-object                   < Attribute OPTIONAL,
    protection                          < Attribute OPTIONAL,
    required-category                   < Attribute OPTIONAL }
```

```
Basic-Logical-Attributes          ::= SET {
    reference-to-object-definition    < Attribute OPTIONAL,
    presentation-attributes           < Attribute OPTIONAL,
    concatenation                     < Attribute OPTIONAL,
    leading-separation                < Attribute OPTIONAL,
    trailing-separation               < Attribute OPTIONAL,
    left-offset                       < Attribute OPTIONAL,
    right-offset                      < Attribute OPTIONAL,
    top-offset                        < Attribute OPTIONAL,
    bottom-offset                     < Attribute OPTIONAL,
    protection                        < Attribute OPTIONAL,
    indivisible                       < Attribute OPTIONAL,
    new-layout-object                 < Attribute OPTIONAL,
    required-category                 < Attribute OPTIONAL }

Attribute                         ::= CHOICE {
    reference-to-object-definition    Object-or-Definition-Identifier,
    position                          [0] IMPLICIT Measure-Pair,
    dimensions                        [1] IMPLICIT Measure-Pair,
    background-texture                [2] IMPLICIT Background-Texture,
    presentation-attributes           [3] IMPLICIT Presentation-Attributes,
    object-path                       [4] IMPLICIT One-Of-Four-Angles,
    indivisible                       CHOICE {
                                         [8] IMPLICIT Object-or-Definition-Identifier,
                                         [9] IMPLICIT Layout-Category-Name },
    leading-separation                [10] IMPLICIT INTEGER,
    trailing-separation               [11] IMPLICIT INTEGER,
    left-offset                       [12] IMPLICIT INTEGER,
    right-offset                      [13] IMPLICIT INTEGER,
    top-offset                        [14] IMPLICIT INTEGER,
    bottom-offset                     [15] IMPLICIT INTEGER,
    concatenation                     [16] IMPLICIT Concatenation,
    new-layout-object                 CHOICE {
                                         [17] IMPLICIT Object-or-Definition-Identifier,
                                         [18] IMPLICIT Layout-Category-Name },
    protection                        [19] IMPLICIT BOOLEAN,
    required-category                 [20] IMPLICIT Layout-Category-Name }
```

8.3.8 Text Units

```
Text-Unit                         ::= SEQUENCE {
    content-portion-attributes        Content-Portion-Attributes OPTIONAL,
    text-information                  Text-Information }

Content-Portion-Attributes        ::= SET {
    content-portion-identifier        Content-Portion-Identifier OPTIONAL,
    alt-content-portion-identifier    Content-Portion-Identifier OPTIONAL,
    type-of-coding                    [0] IMPLICIT Type-of-Coding OPTIONAL,
    coding-attributes                 CHOICE {
       iso6937-attributes                [1] IMPLICIT ISO6937-Attributes,
       t6-attributes                     [2] IMPLICIT T6-Attributes } OPTIONAL,
                                      -- other sets of coding attributes to be specified
    alternative-representation        [3] IMPLICIT Comment-String OPTIONAL }

Type-Of-Coding                    ::= INTEGER {iso6937 (0), t6 (1) }
                                      -- other values to be specified

ISO6937-Attributes                ::= SET {}
                                      -- none in this version of the Standard

T6-Attributes                     ::= SET {
    number-of-pels-per-line           [0] IMPLICIT INTEGER OPTIONAL,
    number-of-lines                   [1] IMPLICIT INTEGER OPTIONAL,
    compression                       [2] IMPLICIT Compression OPTIONAL,
    number-of-pels-discarded          [3] IMPLICIT INTEGER OPTIONAL }

Compression                       ::= INTEGER {uncompressed (0), compressed (1) }

Text-Information                  ::= CHOICE {
                                      TeletexString,
                                      OCTETSTRING }
```

9. CONFORMANCE

9.1 Principles

Conformance to this ECMA Standard is defined by a number of conformance combinations.

Each conformance combination is defined in terms of conformance requirements for:

- document profile;
- document architecture;
- interchange format;
- content architectures.

This section of the Standard defines one or more conformance levels in each of these areas and specifies the conformance combinations of this Standard.

The conformance requirements specified in this section of the Standard apply to the interchange of documents. Conformance requirements for sending and receiving equipment are not within the scope of this Standard. However, the purpose of defining conformance combinations is to meet the requirement for reducing diversity in equipment and assisting with verification of equipment conformance.

The definition of a document architecture level or of a content architecture level includes the definition of a set of attributes for use at that level. For each attribute, the following characteristics are defined in so far as they apply:

- the applicability (mandatory, non-mandatory, defaultable);
- the basic and non-basic values;
- the standard default value if any.

A basic value is a value that is always permitted to be used at the conformance level concerned; a non-basic value is a value that is conditionally permitted, the conditions being that its use in a given document is noted in the profile of that document.

The conformance combinatiions are the permitted combinations of document architecture levels, content architectures, interchange formats and document profile levels: they are defined in Table 6.

9.1.1 Document Architecture Level

This specifies the architectural features that are used in the conformance level concerned, for example which sets of logical objects and definitions, sets of layout objects and definitions and which attributes are used.

9.1.2 Content Architectures

This specifies the content architecture or the combination of content architectures that are used in the conformance level

concerned. In particular, the set of presentation attributes and when applicable the set of graphic elements and the set of control functions that are used.

9.1.3 Interchange Formats

This specifies the data stream formats that are used to represent documents interchanged in the conformance level concerned, and how the descriptions and content portions are ordered in the data stream.

9.1.4 Document Profile Levels

In addition to the document architecture level, the content architectures and the interchange formats, there are conformance levels for the Document Profile.

Contained within the document profile is a group of attributes called reference attributes.

In 6.10 of this Standard there are listed and defined a number of these reference attributes.

Subsets of this list may define the various conformance levels of the document profile.

In the first issue of this Standard two levels are defined:

DPR-0 Document Profile (Reference) level 0
DPR-1 Document Profile (Reference) level 1

DPR-0 does not permit any reference attribute.
DPR-1 allows the use of any number of reference attributes defined in 6.10 of this Standard.

9.1.5 Image, Processible and Formatted Processible Forms

Conformance, as defined in this part of the Standard, is based on the distinction between image form, processible form and formatted processible form of document interchange.

Interchange of a document in image form preserves the image of the text and enables a recipient to present the text in the format defined by the sender.

Interchange of a document in processible form preserves the logical structure of the text and enables a recipient to format and present the text according to the intentions of the sender and/or the conventions of the recipient.

Interchange of a document in formatted processible form preserves both the image and the logical structure of the text and enables a recipient to both present the text in the format defined by the sender, and reformat the text according to the intentions of the sender and/or the conventions of the recipient.

This classification into image form, processible form and

formatted processible form applies to all types of content architecture as well as to the document architecture.

