INTERNATIONAL STANDARD

ISO/IEC 8613-2

Second edition
1995-04-15

Information technology — Open Document Architecture (ODA) and interchange format: Document structures

Technologies de l'information — Architecture des documents ouverts (ODA) et format d'échange: Structures documentaires

Reference number
ISO/IEC 8613-2:1995(E)

Contents

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

Printed in Switzerland

Page

Page

Page

Foreword

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

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

International Standard ISO/IEC 8613-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 18, *Document processing and related communication*, in collaboration with ITU-T. The identical text is published as ITU-T Rec. T.412.

This second edition cancels and replaces the first edition (ISO 8613-2:1989), which has been technically revised.

ISO/IEC 8613 consists of the following parts, under the general title *Information technology — Open Document Architecture (ODA) and Interchange Format*:

- *Part 1: Introduction and general principles*
- *Part 2: Document structures*
- *Part 3: Abstract interface for the manipulation of ODA documents*
- *Part 4: Document profile*
- *Part 5: Open Document Interchange Format*
- *Part 6: Character content architectures*
- *Part 7: Raster graphics content architectures*
- *Part 8: Geometric graphics content architectures*
- *Part 9: Audio content architectures*
- *Part 10: Formal specifications*
- *Part 11: Tabular structures and tabular layout*
- *Part 12: Identification of document fragments*
- *Part 13: Spreadsheet*
- *Part 14: Temporal relationships and non-linear structures*

Annexes A and H form an integral part of this part of ISO/IEC 8613. Annexes B to G and I to K are for information only.

Introduction

This ITU-T Recommendation | International Standard was prepared as a joint publication by ITU-T Study Group 8 and ISO/IEC Joint Technical Committee 1.

At present, the ITU-T Recs. in the T.410-Series | International Standard ISO/IEC 8613 consist of:

- Introduction and general principles;
- Document structures;
- Document profile;
- Open document interchange formats;
- Character content architectures;
- Raster graphics content architectures;
- Geometric graphics content architectures;
- Formal specification of the Open Document Architecture (FODA).

  (The formal specification is applicable to ISO/IEC 8613 only).

Further Recommendations | International Standards may be added to this series of ITU-T Recommendations | International Standards.

Development of this series of ITU-T Recommendations | International Standards was originally in parallel with the ECMA-101 standard: Open Document Architecture.

This series of ITU-T Recommendations | International Standards is a new edition of the CCITT T.410-Series Recommendations (1988) and ISO 8613 (1989).

Significant technical changes are the inclusion of the following amendments as agreed by ITU-T and ISO/IEC:

- Alternative Representation;
- Annex on use of MHS/MOTIS;
- Colour;
- Conformance Testing Annex;
- Document Application Profile Proforma and Notation;
- Security;
- Streams;
- Styles;
- Tiled Raster Graphics.

In addition, a number of technical corrigenda have been applied.

This ITU-T Recommendation | International Standard contains eleven annexes:

- Annex A – Notation used to represent document structures (integral);
- Annex B – Examples of document structures (non-integral);
- Annex C – Examples of particular document architecture features (non-integral);
- Annex D – The defaulting mechanism (non-integral);
- Annex E – Attribute summary tables (non-integral);
- Annex F – Overview of alternative description, technical and implementation aspects (non-integral);
- Annex G – Further information on security aspects within a document (non-integral);
- Annex H – Conversions between the reference colour space and the interchange spaces (integral);
- Annex I – Definitions of colour terms (non-integral);
- Annex J – Colour concepts (non-integral);
- Annex K – Bibliography on colour(non-integral).

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY — OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT: DOCUMENT STRUCTURES

1 Scope

The purpose of the ITU-T Rec. T.410-Series | ISO/IEC 8613 is to facilitate the interchange of documents.

In the context of these Recommendations | International Standards, documents are items such as memoranda, letters, invoices, forms and reports, which may include pictures and tabular material. The content elements used within the documents may include graphic characters, raster graphics elements and geometric graphics elements, all potentially within one document.

NOTE – These Recommendations | International Standards are designed to allow for extensions, including hypermedia features, spreadsheets and additional types of content such as audio and video.

In addition to the content types defined in these Recommendations | International Standards, ODA also provides for arbitrary content types to be included in documents.

These Recommendations | International Standards apply to the interchange of documents by means of data communications or the exchange of storage media.

These Recommendations | International Standards provide for the interchange of documents for either or both of the following purposes:

- to allow presentation as intended by the originator;
- to allow processing, such as editing and reformatting.

The composition of a document in interchange can take several forms:

- formatted form, allowing presentation of the document;
- processable form, allowing processing of the document;
- formatted processable form, allowing both presentation and processing of the document.

These Recommendations | International Standards also provide for the interchange of ODA information structures used for the processing of interchanged documents.

This ITU-T Recommendation | International Standard:

- defines a document architecture intended for representation of documents;
- defines a document processing model;
- defines the document structures, the basic constituents of the architecture and a descriptive representation of these in terms of attributes;
- defines an interface which allows the use of different content architectures with the document architecture;
- defines the reference model of the document layout process;
- defines the reference model of the document imaging process;
- defines the reference model for protecting parts of a document;
- defines three document architecture classes;
- defines a notation used for illustrating and describing document structures;
- provides examples of document structures;
- provides examples of particular document attributes.

2 Normative references

The following ITU-T Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of ISO and IEC maintain registers of currently valid International Standards. The ITU-T Secretariat maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation T.411 (1993) | ISO/IEC 8613-1:1994, *Information technology – Open Document Architecture (ODA) and interchange format: Introduction and general principles.*
- ITU-T Recommendation T.414 (1993) | ISO/IEC 8613-4:1994, *Information technology – Open Document Architecture (ODA) and interchange format: Document Profile.*
- ITU-T Recommendation T.415 (1993) | ISO/IEC 8613-5:1994, *Information technology – Open Document Architecture (ODA) and interchange format: Open Document Interchange Format.*
- ITU-T Recommendation T.416 (1993) | ISO/IEC 8613-6:1994, *Information technology – Open Document Architecture (ODA) and interchange format: Character content architectures.*
- ITU-T Recommendation T.417 (1993) | ISO/IEC 8613-7:1994, *Information technology – Open Document Architecture (ODA) and interchange format: Raster graphics content architectures.*
- ITU-T Recommendation T.418 (1993) | ISO/IEC 8613-8:1994, *Information technology – Open Document Architecture (ODA) and interchange format: Geometric graphics content architectures.*

2.2 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation X.208 (1988), *Specification of Abstract Syntax Notation One (ASN.1);*

  ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1).*

2.3 Additional references

- ISO 2846:1975, *Set of printing inks for offset printing – Colorimetric characteristics.*
- ISO 3664:1975, *Photography – Illumination conditions for viewing colour transparencies and their reproductions.*
- ISO/IEC 6937:1994, *Information technology – Coded graphic character set for text communication – Latin alphabet.*
- CIE Publication S002:1986, *Colorimetric Observers.*
- CIE Publication 15.2:1986, *Colorimetry.*

  NOTE – CIE = International Commission on Illumination.
- ANSI PH2.30-1985, *Graphic Arts and Photographic Viewing Conditions for Color Prints, Transparencies and Photomechanical Reproduction.*
- SMPTE Recommended Practice RP37:1969, *Color Temperature for Color Television Studio Monitors.*
- SMPTE Recommended Practice RP145:1986, *Color Monitor Colorimetry.*

3 Definitions

For the purpose of this Recommendation | International Standard the definitions given in ITU-T Rec. T.411 | ISO/IEC 8613-1 apply.

NOTE – Definitions of some additional terms relating to colour are in Annex I.

4 Abbreviations

For the purpose of this Recommendation | International Standard, the abbreviations given in ITU-T Rec. T.411 | ISO/IEC 8613-1 apply.

5 Conventions

For the purpose of this Recommendation | International Standard the conventions given in ITU-T Rec. T.411 | ISO/IEC 8613-1 apply.

The following additional conventions are used within this Recommendation | International Standard.

5.1 Subordinate

References to the immediate subordinates of an object always use the form "immediate subordinates" or "immediately subordinate".

When the term "subordinate" is used without this qualifier it means subordinate to any hierarchical level.

5.2 Superior

References to the immediate superior of an object always use the form "immediate superior" or "immediately superior".

The term "nearest superior" is used in conjunction with an object and with a particular qualification to mean the first superior going up the hierarchy which satisfies the qualification, i.e. first checking the immediate superior, then its immediate superior, and then each superior in turn in this hierarchic order. For example, "nearest superior that belongs to the specified object class" means the first superior going up the hierarchy from the object which is of the specified object class.

When the term "superior" is used without either of these qualifiers it means superior to any hierarchical level.

5.3 Next and following

When the term "next" or "immediately following" is used in conjunction with an object then it means the object immediately following this object in sequential order (see 7.1.2). Thus, "next layout object" means the immediately following layout object in sequential layout order; "next logical object" means the immediately following logical object in sequential logical order.

When the term "following" is used in conjunction with an object without the qualifier "immediately" then it means an object at any position later in the sequential order than this object.

In some cases the term "next" is used together with a further qualification, for example, "next layout object with the same layout category" means the first of the following layout objects in sequential layout order which has the same layout category as the specified layout object.

5.4 Preceding

When the term "immediately preceding" is used in conjunction with an object then it means the object immediately preceding this object in sequential order (see 7.1.2). Thus, "immediately preceding layout object" means the immediately preceding layout object in sequential layout order; "immediately preceding logical object" means the immediately preceding logical object in sequential logical order.

When the term "preceding" is used in conjunction with an object without the qualifier "immediately", then it means an object at any position earlier in the sequential order than this object.

In some cases the term "preceding" is used together with a further qualification, for example, "preceding layout object with the same layout category" means the last of the preceding layout objects in sequential layout order which has the same layout category as the specified layout object.

6 Architectural principles

6.1 Architectural concepts

For the purpose of the ITU-T Rec. T.410-Series | ISO/IEC 8613, a document is an amount of structured information that can be interchanged as a unit.

This Specification provides the means to represent the structures of documents in two major forms:

- in a *formatted form* which allows documents to be presented (for example, printed or displayed) as intended by the originator;
- in a *processable form* which allows further processing of documents by the recipient, such as editing and formatting as intended by the originator.

It also provides the means to represent a document in *formatted processable form* to satisfy both purposes.

A document contains information that relates to its content and structure.

The *content* of a document consists of any type of information that is intended for human perception, for example, content elements that can be presented in a two-dimensional form, such as printed on paper or displayed on a screen.

The structural information included in a document is provided in order to:

- delimit portions within a document, such as areas for the imaging of different types of content elements;
- delimit portions of a document that have a logical meaning, such as chapters, paragraphs;
- use different types of coding for the different content types;
- allow processing of the document.

The rules for defining the structure and representation of documents are collectively called the *document architecture*.

The document architecture comprises a structural model and a descriptive representation. The *structural model* describes the structural elements of a document and the relationships among these elements. The *descriptive representation* describes how the elements of a document and the properties of these elements are represented.

The structural model and the descriptive representation present complementary views of a document. Both are necessary in order to distinguish between the structural aspects of a document and the data structures which represent those structural aspects. Also, the descriptive representation shows how additional information not described by the structural model is represented. This information includes styles and the document profile. Styles contain information relating to the layout and presentation of a document (see 6.3.5). The document profile includes information relating to the document as a whole (see 6.3.7).