9.2 Document Architecture Conformance

Nine document architecture conformance levels are defined - all these are subsets of the document architecture defined in sections 3, 4, 5 and 6.1 - 6.9 of this Standard. They consist of:

- three image form document architecture levels (with layout structure, without logical structure) identified as:

  IDA0, IDA1 and IDA2;

- four processible form document architecture levels (with logical structure, without layout structure) identified as:

  PDA0, PDA1, PDA2 and PDA3;

- two formatted processible form document architecture levels (with both logical structure and layout structure) identified as:

  FPDA0 and FPDA1.

Apart from the default values of certain attributes:

- IDA0 is a functional subset of IDA1, which is a functional subset of IDA2;

- The features of PDA0 are a functional subset of PDA1, which is a subset of PDA2, which in turn is a functional subset of PDA3;

- FPDA0 is a functional subset of FPDA1;

- IDA2 and PDA3 are functional subsets of FPDA1;

- IDA2 and PDA2 are functional subsets of FPDA0.

Table 1 identifies the nine document architectural levels.

They are classified as:

- Imaging Document Architecture (IDA);
- Processible Document Architecture (PDA);
- Formatted Processible Document Architecture (FPDA).

By reference to the two structures and two sets of definitions of the Office Document Architecture, the table specifies:

i) which objects or object definitions can be used at each document architectural level;

ii) the degree of completeness of the provision of the object

definitions at each document architecture level.

Table 6 Document Architecture Levels

CLASS	LEVEL	LOGICAL		LAYOUT	
		DEFINITIONS	OBJECTS	DEFINITIONS	OBJECTS
IDA	0	----------	--------	----------	DOCUMENT BASIC PAGES
	1	----------	--------	ALL OPTIONAL	DOCUMENT, BASIC PAGES and/or COMPOSITE PAGES and BLOCKS
	2	----------	--------	ALL OPTIONAL	DOCUMENT, {PAGE SETS}, PAGES, {FRAMES}, BLOCKS
PDA	0	----------	DOCUMENT, BASIC OBJECTS	----------	----------
	1	ALL MANDATORY	DOCUMENT, {COMPOSITE OBJECTS}, BASIC OBJECTS	----------	----------
	2	ALL MANDATORY	as PDA1	ALL OPTIONAL	----------
	3	ALL MANDATORY	as PDA1	ALL MANDATORY	----------
FPDA	0	ALL MANDATORY	as PDA1	ALL OPTIONAL	as IDA2
	1	ALL MANDATORY	as PDA1	ALL MANDATORY	as IDA2

In Table 6 above note the following.

i) These levels indicate increasing levels of completeness of the interchanged document in terms of structures and sets of definitions incorporated in the interchange format.

ii) {} means that the object types may or may not be present in any particular document.

iii) "ALL MANDATORY" for logical definitions means that there must exist a definition corresponding to every logical object in the document.

iv) "ALL MANDATORY" for layout definitions means that sufficient layout definitions exist to determine a layout structure for the document.

v) "ALL OPTIONAL" for layout definitions means that some layout definitions may be present.

9.3 Content Architecture Conformance

Table 7 illustrates eight content architectures, they are classified as:

- CI - Character Content Architecture for Image form;
- CP - Character Content Architecture for Processible form;
- CF - Character Content Architecture for Formatted Processible form;
- PH - Photographic Content Architecture.

Each content architecture comprises graphic information and control information embedded in the text.

The different character content architectures are differentiated by the amount of control information and correspond to different groups of applications.
The content architectures consist of:

- three image form character content architecture levels, identified as CI1, CI2 and CI3;

- two processible form character content architecture levels, identified as CP0 and CP3;

- one formatted processible character content architecture level, identified as CF3.

The presentation attributes and control functions of:

- CI1 are a subset of CI2, which in turn are a subset of CI3;

- CP0 is a subset of CP3;

- CI3 and CP3 are subsets of CF3.

These character content architectures themselves are defined in section 7; this part defines the permissable values, and the default values, of the presentation attributes and the control function parameters.

Conformance requirements for photographic content architectures consist of two levels, identified as PH0 and PH1.

Additional clauses specifying conformance for other types of content architecture will be included when these content architectures are defined in future versions of this Standard.

Table 7 - Content Architectures

Content Architecture	Control Functions Used
CI1	Carriage Return, Line Feed and Space only
CI2	ISO 6937/3 Level 2 or CCITT Recommendation T.61 (Teletex)
CI3	All shared and layout control functions as specified in 7.2
CP0	Carriage Return, Line Feed and Space only. (CCITT Recommendation X.420, SFD)
CP3	All shared and logical control functions specified in 7.2
CF3	All control functions specified in 7.2
PH0	CCITT Recommendation T.6 (Group 4 Facsimile)
PH1	CCITT Recommendation T.6 (For use in mixed mode as defined in T.73)

Sub-repertoires of the graphic character repertoire of ISO 6937 may be used, in accordance with the registration procedure defined in ISO 7350. Two such sub-repertoires are of particular interest, the Teletex sub-repertoire, and the minimum (Telex) sub-repertoire.

9.4 Interchange Format Conformance

Two levels of interchange format are identified:

- Data stream A;
- Data stream B.

Data stream A and data stream B are defined in 8.1 of this Standard. Data stream A allows the representation of a document with both logical and layout structures.

Data stream B is compatible with CCITT Recommendations T.73 and X.420 (SFD).

9.5 Conformance Combinations

Table 8 defines nine conformance combinations and specifies the document architecture level, content architectures, interchange formats and document profile level for each conformance combination.

Specification of the conformance combinations is a mandatory attribute of the document profile.

The conformance combinations are classified into three groups as:

i) Image Format;
ii) Processible Format;
iii) Formatted Processible Format.

This numbering is used in the following table.

Table 8 - Conformance Combinations

Classif-cation	Conformance Combinations	Document Architecture Level	Content Architectures	Interchange Format	Reference Attribute Level
i)	PIF0	IDA0	CI1	B	DPR-0
	PIF1	IDA0	CI2	B	DPR-0
	TIF0	IDA0	PH0	B	DPR-0
	MIF1	IDA1	CI2,PH1	B	DPR-0
	MIF2	IDA2	CI3,PH1	B	DPR-1
ii)	TPF0	PDA0	CP0	B	DPR-0
	TPF1	PDA1	CP3	B	DPR-1
	MPF1	PDA3	CP3,PH1	B	DPR-1
iii)	MFF2	FPDA1	CF3,PH1	A	DPR-1

Note: Acronyms

PIF = Page Image Format
TIF = Text Image Format
MIF = Mixed Image Format
TPF = Text Processible Format
MPF = Mixed Processible Format
MFF = Mixed Formatted Processible Format

9.6 Correspondence of Document Architecture Levels and External Standards

Conformance level PIF1 is functionally equivalent to CCITT Recommendation T.61 (Basic Teletex). Conformance level TPF0 is functionally equivalent to CCITT Recommendation X.420 (Simple Formattable Documents).

Conformance level TIF0 is identical to Interchange Format TIF.0 as defined in CCITT Recommendation T.73. The photographic information is coded as in Facsimile group 4 (PH0). Conformance level MIF1 is identical to Interchange Format TIF.1 as defined in CCITT Recommendation T.73.

9.7 Document Architecture Conformance

The specification of every document architecture level that corresponds to external standards is included in the following set of tables.

These specify the objects and definitions and attributes of the sets of logical objects and definitions and layout objects and definitions.

These tables represent the document architecture levels IDA0, IDA1, PDA0.

The other document architecture levels provide the full set of objects, definitions and attributes defined in this Standard and adopt the default values as defined with the attribute definitions.

This section contains two tables for each document architecture level.

i) The objects and attributes applicable. In this table the attributes are qualified as:

* M (mandatory,
* NM (non-mandatory),
* D (defaultable).

The terms mandatory, non-mandatory and defaultable are as defined in 6.2.

M* is used to mean:

> Mandatory unless specified by corresponding object definition or by the attribute "default value lists" at a higher hierarchical level.

ii) The specification of attribute values that have, at a particular document architecture level, defined basic or non-basic values, or standard default values other than those specified in section 6.

IDA0 / 1

Attributes of the layout objects that are applicable for document architecture level IDA0.

Object Type	Attribute Name	Qualifier
Document	Object type	M
	Object identifier	NM
	Reference to subordinate objects	NM
	Default value lists	NM
Page	Object type	M
	Object identifier	NM
	Dimensions	D
	Reference to content portions	NM
	Presentation attributes	D

IDA0 / 2

Attributes that have basic or non-basic values defined, or that have standard default values defined, other than those specified with the attribute definitions: particular to document architecture level IDA0.

Object type ; Attribute Name	Values		
	Basic	Non-basic	Standard default
Page; Dimensions	width ≤ 9920 BMU height ≤ 14030 BMU	(a) width = 10200 BMU height = 13200 BMU (b) width = 14030 BMU height = 19840 BMU	width = 9920 BMU height = 14030 BMU

The non-basic value (a) for page dimensions is the interchange image area for North American letter paper size.

The non-basic value (b) for page dimensions is the interchange image area for ISO A3 paper size.

IDA1 / 1

Attributes of the layout definitions and objects that are applicable for document architecture level IDA1.

Object Type	Attribute Name	Qualifier	
		Object	Definition
Document	Object type	M	M
	Object identifier	NM	--
	Definition identifier	--	M
	Reference to subordinate objects	NM	--
	Generator for subordinate objects (simple constructions only)	--	NM
	Reference to object definition	NM	--
	Default value lists	NM	NM
	User-Readable Comments	NM	NM
Composite Page	Object type	M	M
	Object identifier	NM	--
	Definition identifier	--	M
	References to subordinate objects	NM	--
	Generator for subordinate objects (simple constructions only)	--	NM
	Reference to object definition	NM	--
	Dimensions	D	NM
	Default value lists	NM	NM
	User-Readable Comments	NM	NM

IDA1 / 1 ... / continued
Attributes of the layout definitions and objects that are applicable for document architecture level IDA1.

Object Type	Attribute Name	Qualifier	
		Object	Definition
Basic Page	Object type	M	M
	Object identifier	NM	--
	Definition identifier	--	M
	Reference to object definition	NM	--
	Reference to content portions	NM	NM
	Dimensions	D	--
	User-Readable Comments	NM	NM
	Presentation attributes	M*	NM
Block	Object type	M	M
	Object identifier	NM	--
	Definition identifier	--	M
	Reference to object definition	NM	--
	References to content portions	NM	NM
	Position	D	--
	Dimensions	D	--
	Background Texture	D	--
	User-Readable Comments	NM	NM
	Presentation attributes	M*	NM

IDA1 / 2

Attributes that have basic or non-basic values defined, or that have standard default values defined other than those specified with the attribute definitions: particular to document architecture level IDA1.

Object type ; Attribute Name	Values		
	Basic	Non-basic	Standard default
Composite or Basic Page; Dimensions	width $\leq$ 9920 BMU height $\leq$ 14030 BMU	width $\leq$ 10200 BMU height $\leq$ 13200 BMU	width =9240 BMU height=13200 BMU
Block;Position	X , Y BMU		0 , 0 BMU
Block;Dimensions	X , Y BMU		Page Dimensions
Block;Background Texture	Colourless Transparent		Colourless Transparent

Note. The non-basic value for page dimensions is the interchange image area for North American letter paper size.

PDA0 / 1
Attributes of the logical objects that are applicable for document architecture level PDA0.

Object Type	Attribute Name	Qualifier
Document	Object type	M
	References to subordinate objects	NM
	Default value lists	NM
Basic Logical Object	Object type	M
	References to content portions	NM
	Separation (leading)	D
	Offset (left)	D
	Presentation attributes	D

9.8 Character Content Architecture Conformance

This subsection presents detailed conformance requirements for the content architectures defined in section 7 of this Standard.

The conformance requirements for each content architecture are specified in three sub-sections.

The first sub-section contains a table specifying the permissible values of the presentation attributes applicable to that content architecture.

The second sub-section contains a table specifying the permissible values of the control function parameters.
The third sub-section lists the control functions that do not have parameter values associated with them.

Each table in the first two sub-sections contains four columns:

- the first column specifies the name of the presentation attribute or control function;
- the second column specifies its permissible basic values;
- the third column specifies its default values;
- the fourth column specifies its permissible non-basic values.

All presentation attributes in these tables are defaultable.