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

6.2 Structural model of a document

The structural model introduces the *structural elements* of the document architecture.

6.2.1 Specific structures

The structural model of a document provides for two different but complementary views of the content of a specific document:

- the logical view associates content with structural elements such as chapters, appendices, headings, paragraphs, footnotes and figures;
- the layout view associates content with structural elements relating to presentation media, such as pages and areas within pages.

This is illustrated in Figure 1.

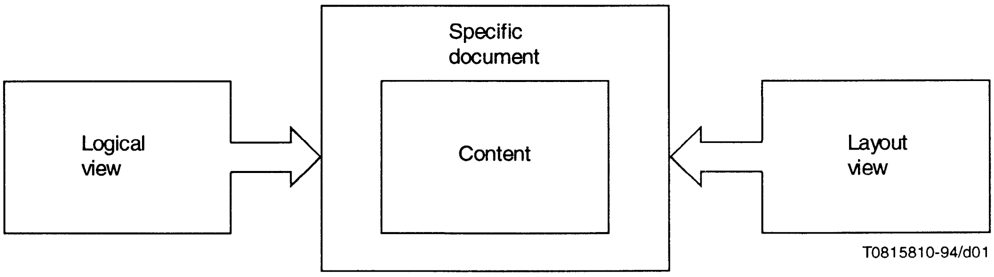


Figure 1 – Views of a document

These structural elements of a specific document are called *objects*. Each view associates the same document content with a separate structure which consists of a hierarchy of objects.

Thus

- the *specific logical structure* associates the content of a document with a hierarchy of *logical objects* and provides for the representation of documents in processable form;
- the *specific layout structure* associates the content of a document with a hierarchy of *layout objects* and provides for the representation of documents in formatted form.

The *specific structure* consists of the specific logical structure and/or the specific layout structure.

The structures are independent of the types of content within a document.

There are two types of relationships among objects in a structure, namely:

- *structural relationships* which specify the hierarchical structure of the objects;
- *non-hierarchical relationships* which specify other relationships, such as cross-references to figures or footnotes.

6.2.2 Content of a document

The structural model of a document partitions the content into structural elements called *content portions*.

The information within each content portion must pertain to a particular type of content and the structure of this information is defined by a content architecture. A *content architecture* consists of the definition of a set of *content elements*, control functions and attributes, with their coded representation, and of the rules for the application of the attributes and control functions to the content elements.

Selection of content architecture depends on the type or types of content elements to be represented. A single document may contain content portions pertaining to different content architectures. There shall be only one content architecture per content portion.

A content architecture does not identify any logical or layout objects of a document. All structural information and all logical and layout objects are specified by the logical and layout structures of the document architecture.

The document architecture supports the incorporation of the content architectures that are specified in other parts of the ITU-T Rec. T.410-Series | ISO/IEC 8613 as well as supporting the incorporation of content architectures for any other types of content. The document architecture has been designed to be separate from the content architectures. This is achieved since the structural model has been designed such that the properties of the structural elements that are used to describe a document are independent of the types of content that may be associated with those structural elements. Both document architecture and one or more content architectures are needed to represent a document.

The interface between the document architecture and the content architecture(s) is defined in clause 8.

This interface allows the use of any of the content architectures defined in other parts of the ITU-T Rec. T.410-Series | ISO/IEC 8613 with the document architecture Other types of content may be used with the document architecture provided a content architecture is defined to this interface.

6.2.3 Generic structures

The generic structure of a document provides for the representation of characteristics common to, and relationships between, a number of objects within a document. In the most comprehensive case it provides for the representation of the common characteristics of a group of documents.

6.2.3.1 Object classes

In many documents, there may be sets of objects with common characteristics; for example:

- logical objects representing sections, consisting of a sequence of subordinate objects representing paragraphs, with the same characteristics;
- pages with the same headers and footers.

An *object class* is a structural element of the document that models such a group of common characteristics.

The ITU-T Rec. T.410-Series | ISO/IEC 8613 does not define particular object classes; however it provides the means by which object classes can be defined.

Any content portions associated with an object class are called *generic content portions*.

For convenience of reference, the term *component* is used to refer collectively to an object or an object class.

6.2.3.2 Document classes

A *document class* is used to represent the common characteristics of a group of documents; for example, a set of reports with common paragraphs and common layout. A document class can be used to maintain consistency of an entire document during editing and/or formatting of the document.

The *generic logical structure* provides for the representation of the common characteristics of logical objects of a document class and the *generic layout structure* provides for the representation of the common characteristics of layout objects of a document class.

The generic logical structure consists of all the logical object classes and associated generic content portions of a document.

The generic layout structure consists of all the layout object classes and associated generic content portions of a document.

The *generic structure* consists of the generic logical structure and/or the generic layout structure.

The ITU-T Rec. T.410-Series | ISO/IEC 8613 does not define particular document classes; however it provides the means by which document classes can be defined.

6.3 Descriptive representation of a document

The descriptive representation introduces the descriptive elements of the document architecture.

For the purpose of interchange, a document is represented as a collection of *constituents*, each of which is a set of attributes.

Within the ITU-T Rec. T.410-Series | ISO/IEC 8613 each *attribute* is identified by a name and has a value, which either represents a characteristic of a structural element or a relationship with other constituents.

The ITU-T Rec. T.410-Series | ISO/IEC 8613 defines these constituents and attributes.

Those constituents that are counterparts to the structural elements are termed *descriptions*.

6.3.1 Constituents of a document

The following *types of constituent* are defined:

- document profile;
- logical object class description;
- layout object class description;
- logical object description;
- layout object description;
- content portion description;
- presentation style;
- layout style;
- sealed document profile description;
- enciphered document profile description;
- pre-enciphered document body part description;
- post-enciphered document body part description.

Each constituent is characterised by its attributes; within a constituent all attribute names are unique.

The *document body* is a term used to describe all the constituents of a document except for the document profile.

6.3.2 Content portion descriptions

Each content portion within a document is characterised by a set of attributes called a *content portion description*.

Any content portion description associated with an object class description (see 6.3.4), for example a logo block or a standard paragraph, is called a *generic content portion description*.

6.3.3 Object descriptions

Each object within a structure is characterised by a set of attributes called an *object description*.

Each attribute has a value and may represent one of the following:

a) a characteristic of the object itself;

b) a structural relationship which specifies a hierarchical relationship between the object and other objects in the same structure;

c) a non-hierarchical relationship between:

- the object and other objects in the same structure;
- the object and other objects in different structures included in the same document;
- the object and object classes included in the same document.

An object description may also be referred to as a *primary description*, in particular when it is required to distinguish between object descriptions and alternative descriptions (see 6.3.6).

6.3.4 Object class descriptions

Each object class within a document is characterised by a set of attributes called an *object class description*.

An object class description for a logical object class is called a *logical object class description* and an object class description for a layout object class is called a *layout object class description*.

In general, each attribute of an object class description specifies a rule to determine the value of a corresponding attribute of an object description for an object of the object class concerned.

Object class descriptions can be used for the following purposes:

a) to improve transmission efficiency;

b) to maintain the internal consistency of a document when it is modified;

c) to facilitate the creation of objects and documents.

Object class descriptions can be used either individually or collectively.

In either case, the object class descriptions may have generic content portion descriptions associated with them.

In the case of individual use, each object class description is used for transmission efficiency and/or to facilitate creation of objects. Each such object class description consists of a set of attributes representing the common characteristics of objects of the object class.

The set of object class descriptions corresponding to this case of individual use is termed a *factor set* of object class descriptions.

In the case of collective use, two situations must be considered.

In the first situation, some or all object class descriptions are related to each other in a particular hierarchical structure. This collection serves to facilitate creation of sets of hierarchically related objects within a specific structure, but need not fully specify all possible specific structures that may be created. Such a collection is called a *partial generator set* of object class descriptions.

In the second situation, all object classes are related to each other such that they fully control the generation of specific structures. This collection serves to maintain consistency of an entire document during editing or formatting of the document. During creation and modification of the document, each logical object class description specifies characteristics of the objects that may be created and how these objects may build up the possible specific logical structures of the entire document. Similarly, during document layout, each layout object class description influences the creation of the possible specific layout structures.

The set of object class descriptions corresponding to this second situation is termed a *complete generator set* of object class descriptions.

A generic logical structure or a generic layout structure whose representation includes a partial generator set of object class descriptions is termed a *partial generic logical structure* or a *partial generic layout structure*, respectively.

A generic logical structure or a generic layout structure whose representation includes a complete generator set of object class descriptions is termed a *complete generic logical structure* or a *complete generic layout structure*, respectively.

6.3.5 Styles

In addition to logical and layout component descriptions, a document may contain a number of layout styles and presentation styles which are distinct from the component descriptions.

A style is a set of attributes which may be referred to from component descriptions. The effect of such a reference is to apply the attributes of the style to the component description which contains the reference. Styles may be derived from other styles. A derived style need only specify the attributes and/or attribute values that differ from those of the style from which it was derived.

A *layout style* is a set of attributes referred to from one or more logical component descriptions which during document layout guides the creation of a specific layout structure. That is, a layout style provides information for the creation of pages and, if required, guides the division of pages into separate areas, for the layout of the document content.

A *presentation style* is a set of attributes referred to from one or more basic logical and/or layout component descriptions which guides the format and appearance of the content of the document on the presentation medium.

The separation of styles from the component descriptions allows the layout and presentation of a document to be modified without affecting the logical structure.

6.3.6 Alternative descriptions

In addition to its primary description (see 6.3.3), a basic logical object or a basic layout object may be represented by one or more alternative descriptions. An *alternative description* is intended by the originator to be used by the recipient of a document in lieu of the primary description when the recipient is not capable of processing the primary description. In the case that there are multiple alternative descriptions, a preference order is defined between these alternative descriptions. An alternative description for a basic object may specify the same or a different set of attributes from those specified by the primary description.

Alternative descriptions provide fallback mechanisms in this Specification. Possible uses include: fallback content architecture classes (e.g. providing a raster image as a fallback for geometric graphics), compatibility with several versions of ODA, compatibility with several document application profiles, and fallback for non-basic values within a document application profile.

NOTE – Document application profiles are defined in ITU-T Rec. T.411 | ISO/IEC 8613-1.

Using an alternative description for an object implies using a different set of associated content portion descriptions, if any. An alternative description for an object in conjunction with any associated content portion descriptions is collectively called an *alternative subtree*. The primary description in conjunction with its associated content portion descriptions is called the *primary subtree*. Thus when required a primary subtree may be replaced by an alternative subtree.

A fallback for a content portion description can be supplied by providing an alternative description for the basic object with which the content portion description is associated, and associating the fallback content portion description with this alternative description.

Alternative subtrees must satisfy the constraint that substituting the alternative subtree for its primary subtree shall result in a valid document for the purposes of performing a layout process and/or an imaging process.

6.3.7 Document profile

The *document profile* consists of a set of attributes which specify characteristics of the document as a whole.

For instance, the document profile indicates which of the following are present in the document:

- logical object descriptions;
- layout object descriptions;
- logical object class descriptions;
- layout object class descriptions;
- presentation styles;
- layout styles.

The document profile specifies the document architecture class used in the document. It also specifies the ODA version, document application profile, content architectures and interchange format class used in the document.

The document profile may describe the document and its history, including information for filing and retrieval and, for example, describing the fonts used in the document.