9.8.1 Character Content Architecture CI1

9.8.1.1 Presentation Attributes

Attribute	Values		
	Basic	Default	Non-basic
Character Sub-repertoire	None	The identifier of the sub-repertoire of ISO 6937/2 corresponding to the IRV of ECMA-6	None

9.8.1.2 Control Functions With Parameters

None apply to this content architecture

9.8.1.3 Control Functions Without Parameters

CR - CARRIAGE RETURN
LF - LINE FEED
SP - SPACE

9.8.2 Character Content Architecture CI2

9.8.2.1 Presentation Attributes

Attribute	Values		
	Basic	Default	Non-basic
Alignment	Left Aligned	Left Aligned	None
Character Orientation	0°	0°	90°
Character Path	0°, 90°	0°	270°
Character Spacing	120 BMU	120 BMU	80,100,200 BMU
Character Sub-repertoire	Teletex sub-repertoire of ISO 6937/2	Teletex sub-repertoire of ISO 6937/2	Any registered sub-repertoire (see 7.2.3)
Graphic Rendition	0 , 4	0	None
Initial Offset	Any	See 7.2.3	None
Line Progression	270°	270°	None
Line Spacing	100,200,300,400 BMU	200 BMU	150 BMU

9.8.2.2 Control Functions With Parameters

Control Function	Values		
	Basic	Default	Non-basic
SELECT CHARACTER SPACING (SHS)	0	0	1,2,3
SELECT LINE SPACING (SVS)	0,1,2,3	0	4
SELECT GRAPHIC RENDITION (SGR)	0,4	0	None
IDENTIFY GRAPHIC SUB-REPERTOIRE (IGS)	0	0	Any registered sub-repertoire

9.8.2.3 Control Functions Without Parameters

BS - BACKSPACE
CR - CARRIAGE RETURN
LF - LINE FEED
SP - SPACE
SUB - SUBSTITUTE CHARACTER
PLD - PARTIAL LINE DOWN
PLU - PARTIAL LINE UP

9.8.3 Character Content Architecture CI3

9.8.3.1 Presentation Attributes

Attribute	Values		
	Basic	Default	Non-basic
Alignment	Left Aligned Right Aligned Centred Justified	Left Aligned	None
Character Orientation	0°	0°	90°, 180°, 270°
Character Path	0°, 90°	0°	180°, 270°
Character Spacing	100, 120 BMU	120 BMU	Any
Character sub-repertoire	Teletex sub-repertoire of ISO 6937/2	Teletex sub-repertoire of ISO 6937/2	Any registered sub-repertoire
Graphic Rendition	0,1,3,4,9, 10-19 inclusive 21,22,23,24,29	0	2
Initial Offset	Any	See Part 7	None
Line Progression	270°	270°	90°
Character Fonts	None	Implementation defined	Any registered font
Line Layout Table	Any	None	None
Line Spacing	100,200,300,400 BMU	200 BMU	Any

9.8.3.2 Control Functions With Parameters

Control Function	Values		
	Basic	Default	Non-basic
SELECT CHARACTER SPACING (SHS)	0, 1	0	2,3
SELECT LINE SPACING (SVS)	0,1,2,3	0	4
SELECT GRAPHIC RENDITION (SGR)	0,1,3,4,9, 10-19 inclusive, 21,22,23,24,29	0	2
IDENTIFY GRAPHIC SUBREPERTOIRE (IGS)	0	0	Any registered sub-repertoire
CHARACTER POSITION RELATIVE(HPR)	Any	0	None
SPACING INCREMENT(SPI)	Any	See 7.2.3	None
CHARACTER POSITION BACKWARD(HPB)	Any	0	None

9.8.3.3 Control Functions Without Parameters

BS - BACKSPACE
CR - CARRIAGE RETURN
EM - END OF MEDIUM
HT - HORIZONTAL TABULATION
LF - LINE FEED
SP - SPACE
SUB - SUBSTITUTE CHARACTER
PLD - PARTIAL LINE DOWN
PLU - PARTIAL LINE UP

9.8.4 Character Content Architecture CP0

9.8.4.1 Presentation Attributes

Attribute	Values		
	Basic	Default	Non-basic
Alignment	Left aligned Centred Justified	Left aligned	None
Graphic Rendition	0, 4	0	None

9.8.4.2 Control Functions With Parameters

None apply to this content architecture.

9.8.4.3 Control Functions Without Parameters

CR - CARRIAGE RETURN
LF - LINE FEED
SP - SPACE

9.8.5 Character Content Architecture CP3

9.8.5.1 Presentation Attributes

Attribute	Values		
	Basic	Default	Non-basic
Alignment	Left Aligned Right Aligned Centred Justified	Left Aligned	None
Character Fonts	None	Implementation defined	Any registered font
Character Orientation	0°	0°	90°, 180°, 270°
Character Path	0°, 90°	0°	180°, 270°
Character Spacing	100, 120 BMU	120 BMU	Any
Character Sub-repertoire	Teletex sub-repertoire of ISO 6937/2	Teletex sub-repertoire of ISO 6937/2	Any registered sub-repertoire
First Line Indentation	Any	0	None
Graphic Rendition	0,1,3,4,9, 10-19 inclusive 21,22,23,24,29	0	2
Line Progression	270°	270°	90°
Line Layout Table	Any	None	None
Line Spacing	100,200,300,400 BMU	200 BMU	Any
Orphan Size	Any	1	None
Widow Size	Any	1	None

9.8.5.2 Control Functions With Parameters

Control Function	Values		
	Basic	Default	Non-basic
SELECT CHARACTER SPACING (SHS)	0, 1	0	2,3
SELECT LINE SPACING (SVS)	0,1,2,3	0	4
SELECT GRAPHIC RENDITION(SGR)	0,1,3,4,9, 10-19 inclusive, 21,22,23,24,29	0	2
IDENTIFY GRAPHIC SUB-REPERTOIRE (IGS)	0	0	Any registered sub-repertoire
SPACING INCREMENT(SPI)	Any	See 7.2.3	None

9.8.5.3 Control Functions Without Parameters

BPH - BREAK PERMITTED HERE
CR - CARRIAGE RETURN
HT - HORIZONTAL TABULATION
LF - LINE FEED
NBH - NO BREAK HERE
SP - SPACE
SUB - SUBSTITUTE CHARACTER
PLD - PARTIAL LINE DOWN
PLU - PARTIAL LINE UP

9.8.6 Character Content Architecture CF3

9.8.6.1 Presentation Attributes

These include all those attributes that are applicable to CP3 (see 9.8.5.1) with the addition of the following:

Attribute	Values		
	Basic	Default	Non-basic
Initial Offset	Any	See 7.2.1	None

9.8.6.2 Control Functions With Parameters

These include all those functions that are applicable to CP3 (see 9.8.5.2) with the addition of the following:

Control Function	Values		
	Basic	Default	Non-basic
CHARACTER POSITION RELATIVE(HPR)	Any	0	None
CHARACTER POSITION BACKWARD(HPB)	Any	0	None

9.8.6.3 Control Functions Without Parameters

These include all those functions that are applicable to CP3 (see 9.8.5.3) with the addition of the following:

BS - BACKSPACE
DCS - DEVICE CONTROL STRING
EM - END OF MEDIUM
ST - STRING TERMINATOR

9.8.7 Summary of Presentation Attributes Applicable to the defined Character Content Architectures

Presentation Attribute	Content Architectures					
	CI1	CI2	CI3	CP0	CP3	CF3
Alignment	-	X	X	X	X	X
Character Fonts	-		X	-	X	X
Character Orientation	-	X	X	-	X	X
Character Path	-	X	X	-	X	X
Character Spacing	-	X	X	-	X	X
Graphic Character Sub-repertoire	X	X	X	-	X	X
First Line Indentation	-	-	-	-	X	X
Graphic Rendition		X	X	X	X	X
Initial Offset		X	X			X
Line Layout Table			X		X	X
Line Progression		X	X		X	X
Line Spacing		X	X		X	X
Orphan Size					X	X
Widow Size					X	X

9.8.8 Summary of Control Functions Applicable to the defined Character Content Architectures

Control Functions	Content Architectures					
	CI1	CI2	CI3	CP0	CP3	CF3
BS - BACKSPACE		X	X			X
BPH - BREAK PERMITTED HERE					X	X
CR - CARRIAGE RETURN	X	X	X	X	X	X
DCS - DEVICE CONTROL STRING						X
EM - END OF MEDIUM			X			X
HPB - CHARACTER POSITION BACKWARD			X			X
HPR - CHARACTER POSITION RELATIVE			X			X
HT - HORIZONTAL TABULATION			X		X	X
IGS - IDENTIFY GRAPHIC SUBREPERTOIRE		X	X		X	X
LF - LINE FEED	X	X	X	X	X	X
NBH - NO BREAK HERE					X	X
PLD - PARTIAL LINE DOWN		X	X		X	X
PLU - PARTIAL LINE UP		X	X		X	X
SGR - SELECT GRAPHIC RENDITION		X	X		X	X
SHS - SELECT CHARACTER SPACING		X	X		X	X
SP - SPACE	X	X	X	X	X	X
SPI - SPACING INCREMENT			X		X	X
ST - STRING TERMINATOR						X
SUB - SUBSTITUTE CHARACTER		X	X		X	X
SVS - SELECT LINE SPACING		X	X		X	X

9.9 Photographic Content Architecture Conformance

9.9.1 Photographic Content Architecture PH0

9.9.1.1 Presentation Attributes

Attribute	Values		
	Basic	Default	Non-basic
Pel Path	0°	0°	None
Line Progression	270°	270°	None
Resolution - Uniform	200 pels per 25,4 mm.	200 pels per 25,4 mm	240, 300, 400 pels per 25,4 mm

9.9.1.2 Coding Attributes

Attribute	Values		
	Basic	Default	Non-basic
Number of Pels per Line	Any	See note 1	None
Number of Pels Discarded	Any	See note 2	None
Compression	Compressed as in T.6	Compressed as in T.6	Uncompressed as in T.6

Notes (1) The default number of pels per line is, depending on the value of the presentation attribute "resolution" : 1728 (200), 2074 (240), 2592 (300) or 3456 (400).

(2) If the number of pels per line exceeds the number of pels that fits across the width of the basic layout object (as specified by the layout attribute "dimension"), the default number of pels to be discarded is half the excess number of pels. If the number of pels does not exceed the number of pels that fits across the width, the default number of pels to be discarded is zero.

9.9.2 Photographic Content Architecture PH1

9.9.2.1 Presentation Attributes

Attribute	Values		
	Basic	Default	Non-basic
Initial Offset	Any	0,0	None
Pel Path	0°	0°	None
Line Progression	270°	270°	None
Resolution - Uniform	240, 300 pels per 25,4 mm	None. See note.	180,200,400,600, 1200 pels per 25,4 mm

9.9.2.2 Coding Attributes

Attribute	Values		
	Basic	Default	Non-basic
Number of Pels per Line	Any	None. (See note)	None
Number of Pels Discarded	Any	0	None
Compression	Compressed as in T.6	Compressed as in T.6	Uncompressed as in T.6

Note. The presentation attribute "resolution" and the coding attribute "number of pels per line" are mandatory for PH1.

9.9.3 Summary of Presentation Attributes Applicable to the defined Photographic Content Architectures

Atttribute	Content Architectures	
	PH0	PH1
Initial Offset	-	X
Pel Path	X	X
Line Progression	X	X
Resolution		
- Uniform	X	X
- Non-uniform	-	-

9.9.4 Summary of Coding Attributes Applicable to the defined Photographic Content Architectures

Atttribute	Content Architectures	
	PH0	PH1
Number of Pels per Line	X	X
Number of Pels Discarded	X	X
Number of Lines	-	-
Compression	X	X

APPENDIX A - DEFINITIONS OF CONTROL FUNCTIONS

Code extension control functions and sequence introducers have not been included explicitly in this appendix. Any code extension control function or sequence introducer (eg. CSI) required to represent a graphic character or a control function is permitted to be used in the character content architecture.

A.1 BS - BACKSPACE

A control function which causes the active position to be moved backwards on the same line with an amount equal to the "unit character spacing" for constant spacing, and equal to the width of the character SP as defined by the current font for proportional spacing.

NOTE - The primary purpose of BACKSPACE is to move the active position backwards from the line home position, when needed, after an occurrence of CARRIAGE RETURN. BACKSPACE shall not be used to combine the images of two or more graphic characters unless such application of BACKSPACE forms part of the coding method for the graphic characters of the repertoire in use.

A.2 BPH - BREAK PERMITTED HERE

A control function which indicates a point where a line break may occur when the text is formatted.

A.3 CR - CARRIAGE RETURN

A control function which causes the active position to be moved to the reference point of the line box but not to be moved vertically.

NOTES

1. CARRIAGE RETURN causes a forward movement of the active position when the active position has been moved backwards from the reference point of the line box and has not subsequently been advanced to a position at or in advance of this point.

2. The primary purpose of CARRIAGE RETURN is to move the active position to the reference point of the line box at the beginning of a new line of text. CARRIAGE RETURN is, therefore, normally used in combination with LINE FEED. CARRIAGE RETURN shall not be used to combine the images of two or more graphic characters unless such application of CARRIAGE RETURN forms part of the coding method for the graphic characters of the repertoire in use.

A.4 DCS - DEVICE CONTROL STRING

A control function that acts as the opening delimiter of a control string. The control string is closed by the terminating delimiter STRING TERMINATOR (ST).

A control string may contain occurrences of graphic characters and control functions, in particular CARRIAGE RETURN, LINE FEED and HYPHEN, introduced as a result of a formatting process.

A.5 EM - END OF MEDIUM

A control function which indicates that the current line must not be justified.

A.6 HPB - CHARACTER POSITION BACKWARD

A control function with one numeric parameter which causes the active position to be moved backwards horizontally the number of Basic Measurement Units (BMU) specified by the parameter.

The default value of the parameter is 1.

NOTE - The primary purpose of HPB is to move the active position backwards from the reference point of the line box, when needed, after an occurrence of CARRIAGE RETURN.

A.7 HPR - CHARACTER POSITION RELATIVE

A control function with one numeric parameter which causes the active position to be advanced horizontally the number of Basic Measurement Units (BMU) specified by the parameter.

The default value of the parameter is 1.

NOTE - Although HPR has a control effect similar to that of one or more space characters, it does not have the graphic equivalent of space characters. Therefore, HPR does not cause spaces to be imaged in accordance with the current graphic rendition, such as underlined, possibly specified by a preceding occurrence of SGR.

A.8 HT - HORIZONTAL TABULATION

A control function which specifies the positioning and alignment of subsequent text, until either the next occurrence of another HT or the end of the current line.

This text string is to be positioned at the next tabulation stop along the line, and aligned in accordance with the properties of that tabulation stop.