For the convenience of the recipient, the document profile may duplicate information usually found in the document content (for example, document name, author, date, etc.). However, the document profile does not include data specific to a particular mode of transmission, such as mail, message or Teletex.

A document profile may be interchanged alone to allow a sender to test the capability of the recipient, or for the recipient to get information about the document without sending the complete document.

A complete specification of the document profile is contained in ITU-T Rec. T.414 | ISO/IEC 8613-4.

6.3.8 Protected part descriptions

A *protected part description* is a set of attributes which may be referred to from the document profile.

There are four kinds of protected part descriptions:

- sealed document profile description;
- enciphered document profile description;
- pre-enciphered document body part description;
- post-enciphered document body part description.

6.3.9 Document class descriptions

A document class is specified by a *document class description*.

A document class description consists of one of the following:

a) a document profile;
a complete generator set of logical object class descriptions;
optionally layout styles;
optionally presentation styles;
optionally generic content portion descriptions; or

b) a document profile;
a complete generator set of layout object class descriptions;
optionally presentation styles;
optionally generic content portion descriptions; or

c) a document profile;
a complete generator set of logical object class descriptions;
a complete generator set of layout object class descriptions;
optionally layout styles;
optionally presentation styles;
optionally generic content portion descriptions.

A document class description defines how to generate an entire specific logical and/or specific layout structure for any document of the class.

Documents may reference one externally specified document class description (see 6.3.11). In this case, the document profile indicates whether an external-document class description is referenced by the document, and if so, which one.

6.3.10 Generic-documents

A *generic-document* description consists of one of the following:

a) a document profile;
a complete generator, partial generator or factor set of logical object class descriptions;
optionally layout styles;
optionally presentation styles;
optionally generic content portions; or

b) a document profile;
a complete generator, partial generator or factor set of layout object class descriptions;
optionally presentation styles;
optionally generic content portions; or

c) a document profile;
a complete generator, partial generator or factor set of logical object class descriptions;
a complete generator, partial generator or factor set of layout object class descriptions;
optionally layout styles;
optionally presentation styles;
optionally generic content portions.

A generic-document may be interchanged and used to aid in the generation of documents. Resource-documents and external-documents are generic-documents.

6.3.11 External-documents

A document that does not contain a generic structure can refer to an *external-document.* The external-document is identified in the document profile of the document that is interchanged. The reference is ignored if any generic layout structure or generic logical structure is present in the interchanged document. Styles may be present in both the interchanged document and the external-document.

The external-document can provide any or all of :

- constituents representing a complete generic logical structure;
- constituents representing a complete generic layout structure;

and optionally,

- layout styles;
- presentation styles.

An external-document contains a document profile which supplies information for the constituents and styles in the external-document, for example, a fonts list. The information in this document profile cannot be used by the interchanged document except by reference to the constituents and styles in the external-document.

If layout or presentation styles in the interchanged document and the external-document have the same identifier then the style in the interchanged document is used and the style in the external-document is ignored.

6.3.12 Resource-documents

An object class description in a document that is interchanged may contain a reference to an object class description external to the document, in a generic-document. The latter generic-document is called the *resource-document* of the interchanged document.

The object class descriptions contained in the resource-document act as models for object class descriptions in the interchanged document.

The attributes constituting an object class description in the resource-document may supply values for the corresponding attributes of those object class descriptions in the interchanged document that refer to this object class description in the resource-document.

A resource-document may contain generic content portion descriptions to be included by reference into an interchanged document referring to this resource-document.

Thus, the relationship between an object class description in the interchanged document and the corresponding object class description in the resource-document is similar to the relationship between an object description and the corresponding object class description in the interchanged document.

Styles may be present in both the interchanged document and the resource-document. If styles in the interchanged document and the resource-document have the same identifier then references from the resource-document are to the style in the resource-document, and references from the interchanged document are to the style in the interchanged document.

A resource-document is separate from any document or documents referring to it.

6.3.13 Classes of document architecture

A *document architecture class* is a set of rules for defining the structure and representation of documents in formatted form, processable form, or formatted processable form.

Clause 13 defines three document architecture classes that may be used in conjunction with document application profiles, as defined in ITU-T Rec. T.411 | ISO/IEC 8613-1.

The three classes of document architecture are:

a) *Formatted document architecture class* which allows for document content to be presented as intended by the originator; for example, printed or displayed. A document of this class includes a document profile and constituents representing a specific layout structure. It may also include constituents representing a generic layout structure and presentation styles.

b) *Processable document architecture class* which allows for document content to be processed; for example, edited or formatted. A document of this class includes a document profile and constituents representing a specific logical structure. It may also include constituents representing a generic logical structure, a generic layout structure, layout styles and presentation styles.

c) *Formatted processable document architecture class* which allows for document content to be processed as well as presented as intended by the originator. A document of this class includes a document profile, constituents representing a specific logical structure, a specific layout structure and a generic layout structure. The generic layout structure can be omitted if there is a reference to an external-document containing at least a complete generic layout structure. The document may also include constituents representing a generic logical structure, layout styles and presentation styles.

A generic-document may be assigned to one of the preceding classes, by the following rules:

a) If the generic-document contains logical object class descriptions, and no layout object class descriptions, it is assigned to the processable document architecture class.

b) If the generic-document contains layout object class descriptions, and no logical object class descriptions, it is assigned to the formatted document architecture class.

c) If the generic-document contains both logical and layout object class descriptions, it is assigned to the formatted processable document architecture class.

6.3.14 Sets of constituents

A document is represented by constituents, which are grouped into sets of constituents, and which have interrelationships, as defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613.

The possible types of constituent in the descriptive representation of a document are shown in Figure 2.

Figure 2 illustrates that

a) the document consists of one document profile and optionally a number of constituents that form the document body;

b) the document body consists of one of the following sets:

   1) constituents representing the generic part of the document;

   2) constituents representing the style constituents of the document;

   3) constituents representing the specific part of the document;

   4) constituents representing protected parts of the document.

c) the constituents representing the generic part of the document consist of constituents representing the generic logical structure and/or constituents representing the generic layout structure;

d) the style constituents of a document consist of layout styles and/or presentation styles;

e) the constituents representing the specific part of the document consist of constituents representing the specific logical structure and/or constituents representing the specific layout structure;

f) the constituents representing the generic logical structure consist of the logical object class descriptions and any associated generic content portion descriptions;

g) the constituents representing the generic layout structure consist of the layout object class descriptions and any associated generic content portion descriptions;

h) the constituents representing the specific logical structure consist of the logical object descriptions and any associated content portion descriptions;

j) the constituents representing the specific layout structure consist of the layout object descriptions and any associated content portion descriptions;

k) if both the specific logical structure and specific layout structure are present in a document, then the content portions associated with these structures are, in general, common to both;

l) the constituents representing the protected parts of the document consist of sealed document profile descriptions and/or enciphered document profile descriptions and/or pre-enciphered document body part descriptions and/or post-enciphered document body part descriptions.

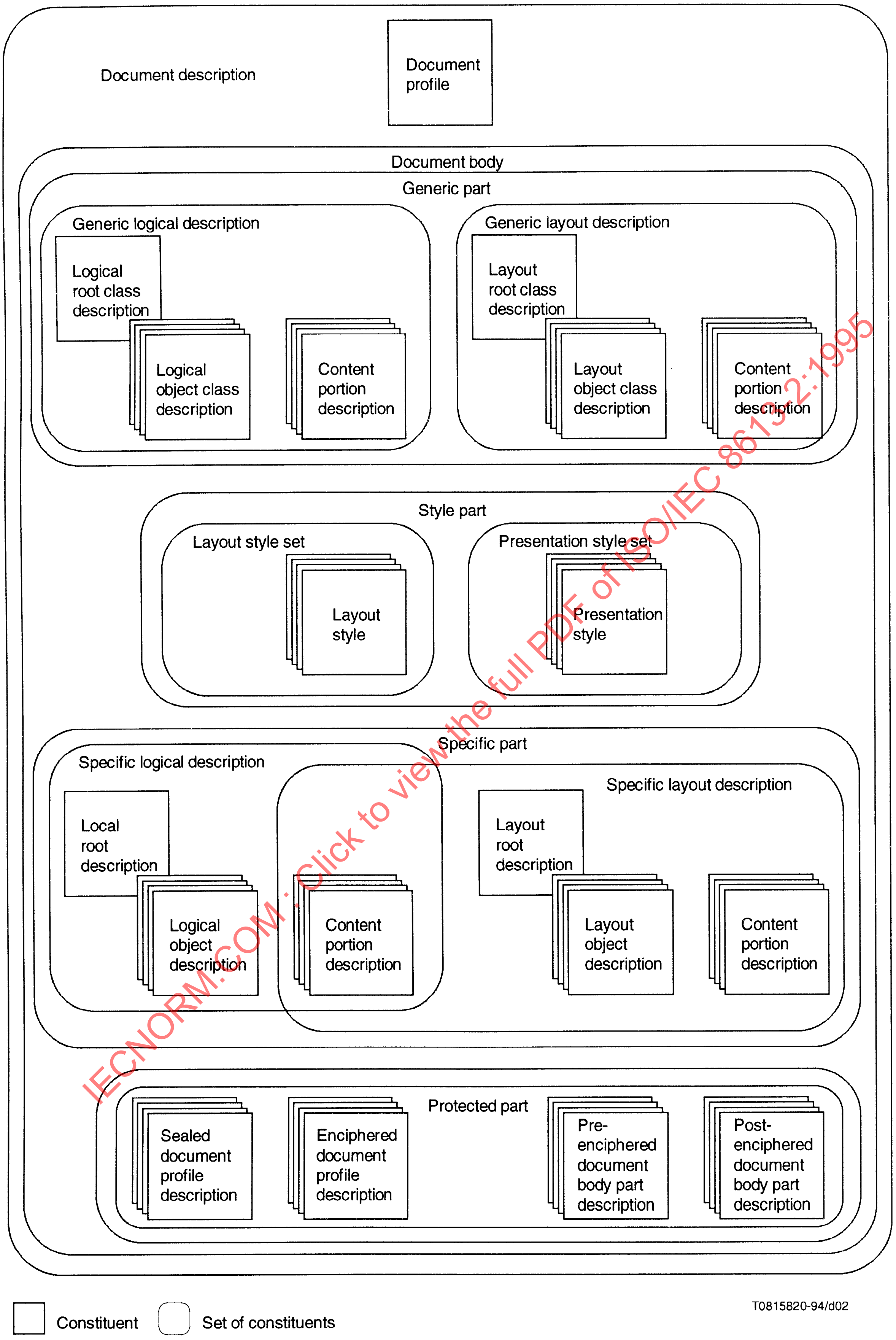

Document description
Document profile
Document body
Generic part
Generic logical description
Logical root class description
Logical object class description
Content portion description
Generic layout description
Layout root class description
Layout object class description
Content portion description
Style part
Layout style set
Layout style
Presentation style set
Presentation style
Specific part
Specific logical description
Local root description
Logical object description
Content portion description
Specific layout description
Layout root description
Layout object description
Content portion description
Protected part
Sealed document profile description
Enciphered document profile description
Pre-enciphered document body part description
Post-enciphered document body part description
Constituent
Set of constituents
T0815820-94/d02

Figure 2 – Descriptive representation of a document

6.4 Document processing model

This subclause describes a conceptual model for document processing.

It addresses only those aspects of document processing that are relevant to the document architecture defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613. As such, it is not a complete model of document processing since it does not specify all processing steps from document creation to document imaging.

It describes the principal operations performed on a document as a basis for understanding the semantics of the attributes defined in clause 9.

It 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 d

ocument processing model (see Figure 3) is summarized in this sub-clause. Three processes are shown:

- the editing process;
- the layout process;
- the imaging process.

The order of processes in the diagram is not intended to imply that they are performed sequentially in an actual implementation.

The document processing model provides for manual intervention only for carrying out editing processes on the specific logical structure and content. Manual intervention can occur, however, at many stages in the model, but it is neither explicitly allowed nor prohibited by the ITU-T Rec. T.410-Series | ISO/IEC 8613.

For example, in an actual implementation it may be possible to create and amend the logical and layout object class descriptions, and layout and presentation styles, but these processes are not included in this model since the ITU-T Rec. T.410-Series | ISO/IEC 8613 does not place any constraints on such editing.

6.4.1 The editing process

The editing process includes both document creation and document revision as these activities are indistinguishable from an architectural perspective.

This *editing process* includes the content editing process and the logical structure editing process. The *content editing process* is concerned with the creation of new content or the modification of previous content. The *logical structure editing process* is concerned with the creation of a specific logical structure or the modification of a previous specific logical structure and the allocation of content to basic logical objects. Modifications to the specific logical structure are required to conform to the rules specified in the generic logical structure, if present.

Included in the document creation and document revision activities are the creation of the generic logical structure, the generic layout structure, layout styles and presentation styles. Styles may be altered to represent changes made during the logical structure editing process. The generic layout structure may be edited in order to alter the intended layout of the document.

6.4.2 The layout process

The *layout process* includes the document layout process and the content layout process. These processes are concerned with the creation of a specific layout structure which can be used by the imaging process to present the document in human perceptible form on a presentation medium.

The *document layout process* creates a specific layout structure in accordance with the generic layout structure and information derived from the specific logical structure, the generic logical structure and layout styles (if present).

This process also determines the areas that are available within the created layout objects for the formatting of the document content (as described below) and is responsible for allocating the content to these *available areas.*

The *content layout process* is responsible for formatting (or laying out) the content portions into the available areas specified by the document layout process. This process makes use of information contained in the presentation attributes that apply to those content portions.

The document and content layout processes may interact, for instance if a content layout process cannot layout the content into the available area then it shall ask the document layout process for a larger available area and if provided with a larger available area shall repeat the content layout process.

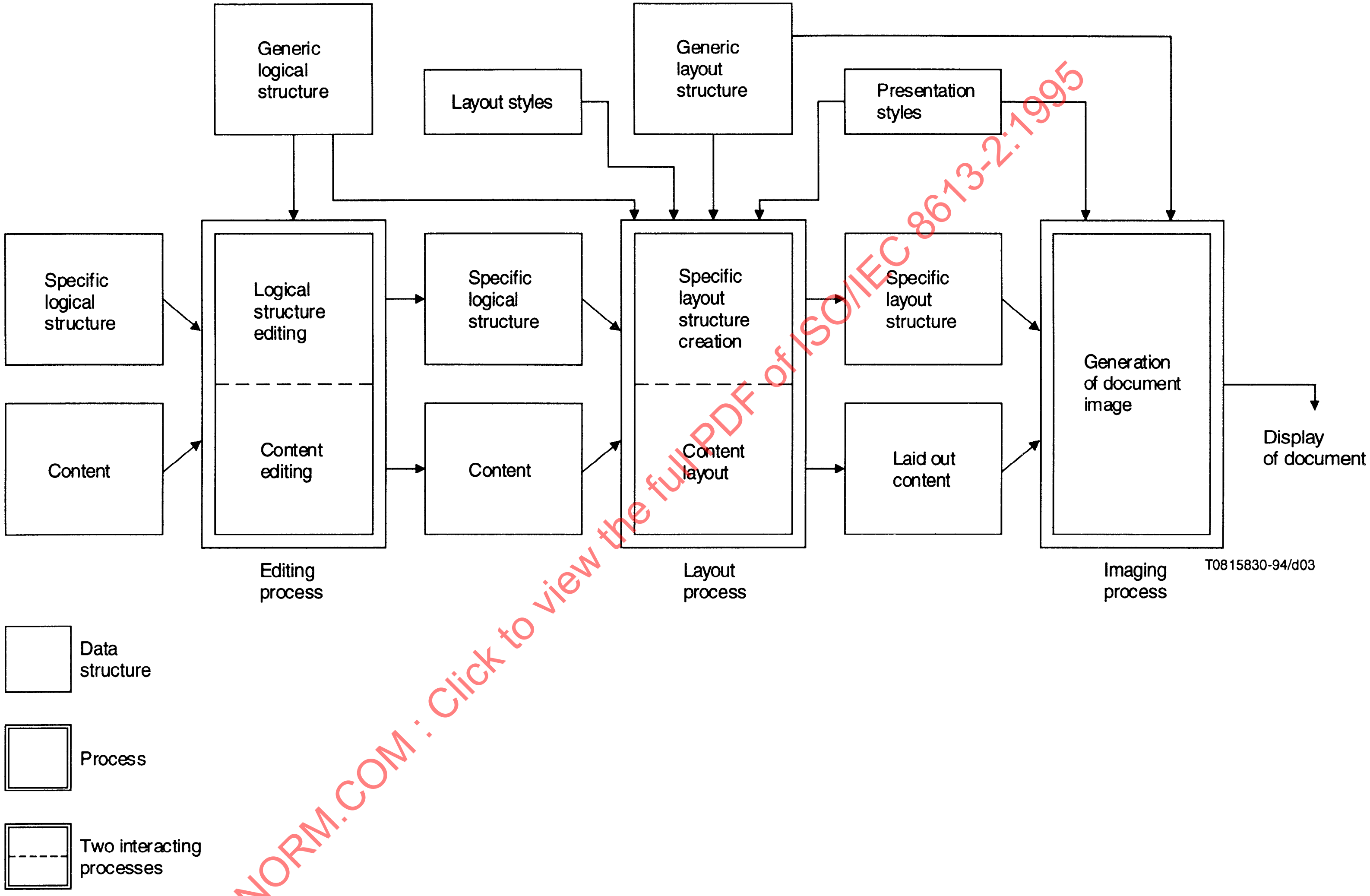

Generic logical structure
Layout styles
Generic layout structure
Presentation styles
Specific logical structure
Logical structure editing
Content editing
Specific logical structure
Specific layout structure creation
Content layout
Specific layout structure
Generation of document image
Content
Content
Laid out content
Display of document
Editing process
Layout process
Imaging process
T0815830-94/d03
Data structure
Process
Two interacting processes

Figure 3 – Document processing model

During the layout process, presentation attributes may be derived from presentation styles referenced from the specific logical structure, from the generic layout structure, and from the generic logical structure. Derivation of presentation attributes from the logical structures has precedence over derivation from the generic layout structure. That is, those presentation attributes derived from the logical structures that differ from those derived from the generic layout structure shall be specified explicitly in the specific layout structure by the layout process.

The document layout process is described in clause 10. The content layout process depends on the content architecture pertaining to the content being laid out and is described in those parts of the ITU-T Rec. T.410-Series | ISO/IEC 8613 that are concerned with particular content architectures.

The role of the constituents that represent the generic logical structure is different in the editing process and the layout process. In the editing process, the logical object class descriptions are used to construct the specific logical structure. In the layout process, a logical object class description is used as a source of attributes and content, if any, that is common to the logical objects of the class.

6.4.3 The imaging process

The *imaging process* consists of taking a specific layout structure and a corresponding generic layout structure (if present), with associated formatted content portions and information contained in presentation styles, and displaying it on a suitable presentation medium.

Some aspects particular to the imaging process are described in clause 11. However, the imaging process is a locally defined process. Hence, apart from defining the input information required by this process, which comprises the specific and generic layout structures, referenced presentation styles and the formatted content portions, this process is not formally defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613.

During the imaging process, presentation attributes are derived from the layout structures and referenced presentation styles. Any presentation attributes specified by the logical structures and referenced presentation styles are ignored.

The role of the constituents that represent the generic layout structure is different in the document layout process from that in the imaging process. In the document layout process, the layout object class descriptions are used to construct the specific layout structure. In the imaging process, a layout object class description is used as a source of attributes and content, if any, that is common to the layout objects of the class.

Some aspects of imaging depend on the content architecture pertaining to the content to be imaged and are described in those parts of the ITU-T Rec. T.410-Series | ISO/IEC 8613 that are concerned with particular content architectures.

6.5 Roles of the document architecture constituents in document processing

6.5.1 Editing process

This subclause describes the role of the various parts of the document architecture when the editing process is applied to documents of processable or formatted processable class.

6.5.1.1 Generic logical structure

The generic logical structure can be used to control the logical structure editing process that can be carried out on a document. This is accomplished by providing references from logical object descriptions to logical object class descriptions.

If a complete generator set of logical object class descriptions is present in the document then there is a reference from every logical object description to a logical object class description.

A complete generator set of logical object class descriptions controls the specific logical structure which is produced during the document creation and editing process. This is achieved by ensuring that logical objects are only created, deleted or modified according to corresponding logical object classes in the generic logical structure. In addition, the creation of a new logical object is facilitated because a logical object class description serves as a template for the logical object description.

A partial generator set of logical object class descriptions is similar to a complete generator set in facilitating the editing operations; it differs in that it guides rather than constrains the editing operations and may only apply to portions of the specific logical structure rather than all of it.

If a factor set of logical object class descriptions is present in the document, then the object classes in that structure can be used to facilitate the creation of objects in the specific logical structure that have common properties. This is achieved by providing references from a logical object description to a logical object class description. This provides a means of reducing the amount of common information in objects of the specific logical structure; this is referred to as "factorization" of information. A factor set of logical object class descriptions has no control over the structure of the specific logical structure.

6.5.1.2 Specific logical structure

The logical structure editing process involves making changes in the specific logical structure. These changes may be as follows:

- creating or deleting a logical object;
- changing the position of an object in the specific logical structure;
- changing the characteristics of an object by adding, deleting or modifying attributes specified by the object description.

Changing the characteristics of an object can also involve:

- changing the layout styles referenced by the object description;
- changing, adding or deleting a default value for an attribute in a default value list for a logical object description.

6.5.1.3 Content

The content editing process involves making changes to the content. Representing this can involve making changes to the presentation styles applicable to the basic logical object descriptions.

In order to edit the content information it must be in processable or formatted processable form. The content is changed by

a) adding, modifying or deleting one or more content elements;

b) adding, modifying or deleting embedded control functions.

Editing of formatted form content is outside the scope of the ITU-T Rec. T.410-Series | ISO/IEC 8613. However, all three forms of content may be specified by the content portions of the logical structure.

The editing algorithms used to change the content are not described in this model as they are outside the scope of the ITU-T Rec. T.410-Series | ISO/IEC 8613.

6.5.1.4 Generic layout structure

The generic layout structure, if present in the document, plays no direct role in the editing process.

However, the generic layout structure may be edited in order to alter the intended layout of the document. This process is of local concern only and is outside the scope of the ITU-T Rec. T.410-Series | ISO/IEC 8613.

6.5.1.5 Specific layout structure

The specific layout structure, if present in the document, plays no direct role in the editing process.