A.9 IGS - IDENTIFY GRAPHIC SUBREPERTOIRE

A control function with a selective parameter which is used to indicate that a subrepertoire of the graphic character repertoire of ISO 6937 is used in the subsequent text. All graphic character sets that are used to represent the indicated graphic character subrepertoire must be explicitly or implicitly designated, but need not be invoked, prior to the occurrence of IGS.

The identification of the graphic character subrepertoire may be changed at any point within a document and becomes effective

immediately. No graphic characters other than those of the specified sub-repertoire shall be used in the text following the occurrence of IGS. The effect of a graphic character sub-repertoire identification ceases upon the next occurrence of:

i) another IGS;

ii) the end of the current basic object;

iii) the designation of any graphic character set.

The maximum number of parameter values is 1.

The parameter value is the identifier assigned to a sub-repertoire of the repertoire of ISO 6937 in accordance with the registration procedure specified in ISO 7350.

In the absence of IGS, the sub-repertoire identified by the "graphic character sub-repertoire" attribute, if any, applies or the entire repertoire of the currently designated graphic character sets otherwise.

An occurrence of IGS without a parameter value cancels any sub-repertoire identified by either a preceding IGS or the "graphic character sub-repertoire" attribute.

A.10 LF - LINE FEED

A control function which causes the active position to be advanced by a unit line spacing in the direction of line progression but not to be moved in the direction of the character path. The amount of vertical movement is that specified by the most recent occurrence of SPACING INCREMENT or SELECT LINE SPACING, if any, or by the "line spacing" attribute otherwise.

A.11 NBH - NO BREAK HERE

A control function which indicates a point where no line break must occur when the text is formatted. All consecutive SP following NBH are not potential linebreak points.

A.12 PLD - PARTIAL LINE DOWN

A control function which causes either the start of subscript rendition or the end of superscript rendition of graphic characters.

When superscript rendition is in effect, it is terminated by PLD; otherwise, subscript rendition is initiated by PLD.

Any occurrence of PLD to start subscript rendition shall be followed by PLU in the same line before another PLD or LINE FEED is used.

PLD does not affect the position of any lines used to implement the graphic rendition "underlined", "crossed-out" or "doubly

underlined" when such a graphic rendition is in effect prior to the occurrence of PLD.

NOTES

1. The implementation of the subscript rendition initiated by PLD may be accomplished with special character fonts and/or movement of the active position not exceeding half the font size.

2. The graphic rendition "underlined" may have been effected, prior to the occurrence of PLD, either by SELECT GRAPHIC RENDITION or by the non-spacing underline character (see ISO 6937/2).

A.13 PLU - PARTIAL LINE UP

A control function which causes either the start of superscript rendition or the end of subscript rendition of graphic characters.

When subscript rendition is in effect, it is terminated by PLU; otherwise, superscript rendition is initiated by PLU.

Any occurrence of PLU to start superscript rendition shall be followed by PLD in the same line before another PLU or LINE FEED is used.

PLU does not affect the position of any lines used to implement the graphic rendition "underlined", "crossed-out" or "doubly underlined" when such a graphic rendition is in effect prior to the occurrence of PLU.

NOTES

1. The implementation of the superscript rendition initiated by PLU may be accomplished with special character fonts and/or movement of the active position not exceeding half the font size.

2. The graphic rendition "underlined" may have been effected prior to the occurrence of PLU, either by SELECT GRAPHIC RENDITION or by the non-spacing underline character (see ISO 6937/2).

A.14 SGR - SELECT GRAPHIC RENDITION

A control function with a selective parameter which specifies one or more graphic rendition aspects for graphic characters and space characters in the subsequent text. The specified graphic rendition aspects take effect immediately and remain in effect until they are changed by a subsequent occurrence of SGR in the current basic object.

The meaning of the parameter value is:

0: normal intensity, not italicized, not underlined, not crossed out, primary font resets the effect of any preceding

occurrence of SGR and cancels the effect of the "graphic rendition" attribute;
1: bold or increased intensity;
2: faint or decreased intensity;
3: italicized;
4: underlined;
9: crossed-out (characters still legible but marked as to be deleted);
10: primary (default) font;
11: first alternative font;
12: second alternative font;
13: third alternative font;
14: fourth alternative font;
15: fifth alternative font;
16: sixth alternative font;
17: seventh alternative font;
18: eigth alternative font;
19: ninth alternative font;
21: doubly underlined;
22: normal intensity (not bold);
23: not italicized;
24: not underlined (neither singly or doubly);
29: not crossed-out;

The default value of the parameter is 0.

Any graphic rendition aspect specified by an occurrence of SGR, except the default rendition, is combined with the graphic rendition aspects that are in effect prior to that occurrence of SGR as a result of either an earlier occurrence of SGR or the "graphic rendition" attribute of the current basic object.

When SGR is used to start underlining (singly or doubly), or crossing out, within the scope of subscript or superscript image (see PARTIAL LINE DOWN and PARTIAL LINE UP), any lines used to implement such a graphic rendition are lowered or raised in order that the graphic rendition concerned applies to the subscript or superscript characters.

NOTES

1. Several parameter values can be used in combination, in order to obtain, for example, underlined italics.

2. The default parameter value cannot be used in combination with any other parameter value.

3. The parameter values 4 and 21 cannot be used in combination with each other.

4. When "underlined" (parameter value 4) is specified, "doubly underlined" (parameter value 21) is terminated.

5. When "doubly underlined" (parameter value 21) is specified, "underlined" (parameter value 4) is terminated.

A.15 SHS - SELECT CHARACTER SPACING

A control function with a selective parameter which specifies the character spacing for subsequent text. The specified character spacing takes effect immediately only when the character spacing is constant; when proportional spacing is active, the character spacing does not become effective until constant spacing is invoked by an occurrence of SGR. The specified character spacing remains in effect until it is changed by a subsequent occurrence of either SHS or SPI in the current basic object.

The meaning of the parameter value is:

0: 10 characters per 25,4 mm;
1: 12 characters per 25,4 mm;
2: 15 characters per 25,4 mm;
3: 6 characters per 25,4 mm.

The default value of the parameter is 0.

The maximum number of parameter values is 1.

A.16 SPI - SPACING INCREMENT

A control function with two numeric parameters which specifies the line spacing and the character spacing for subsequent text. The specified line spacing takes effect immediately and remains in effect until it is changed by a subsequent occurrence of either SPI or SVS in the current basic object. The specified character spacing takes effect immediately only when the character spacing is constant; when proportional spacing is active, the character spacing does not become effective until constant spacing is invoked by an occurrence of SGR. The specified character spacing remains in effect until it is changed by a subsequent occurrence of either SHS or SPI in the current basic object.

Both the line spacing and the character spacing are expressed as integral multiples of Basic Measurement Units (BMU).

The default value of the first parameter is the current line spacing, ie. that specified by the most recent occurrence of SPACING INCREMENT or SELECT LINE SPACING, if any, or by the "line spacing" attribute otherwise.