However, the specific layout structure which will be produced by the document layout process may be changed as a consequence of the editing process modifying the specific logical structure or the document content.

6.5.1.6 Layout styles

Layout styles are not used to represent controls on the editing process.

Layout styles may be altered to represent changes made during the logical structure editing process (see 6.5.1.2). This involves the addition, deletion or modification of the layout directive attributes contained in the layout styles. This can affect the layout characteristics of all the logical objects whose representation refers to the layout styles that are changed and can affect the relationships that the objects have with other objects in the document. Changing the layout styles may cause changes in the specific layout structure produced by the document layout process.

6.5.1.7 Presentation styles

Presentation styles are not used to represent controls on the editing process.

Presentation styles may be altered to represent changes made during the content editing process. This involves the addition, deletion and modification of attributes contained in the presentation styles. This can affect the layout and imaging of the content associated with all the basic logical objects whose representation refers to the presentation styles that are changed. For some presentation attributes, the same effects may be produced by editing the control functions that are embedded in the content but such editing will only alter the layout and imaging of the content in which the edited control functions are embedded.

6.5.1.8 Alternative descriptions

Alternative descriptions usually play no direct role in the editing process but may be derived automatically by the originating system from the primary descriptions. It is then the responsibility of the originating system to keep the alternative descriptions consistent with the primary description. Alternative descriptions may also have to be manually derived from the primary description (for example, descriptive text indicating the contents of the primary description); in this case it is the responsibility of the originator to maintain consistency during the editing process.

6.5.2 Layout process

This subclause describes the role of the various parts of the document architecture when the layout process is applied to documents of processable or formatted processable class. A model of this process is described in clause 10.

6.5.2.1 Generic logical structure

During the layout process, the generic logical structure may provide layout styles, presentation styles and generic content portions which may affect the creation of the specific layout structure.

Also, object classes may be referenced by the attribute "logical source" that is contained within layout object class descriptions. This results in the creation of additional layout objects that have no correspondence with any objects in the specific logical structure.

6.5.2.2 Specific logical structure

During the layout process, the sequential logical order of the objects in the specific logical structure determines the sequence in which the content of the document is considered by the layout process.

In addition, the logical objects can provide layout directive attributes which direct the document layout process and presentation attributes which direct the content layout process. These layout directive attributes and presentation attributes are specified by referencing a layout style or a presentation style, respectively.

6.5.2.3 Layout styles

During the layout process, layout styles provide layout directive attributes which direct the document layout process.

6.5.2.4 Presentation styles

During the layout process, presentation styles provide presentation attributes which direct the content layout process.

6.5.2.5 Content

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 specific structures. In addition, the content layout process may insert control functions into the content to facilitate the imaging process.

6.5.2.6 Generic layout structure

During the layout process, a complete generator set of layout object class descriptions must be available to determine a specific layout structure for the document.

The construction expressions specified by the layout object class descriptions determine all permissible specific layout structures which may be created by the layout process. Which of these permissible structures is used is determined from the specific logical structure, the generic logical structure, the content, and the layout and presentation styles.

6.5.2.7 Specific layout structure

The specific layout structure results from applying the document and content layout processes to the specific logical structure and the content, guided by the generic logical structure, the generic layout structure, layout styles and presentation styles.

6.5.2.8 Generic content

Generic content portions associated with logical object classes may be in formatted, processable or formatted processable form. When a logical object class specifying generic content portions is referred to from a basic logical object then when that basic logical object is encountered in the layout process, the generic content is laid out in accordance with the document and content layout processes and a new basic layout object is created and added, together with the created content portions, to the specific layout structure.

Generic content portions associated with layout object classes may be in formatted or formatted processable form. The layout of such content portions during the layout process does not cause additional content portions to be added to the specific layout structure. However, the content portions are indirectly associated with the specific layout structure by reference to the appropriate layout object class description.

6.5.2.9 Alternative descriptions

Alternative descriptions do not influence the reference layout process. If a system is to lay out a document that contains primary descriptions which it is not capable of processing, it may substitute alternative descriptions for those primary descriptions prior to the layout process.

A system that provides a layout process where both primary and alternative descriptions influence the layout process is outside the scope of this Specification.

6.5.3 Imaging process

This subclause describes the role of the various parts of the document architecture when the imaging process is applied to documents of formatted or formatted processable class.

6.5.3.1 Content

In the imaging process, the content, together with the specific layout structure, is used to produce an image of the document perceptible to a human.

After layout, the content may be either in formatted form or in formatted processable form. Both forms of content are suitable for imaging.

6.5.3.2 Generic layout structure

During the imaging process, the generic layout structure may provide, for the layout objects in the specific layout structure, any combination of the following:

a) attributes that direct the imaging of the content;

b) generic content portions.

6.5.3.3 Specific layout structure

In the imaging process, the specific layout structure, together with the content, is used to produce a human-perceptible image of the document.

The sequential order of the objects in the specific layout structure determines the precedence for imaging the content of the document by the imaging process. It is also possible to override the precedence of the layout objects by specifying an imaging order which is different from the sequential layout order.

6.5.3.4 Layout styles

In the imaging process, layout styles play no role.

6.5.3.5 Presentation styles

In the imaging process, the presentation attributes of the presentation styles referenced by the layout structures may affect the generated image of the content.

6.5.3.6 Alternative descriptions

Alternative descriptions do not influence the reference imaging process. If a system is to image a document that contains primary descriptions which it is not capable of processing, it may substitute alternative descriptions for these primary descriptions prior to the imaging process.

6.6 Security protection of parts of a document

This Specification distinguishes between the following two sets of security protected parts of a document:

a) Parts of a document profile description.

b) Parts of the document body consisting of complete object descriptions, object class descriptions, layout styles and presentation styles. A complete composite object description comprises the object descriptions for all subordinate objects and content portion descriptions for all associated content portions.

Two concepts of document security exist:

a) Information indicating how the whole document shall be handled as a single unit, according to the security policy of the security domain to which the originator belongs. The originator is responsible for making the indication; the actual security handling of the document is outside the scope of both originator and recipient, and outside the scope of the ITU-T Rec. T.410-Series | ISO/IEC 8613.

b) Information to be interchanged between the originator and the recipient on how security aspects of parts of the document shall be handled. The handling of this component of security is under the control of the originator and the recipient.

For parts of the document, concept b) covers the properties of:

- confidentiality;
- integrity;
- authenticity, including signature and non-repudiation of origin.

It is not the purpose of the ITU-T Rec. T.410-Series | ISO/IEC 8613 to specify any particular security scheme or method, but rather to provide the means in the document for a variety of possible security implementations as required by various security policies.

Two cryptographic techniques may be used to provide the security protections in the ITU-T Rec. T.410-Series | ISO/IEC 8613:

- encipherment of clear text to provide for confidentiality and possibly integrity of data;
- production of cryptographically derived information to provide for integrity and authenticity of data.

There are two phases, a generation phase and an interpretation phase, involved with security protection of parts of a document.

The generation phase enciphers or seals parts of the document and creates security information that is added to the document. The protected part descriptions are created in this phase. These descriptions contain the enciphered and the sealed versions of the protected parts of the document.

The interpretation phase deciphers enciphered protected parts or validates seals that have been created in the generation phase.

6.6.1 Intended and privileged recipients

Two categories of recipients are defined, namely intended recipients and privileged recipients.

An intended recipient of a document is a recipient that is expected by the originator to receive or have access to the document.

A privileged recipient of a document is a recipient that, in addition to being an intended recipient, has the right to perform certain security-related operations intended for that particular recipient, such as to interpret specified enciphered parts of the document and to perform integrity and authenticity checks on specified parts of the document.

6.6.2 Protecting parts of the document profile

Protected parts of a document profile are specified in two sets of document profile descriptions, the set of enciphered document profile descriptions and the set of sealed document profile descriptions.

The sealed document profile descriptions are for integrity, authenticity and non-repudiation of origin, one for each seal of parts of the document profile.

The enciphered document profile descriptions are for confidentiality, one for each encipherment of parts of the document profile.

All information about each sealed or enciphered document profile description is found in the document profile.

6.6.3 Protecting parts of the document body

The parts of the document body that may be protected are object class descriptions, object descriptions, layout styles and presentation styles.

Confidentiality is based on encipherment. Integrity, authenticity and non-repudiation of origin are based on a seal.

For encipherment:

- The protection of a composite object description implies that itself and all its subordinates and all the content portions directly referred to from any of its subordinates are protected.
- The protection of a basic object class description implies that the basic component itself and all content portions directly referred to from it are protected.
- The protection of a basic object description implies that for processable form and formatted form documents, itself and all the content portions directly referred to from it are protected. In a formatted processable form document all content portions directly referred to from a protected basic object description in one of the structures will be protected, but it may be that only some of the content portions referred to from a basic object description in the other structure are protected.
- The protection of a composite object class description, layout style or presentation style implies that all of it is protected.

For sealing:

- The protection of a composite object description implies that itself, all its subordinates and all the content portions directly referred to from any of its subordinates are protected.
- The protection of a basic object class description implies that the basic component itself and all content portions directly referred to from it are protected.
- The protection of a basic object description implies that for processable form and formatted form documents, itself and all the content portions directly referred to from it are protected. In a formatted processable form all content portions directly referred to from a protected basic object in one of the structures will be protected, but it may be that only some of the content portions referred to from a basic object description in the other structure are protected.
- The protection of a composite object class description, layout style or presentation style implies that all of it is protected.

For confidentiality enciphered document body parts are defined.

There are two sets of enciphered document body part descriptions: one set for pre-enciphered document body parts and one set for post-enciphered document body parts.

A pre-enciphered document body part description is provided for each encipherment performed before the layout process, and a post-enciphered document body part description is provided for each encipherment performed after the layout process.

All information about each pre- or post-enciphered document body part is found in the document profile.

For integrity, authenticity and non-repudiation of origin seals are provided. The seals and all information about them are found in the document profile.

7 Document structures and colour

7.1 Specific structures

7.1.1 General principles

The specific layout and specific logical structures of a document are hierarchical structures of objects.

The object at the highest level in the hierarchy of the specific layout structure is called the document layout root and the object at the highest level in the hierarchy of the specific logical structure is called the document logical root.

A *composite object* is an object that has one or more subordinate objects. The structural relationships of a composite object identify its immediate subordinates.

The minimum number of hierarchical levels below the highest level in either structure is one. Thus, the document root object is always a composite object. The actual number of levels is variable and depends upon a given document.

In general, at any level in a specific structure, the subordinates of any composite object can consist of any number and combination of basic objects and composite objects.

Basic objects are objects that have no subordinate objects.

Basic objects are also distinguished from composite objects since basic objects are the only objects with which the content of a document is *associated* (see 7.1.3). Every basic object has content, either in the form of one or more specific content portions, in the form of an expression for generating content or derived from an object class to which the basic object belongs.

Each object in a structure is of a certain object type. The object types that can occur within a specific logical structure or a specific layout structure are defined in 7.2 and 7.3, respectively. The *object type* determines which attributes are applicable to an object description.

The particular attributes and attribute values comprising the object description characterise the object, that is, the attributes specify the characteristics of the object itself and specify the relationships that it has with other components in the document.

Each object in a structure is uniquely identified within that structure. The structural relationships between a composite object and its immediate subordinates are defined in the composite object description.