The default value of the second parameter is the current character spacing, ie. that specified by the most recent occurrence of SPACING INCREMENT or SELECT CHARACTER SPACING, if any, or by the "character spacing" attribute otherwise.

A.17 ST - STRING TERMINATOR

A control function that acts as the terminating delimiter in a control string opened by DEVICE CONTROL STRING (DCS).

A.18 SUB - SUBSTITUTE CHARACTER

A control function which is used in the place of a character that has been found invalid or in error.

A.19 SVS - SELECT LINE SPACING

A control function with a selective parameter which specifies the line spacing for subsequent text. The specified line spacing takes effect immediately and remains in effect until it is changed by a subsequent occurrence of either SPI or SVS in the current basic object.

The meaning of the parameter value is:

0: 6 lines per 25,4 mm;
1: 4 lines per 25,4 mm;
2: 3 lines per 25,4 mm;
3: 12 lines per 25,4 mm;
4: 8 lines per 25,4 mm;
The default value of the parameter is 0.

The maximum number of parameter values is 1.

APPENDIX B - CODED REPRESENTATIONS OF CONTROL FUNCTIONS

Coded representations of control functions are defined in other standards, viz. ECMA-6, ECMA-48 and ISO 6937.

A summary of the coded representations of the control functions defined in this Standard is given below.

In this table, the symbol P...P denotes a sequence of one or more bit combinations that represent the parameter or parameters of the control function concerned.

Control Function	Coded Representation	Reference Standard
BS - Backspace	0/8	ECMA-6
BPH - Break Permitted Here	8/2	ISO 6937/4*
CR - Carriage Return	0/13	ECMA-6
DCS - Device Control String	9/0	ECMA-48
EM - End of Medium	1/9	ECMA-6
HPB - Character Position Backward	9/11 P...P 6/10	ECMA-48
HPR - Character Position Relative	9/11 P...P 6/1	ECMA-48
HT - Horizontal Tabulation	0/9	ECMA-6
IGS - Identify Graphic Subrepertoire	9/11 P...P 2/0 4/13	ECMA-48
LF - Line Feed	0/10	ECMA-6
NBH - No Break Here	8/3	ISO 6937/4*
PLD - Partial Line Down	8/11	ECMA-48
PLU - Partial Line Up	8/12	ECMA-48
SGR - Select Graphic Rendition	9/11 P...P 6/13	ECMA-48
SHS - Select Character Spacing	9/11 P...P 2/0 4/11	ECMA-48
SPI - Spacing Increment	9/11 P...P 2/0 4/7	ECMA-48
ST - String Terminator	9/12	ECMA-48
SUB - Substitute Character	1/10	ECMA-6
SVS - Select Line Spacing	9/11 P...P 2/0 4/12	ECMA-48

* Currently in preparation by ISO/TC 97/SC 2.

APPENDIX C - EXAMPLES OF DOCUMENT STRUCTURES

This appendix does not form part of the ECMA Standard.

The examples in this appendix consist of a set of logical definitions and a set of layout definitions for a document class called 'report'.

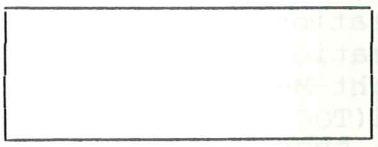
One particular aspect of the logical and layout is emphasized, viz. the construction rules.

The logical construction rules specify that the logical structure of a 'report' consists of a report header, a table of contents and one or more chapters, and specify the subordinate logical structure of each of these major composite logical objects.

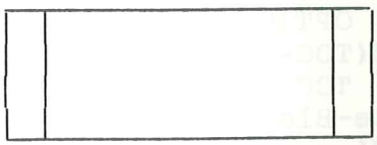
The layout construction rules specify that the layout structure of a 'report' consists of a title page, a page set for the table of contents and one or more page sets for the chapters, and specify the subordinate layout structure of each of these major composite layout objects.

The construction rules are represented in the form of diagrams (Figures C-1 and C-2) and in the form of sets of production rules using a language similar to ASN.1 (sections C.1 and C.2).

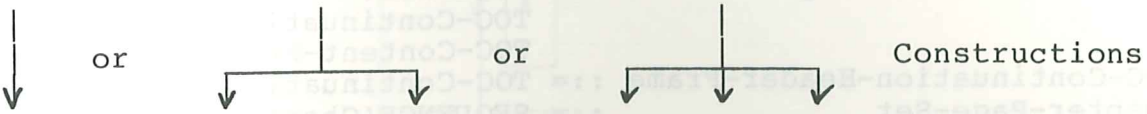
The following symbols are used in the diagrams:



Object definition



Common content portion



SEQ	sequence construction
AGGR	aggregate construction
CHOICE	choice construction
REP	repetitive construction
OPT	optional construction

The production rules use the following symbols (see 6.5.5.2) :

SEQUENCE	sequence construction
SET	aggregate construction
CHOICE	choice construction
SEQUENCE OF	repetitive construction
OPTIONAL	optional construction

C.1 Logical Construction Rules

```
Report                        ::= SEQUENCE(Report-Heading,
                                           Table-of-Contents,
                                           SEQUENCE OF Chapter)
Report-Heading                ::= SEQUENCE(Authors-Name,Title,Date)
Table-of-Contents             ::= SEQUENCE OF Table-of-Contents-Entry
Table-of-Contents-Entry       ::= SEQUENCE(Chapter-Number,
                                           Chapter-Title,
                                           Page-Reference)
Chapter                       ::= SEQUENCE(Chapter-Heading,
                                           Chapter-Summary OPTIONAL,
                                           SEQUENCE OF Section)
Section                       ::= SEQUENCE(Section-Heading,
                                      SEQUENCE OF Section-Element)
Section-Element               ::= CHOICE(Paragraph,Figure)
Figure                        ::= SET(Picture,Caption)
```

C.2 Layout Construction Rules

```
Report                                ::= SEQUENCE(Title-Page,
                                                   TOC-Page-Set,
                                                   SEQUENCE OF
                                                   Chapter-Page-Set)
Title-Page                            ::= SEQUENCE(Authors-Name-Frame,
                                                   Title-Frame,
                                                   Date-Frame,
                                                   Organization-Frame)
Organization-Frame                    ::= SEQUENCE(
                                           Organization-Name-Block,
                                           Organization-Address-Block,
                                           Copyright-Message-Block)
TOC-Page-Set                          ::= SEQUENCE(TOC-First-Page,
                                                   SEQUENCE OF
                                                   TOC-Continuation-Page
                                                   OPTIONAL)
TOC-First-Page                        ::= SEQUENCE(TOC-Title-Frame,
                                                   TOC-Content-Frame)
TOC-Title-Frame                       ::= TOC-Title-Block
TOC-Continuation-Page                 ::= SEQUENCE(
                                          TOC-Continuation-Header-Frame,
                                          TOC-Content-Frame)
TOC-Continuation-Header-Frame         ::= TOC-Continuation-Header-Block
Chapter-Page-Set                      ::= SEQUENCE(Chapter-First-Page,
                                           SEQUENCE OF
                                           Chapter-Continuation-Page
                                           OPTIONAL)
Chapter-First-Page                    ::= SEQUENCE(
                                              Chapter-Heading-Frame,
                                              Chapter-Content-Frame,
                                              Page-Number-Frame)
Chapter-Continuation-Page             ::= SEQUENCE(
                                              Chapter-Content-Frame,
                                              Page-Number-Frame)
```

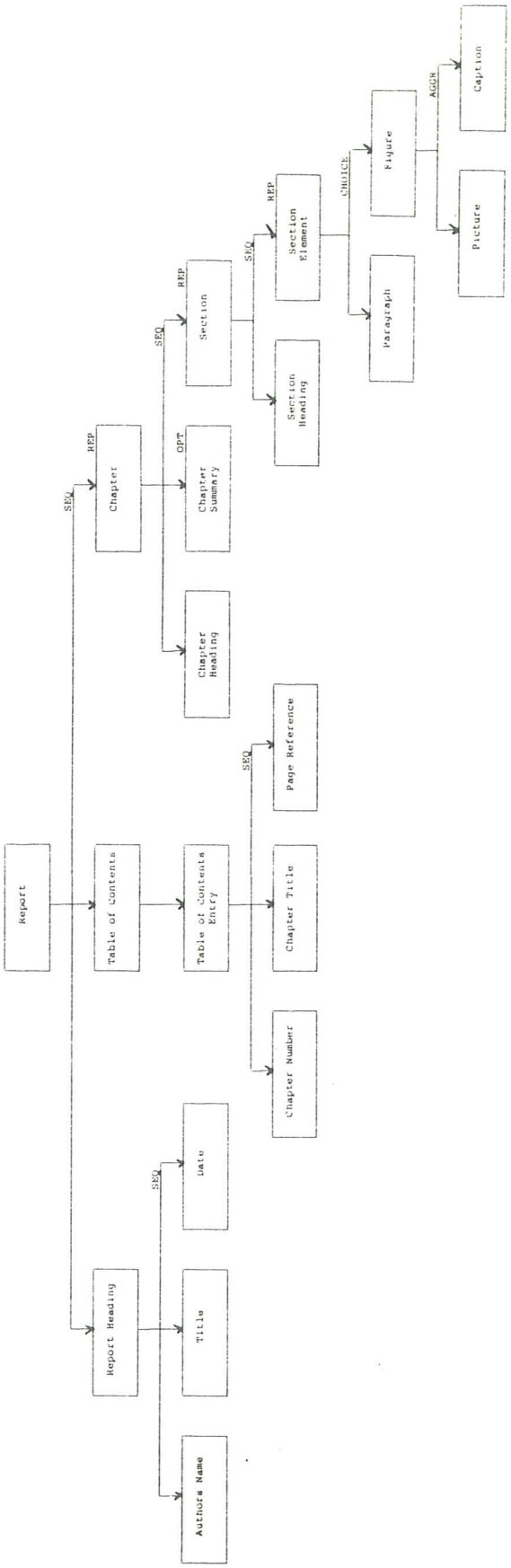


Figure C-1: Logical Construction Rules

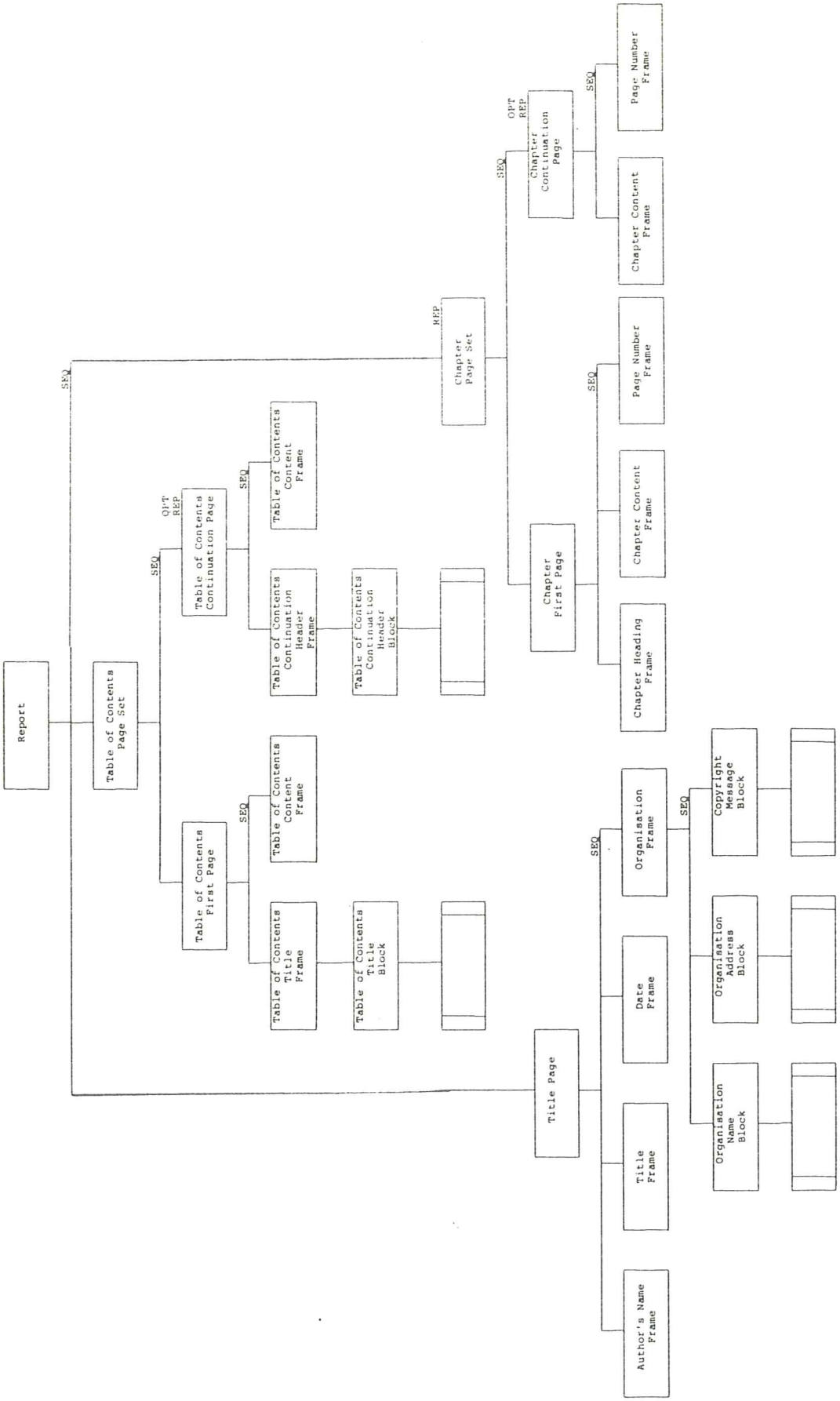


Figure C-2: Layout Construction Rules