7.1.2 Sequential order

In the case that more than one immediate subordinate is identified by a composite object then the composite object description specifies an ordering of these immediately subordinate objects. This subordinate ordering is used to define a unique sequential ordering of all the objects in a structure.

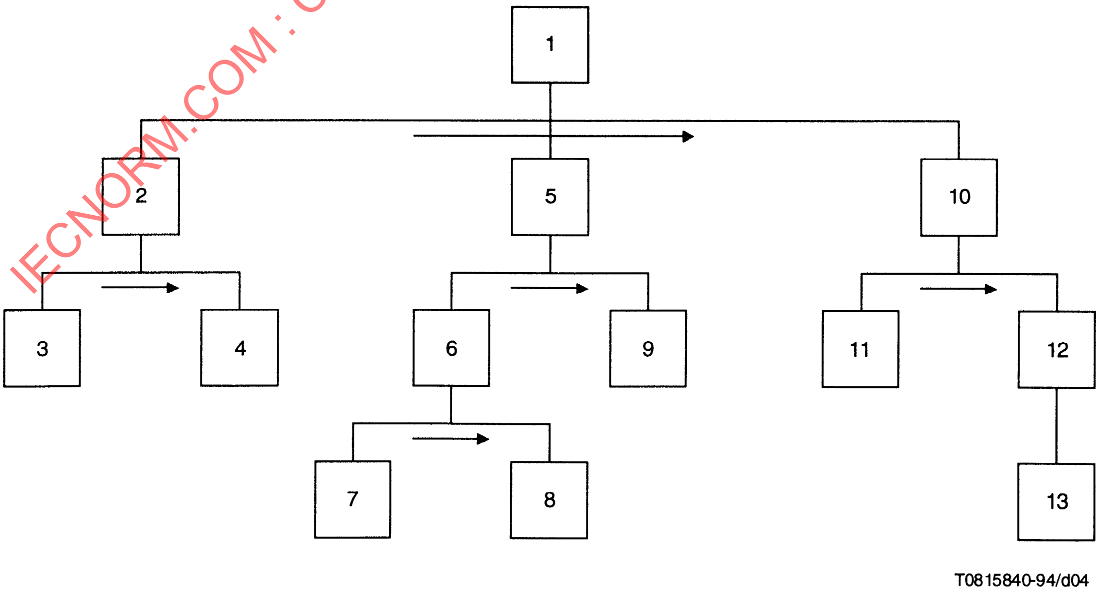
This sequential order is such that each object in the structure is succeeded by all of its immediate subordinates, before any other objects with the same immediate superior. Each of the immediate subordinates is followed by all of its immediate subordinates, before proceeding to the next immediate subordinate in sequence. The immediate subordinates occur in the subordinate ordering defined within the object description.

The sequential order of the specific logical structure is called the sequential logical order; the sequential order of the specific layout structure is called the sequential layout order.

The *sequential logical order* defines the order in which the logical objects are intended to be processed by the layout process (see clause 10).

The *sequential layout order* defines the order of precedence for imaging unless this is overridden by the specification of an imaging order in the object description (see 11.1).

An example of the sequential order of a structure is shown in Figure 4, where the numbers indicate the sequential order.



Note – Arrows define sequential order.

Figure 4 – Example of sequential order

7.1.3 Content portions

The content of a document is divided into content portions to allow the document architecture to address each content portion as a unit. The amount of content to be placed into a content portion may vary from no content elements to the entire document content. The maximum number of consecutive content elements that have the same properties and characteristics will typically be placed together in one content portion. For example, a content portion may represent a heading, paragraph, picture or perhaps just that amount of content that needs to have particular constraints regarding its imaging or processing.

Each content portion is structured according to a single content architecture. When a document contains only a single specific structure, either a specific layout structure or a specific logical structure, each content portion in the specific structure is associated with a single basic object.

Each basic object may have more than one associated content portion, provided that all of these content portions pertain to the same content architecture. In the case that more than one content portion is associated with a basic object then the object description specifies the ordering among these content portions.

7.2 Specific logical structures

The specific logical structure provides a means of modelling the structure of a document in terms of logical objects which have meanings that are significant to the application or user.

For example, the specific logical structure can be used to model a document in terms of chapters, sections, paragraphs and footnotes. The ITU-T Rec. T.410-Series | ISO/IEC 8613 does not standardise such particular kinds of logical objects. However, it does provide the means by which any such construction can be defined in terms of logical objects to be interpreted as chapters, sections, paragraphs or footnotes, etc.

The significance of this form of definition is that instead of standardising a few particular kinds of logical objects it allows any number of differing kinds of logical objects to be constructed, so as to meet a wide variety of needs.

NOTE 1 – Particular kinds of logical objects are standardised by the various Recommendations | International Standards for document application profiles. Each document application profile defines particular kinds of logical objects, to meet the needs of that profile. (Document application profiles are defined further in ITU-T Rec. T. 411 | ISO/IEC 8613-1.)

Logical objects provide the means to specify the specific logical structure of a document that is relevant to a particular application of the ITU-T Rec. T.410-Series | ISO/IEC 8613. For example, the specific logical structure may be required in the processing of a document, such as editing and laying out the document.

Clause 9 defines the attributes that are used to describe the characteristics of logical objects, including the use of presentation styles and layout styles.

The objects that can occur within a specific logical structure of a document are of the following object types:

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

The allowable hierarchical relationships between logical objects are shown in Figure 5.

NOTE 2 – The notation used in Figure 5 is that defined in Annex A.

7.2.1 Document logical root

The *document logical root* is the highest level object in the hierarchy of the specific logical structure. It is a composite object whose immediate subordinates consist of any number and combination of basic logical objects and composite logical objects.

7.2.2 Composite logical objects

A *composite logical object* is a composite object that is immediately subordinate either to the document logical root or to a composite logical object of a hierarchically higher level. Its immediate subordinates consist of any number and combination of composite logical objects and basic logical objects. Thus, the number of hierarchical levels between a basic logical object and the document logical root may vary from one basic logical object to another within any particular document.

The use of composite objects is optional. Content portions cannot be directly associated with composite logical objects.

The hierarchical relationships between a composite logical object and its subordinates can express a logical relationship between that composite logical object and its subordinates that has a significance relevant to a particular application of the ITU-T Rec. T.410-Series | ISO/IEC 8613. For example, these relationships may be used to specify which sections, paragraphs and diagrams relate to a certain chapter in a document.

7.2.3 Basic logical objects

A *basic logical object* is a basic object that is immediately subordinate either to the document logical root or to a composite logical object.

A basic logical object is a container for portions of the document content. Zero, one or more content portions are associated with a basic logical object. In the case that zero content portions are associated, the content is either specified in the form of an expression for generating content or is derived from a basic object class description.

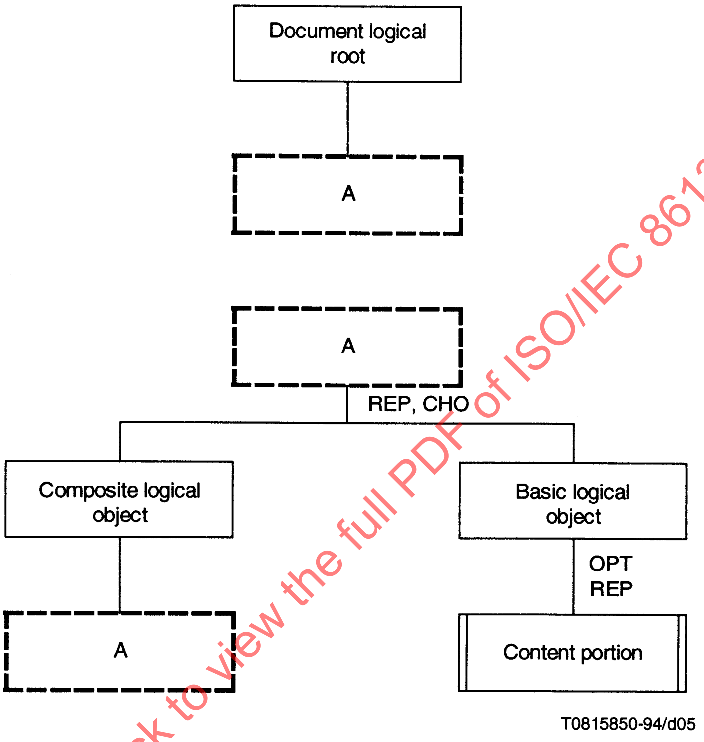


Figure 5 – Permissible specific logical structures

7.3 Specific layout structure

The specific layout structure provides a means of modelling the structure of a document in terms of layout objects which have meanings that are significant for the layout process and the imaging process.

Clause 9 defines the attributes that are used to describe the characteristics of layout objects, including the use of presentation styles.

7.3.1 Objects of the layout structure

The objects that can occur within the specific layout structure of a document are of the following object types:

- document layout root;
- page set;
- composite or basic page;
- frame;
- block.

The allowable hierarchical relationships between layout objects are shown in Figure 6.

NOTE – The notation used in Figure 6 is that defined in Annex A.

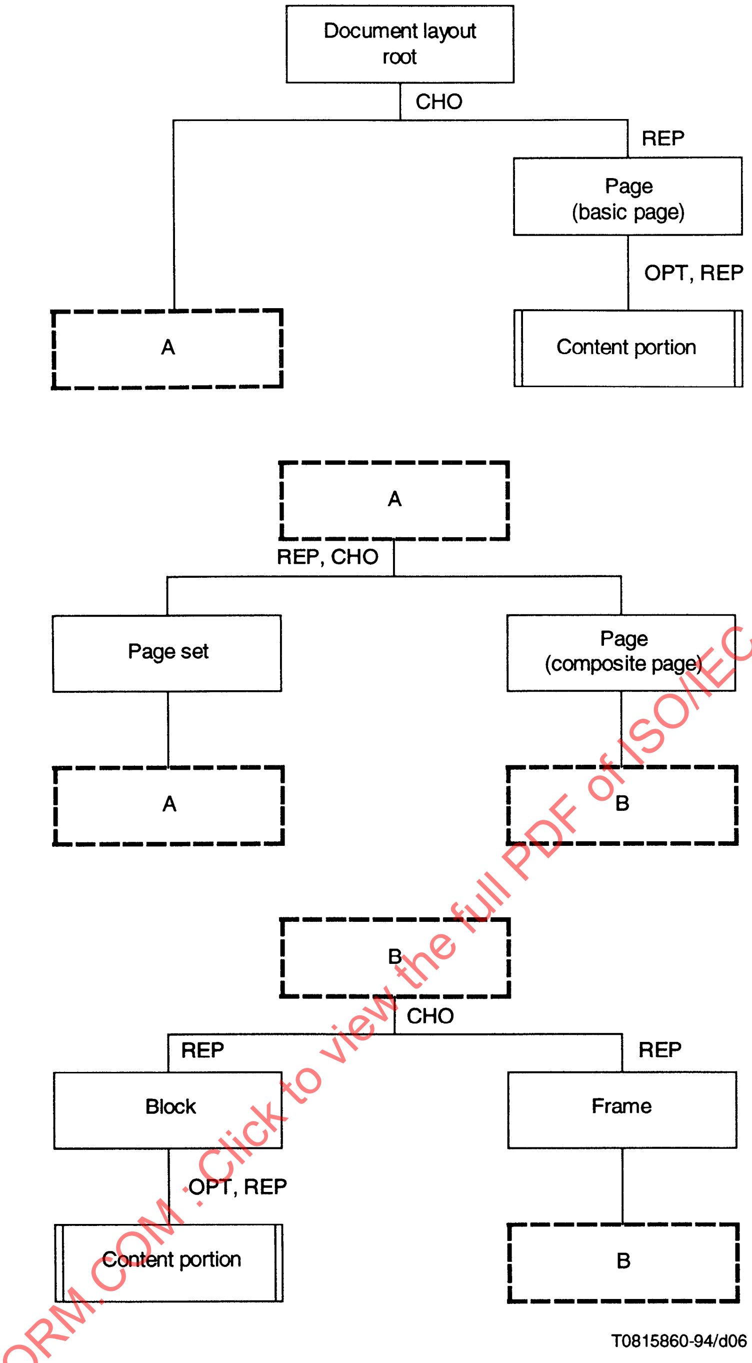


Figure 6 – Permissible specific layout structures

7.3.1.1 Document layout root

The *document layout root* is the highest level object in the hierarchy of the specific layout structure. It is a composite object whose immediate subordinates consist of any number and combination of page sets and pages.

7.3.1.2 Page set

A page set can be used to identify a number of page sets or pages (or a combination of both) as a group, for example, the pages which contain a particular section of a document.

A *page set* is a composite object that is immediately subordinate to either the document layout root or to a page set of a hierarchically higher level. Its immediate subordinates consist of any number and combination of page sets and pages. Thus the number of hierarchical levels between a page and the document layout root may vary from page to page within any particular document.

7.3.1.3 Page

A *page* is a rectangular area used as the reference area for positioning and imaging the content of the document. Its size may be smaller than, equal to or greater than the size of the presentation medium.

A page is immediately subordinate either to the document layout root or to a page set.

A page can be a basic object or a composite object and is correspondingly termed either a *basic page* or a *composite page*.

The pages within a layout structure can be either basic pages or composite pages, but not a combination of these.

A basic page has zero, one or more content portions associated with the page. In the case that zero content portions are associated, the content is derived from a basic object class description.

A composite page has as immediate subordinates either any number of frames or any number of blocks.

Basic pages and composite pages whose immediate subordinates are blocks shall only be included within documents of the formatted document architecture class.

7.3.1.4 Frame

A *frame* is a rectangular area that is contained entirely within the area of the object to which it is immediately subordinate. It is positioned so that its sides are parallel to the sides of the enclosing page. Frames are used to define areas for laying out the content associated with composite pages.

A frame is a composite object that is immediately subordinate to a composite page or to an enclosing frame. The immediate subordinates of a frame consist of either any number of frames or any number of blocks. Thus the number of hierarchical levels between a block and the enclosing page may vary from block to block within any particular page.

A *lowest level frame* on any branch of the layout structure is a frame which does not have any subordinate frames. Only lowest level frames can contain blocks.

7.3.1.5 Block

A *block* is a rectangular area that is contained entirely within the area of the object to which it is immediately subordinate. It is positioned such that its sides are parallel to the sides of the enclosing page. A block is an area for the positioning and imaging of portions of the document content.

A block is a basic object that is immediately subordinate to a page or a frame. Zero, one or more content portions are associated with a block. In the case that zero content portions are associated, the content is derived from a basic object class description.

7.3.2 Positioning layout objects

This subclause defines the rules for positioning and dimensioning frames and blocks within pages. These layout objects are used to layout and image the content of documents.

7.3.2.1 Page co-ordinate system

The positions of all layout objects subordinate to pages are specified, directly or indirectly, by means of an orthogonal *page co-ordinate system*. The origin of the page co-ordinate system is the top left corner of the page. The horizontal axis coincides with the top edge and the vertical axis coincides with the left edge of the page. The horizontal and vertical axes determine the *horizontal direction* and *vertical direction*, respectively, of the subordinate layout objects, as shown in Figure 7. This figure also defines names for the corners of a layout object: *top left corner, top right corner, bottom left corner* and *bottom right corner*. Horizontal positions are measured positively from the vertical axis to the right and vertical positions are measured positively from the horizontal axis downward.

The origin of the page co-ordinate system is the reference point used for positioning.

7.3.2.2 Positioning of frames

The reference point for positioning a frame is the top left corner of that frame. Frames are positioned relative to the reference point of the object to which they are immediately subordinate. Thus, frames that are immediately subordinate to a page are positioned relative to the origin of the page coordinate system. Frames which are immediately subordinate to another frame are positioned relative to the reference point of that frame.

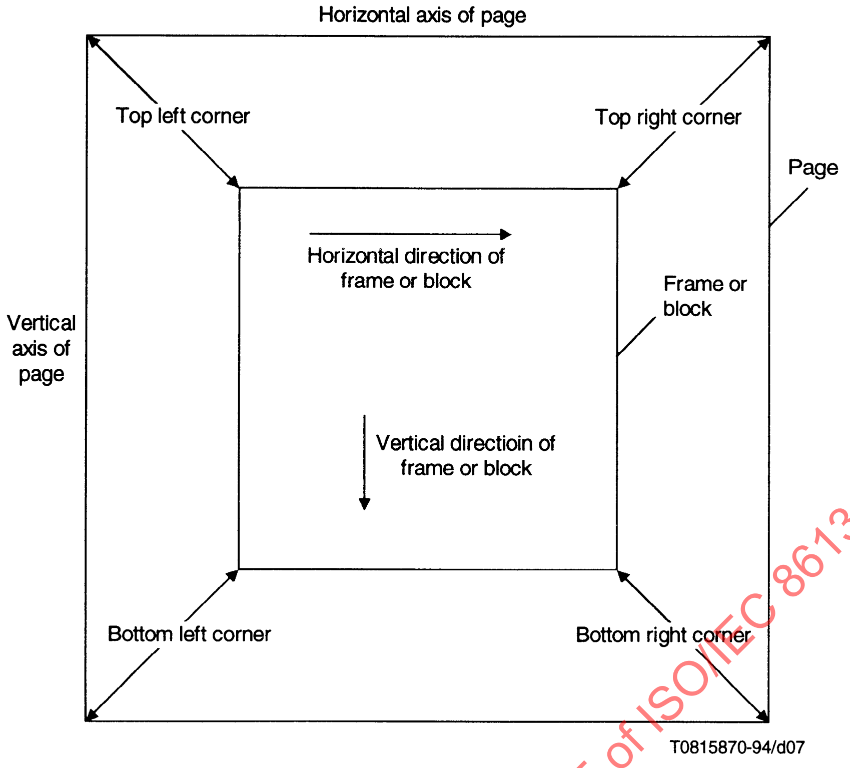


Figure 7 – The horizontal and vertical directions of layout objects

7.3.2.3 Positioning of blocks

The reference point for positioning a block is the top left corner of that block. Blocks are positioned relative to the reference point of the layout object to which they are immediately subordinate.

Thus, blocks that are immediately subordinate to a page are positioned relative to the origin of the page coordinate system. Blocks that are immediately subordinate to a frame are positioned relative to the reference point of that frame.

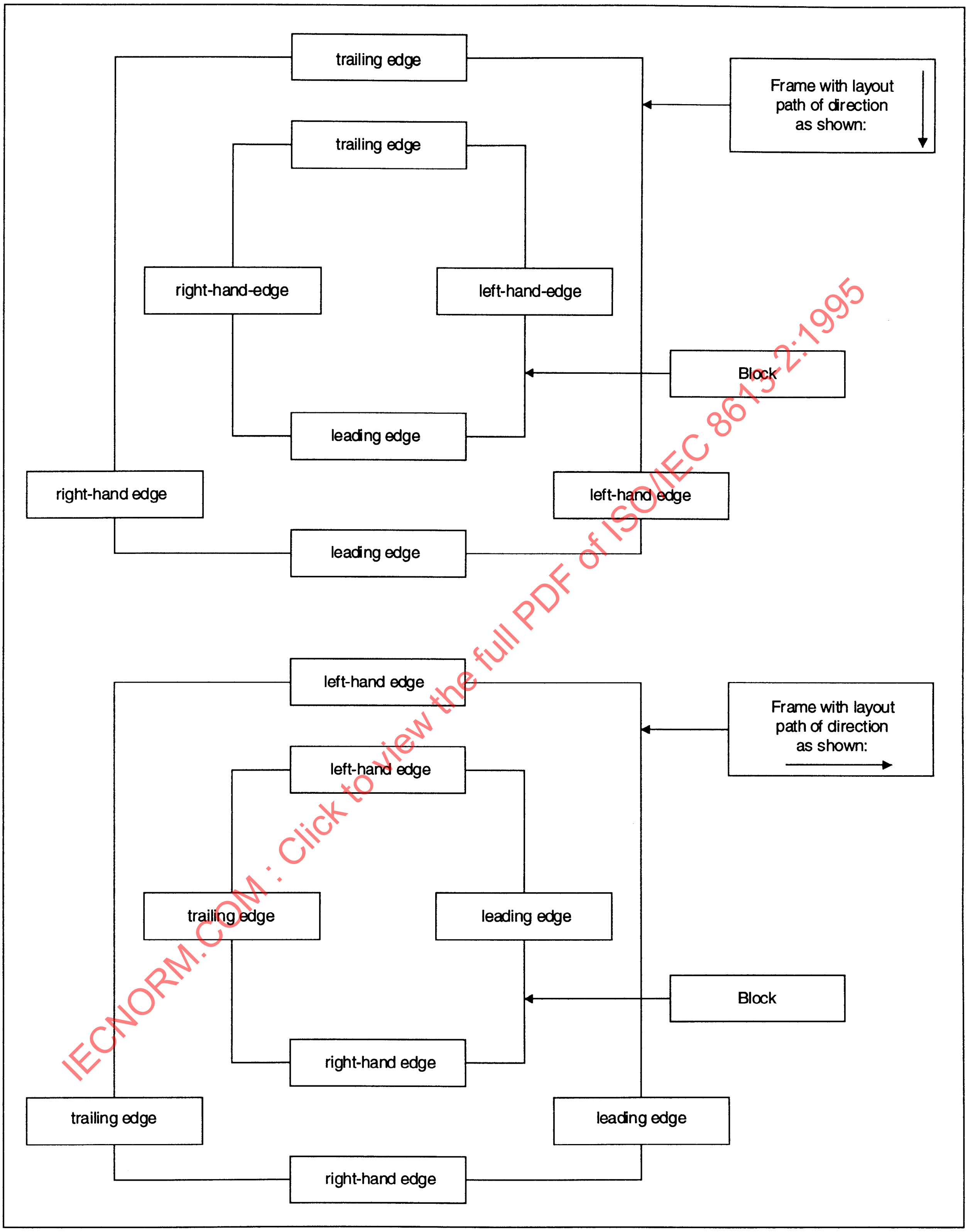
7.3.3 Naming of edges of frames and blocks

In the following definition, the layout path (see 9.4.2.2, 10.5 and 10.6) referred to is that specified for a frame, or, in the case of a block, for its immediately superior frame.

For a frame or block, the leading and trailing edges are defined as the two opposite edges of the same frame or block that are orthogonal to the direction of layout path, such that the direction from the trailing edge to the leading edge is in the same direction as the layout path.

The left-hand and right-hand edges of a frame or block are defined as the two opposite edges of the same frame or block that are parallel to the direction of layout path, such that the direction from the right-hand edge to the left-hand edge is at an angle of 90° anti-clockwise relative to the direction of the layout path.

The names of the edges of frames and blocks are illustrated in Figure 8.



T0815880-94/d08

Figure 8 – Naming of the edges of blocks and of their immediately superior frames

7.3.4 Measurement

7.3.4.1 Basic measurement unit

For the purpose of conveying the originator's intentions, all dimensions and positions are expressed in basic measurement units. The value of the basic measurement unit (BMU) is equal to 1/1200 of 25.4 mm. A locally defined scaling factor may be used to map the document to a particular imaging device.

7.3.4.2 Scaled measurement unit

All attributes and numeric control function parameters that specify absolute or relative positions and dimensions are expressed as integral multiples of a scaled measurement unit (SMU) that is equal to m/n basic measurement units. The *unit scaling factor* is specified in the form of two integers m and n by the document profile attribute “unit scaling” (see ITU-T Rec. T.414 I ISO/IEC 8613-4). When the attribute “unit scaling” is not specified, the scaled measurement unit used in the positioning and dimensioning attributes is equal to the basic measurement unit.

7.3.5 Borders of frames and blocks

Borders can be specified to be around the edges of blocks or within the edges of frames.

A *border* consists of the border freespace and the border line (see Figures 9 and 10).

The border line is described by specifying its width and line type. Line type can take the values 'solid', 'dashed', 'dot', 'dash-dot', 'dash-dot-dot' or 'invisible'.

The border freespace is described by specifying its width.

The border may be specified for a particular set of edges of the frame or block, or for all edges. The characteristics of the border for each of the edges may have a separate specification.

7.3.5.1 Borders of blocks

A border around a block is illustrated by Figure 9.

In the case of a block, the border freespace provides a region which surrounds the block between the block boundary and the border line.

The border surrounds the block and does not intrude on the block. The border is contained entirely within the area of the object to which the block is immediately subordinate. This means that the document layout process shall take the border into account in determination of the available area.

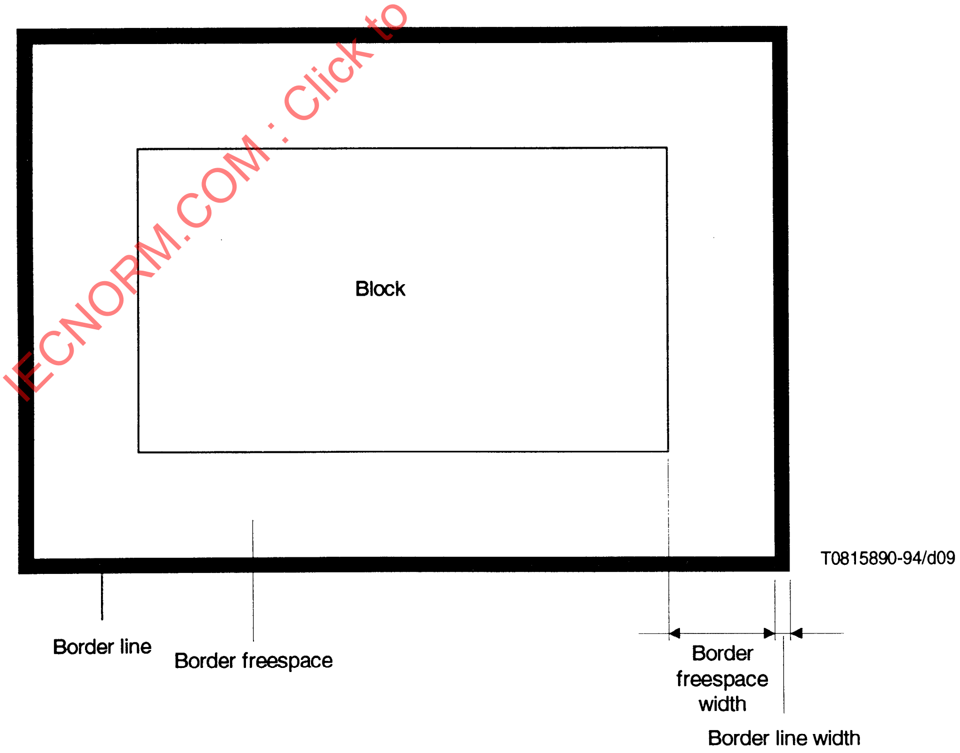


Figure 9 – Border around a block

7.3.5.2 Borders of frames

A border within a frame is illustrated by Figure 10.

In the case of a frame, the border freespace provides a region which is within the frame between the frame boundary and the border line.

The border reduces the available area within the frame for layout purposes. This means that the document layout process shall take the border into account in determination of the available area.

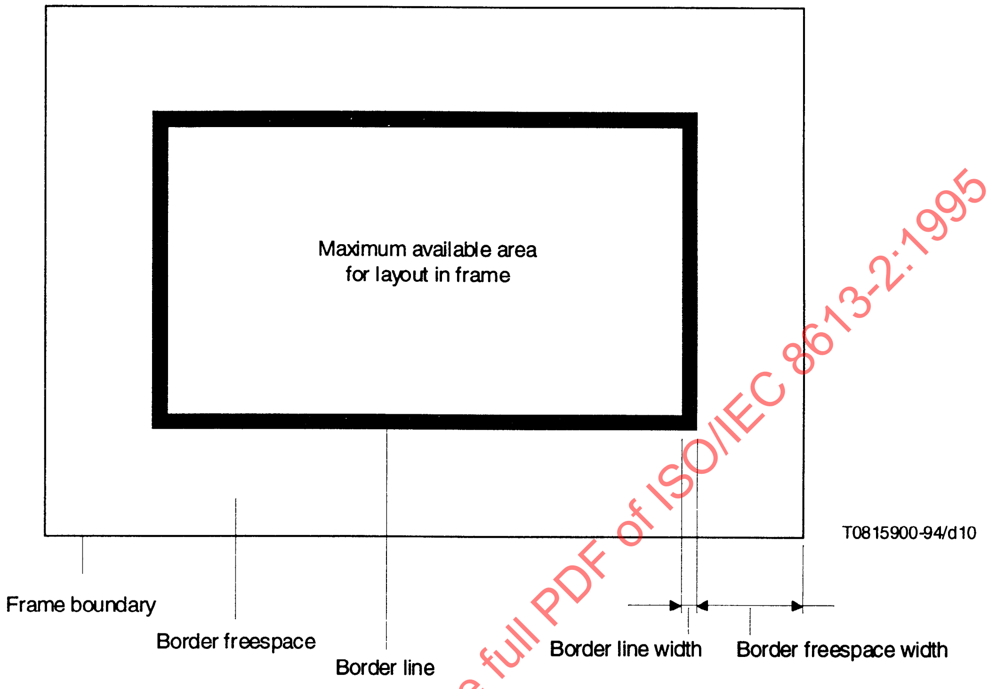


Figure 10 – Border within a frame

7.4 Documents containing both specific structures

When a document contains both a specific layout structure and a specific logical structure, each content portion in the specific structure is in general associated with both of the structures. However, some of the content portions can be associated with the specific layout structure only. These are the content portions that:

- represent formatted content corresponding to generic content portions associated with a basic logical object class of the generic logical structure or resource-document;
- are created as a result of the attribute "content generator";
- are created as a result of the attribute "logical source" of a layout object class description.

The allocation of content portions to basic objects in the specific layout structure need not correspond to the allocation of content portions to basic objects in the specific logical structure.

For example, a basic logical object representing a paragraph would need to contain at least two content portions in the case where the paragraph was split across a page boundary. This is shown in Figure 11 if the middle basic logical object represents the split paragraph and the composite layout objects represent the two separate pages.

This is in contrast to the situation when a document contains only one specific structure, in which case there is no need for dividing the content associated with a basic component into more than one content portion.

Figure 11 also illustrates the correspondence between logical and layout objects in the case that there is both a specific logical structure and a specific layout structure.

The presentation attributes for a content portion are specified by the specific layout structure and referenced presentation styles, any specified by presentation styles referenced by the specific logical structure are ignored.

In the imaging process, the logical structure and referenced styles are ignored.

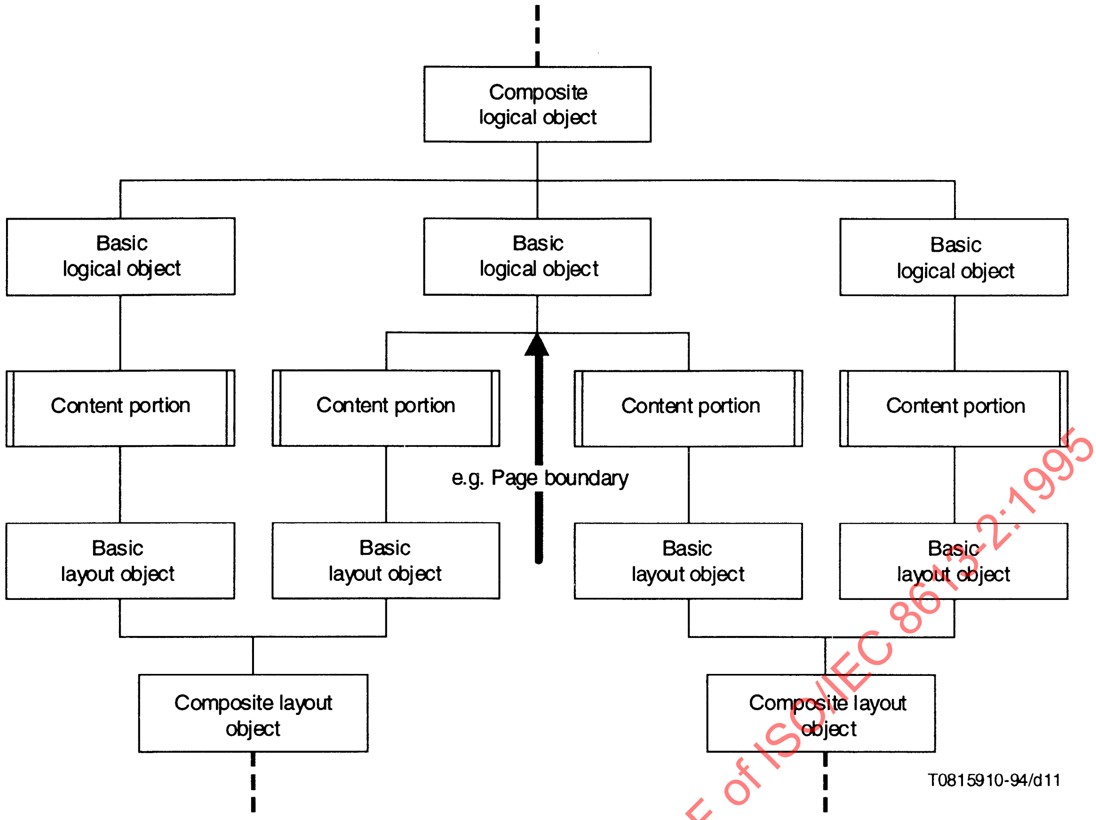


Figure 11 – Example of relationship between logical and layout objects and associated content portions

7.5 Generic structures

7.5.1 General principles

The generic structure of a document describes characteristics common to a number of objects within a document. In the most comprehensive case it describes the common characteristics of a document class.

The generic structure can be used to

- improve transmission efficiency by factorisation;
- maintain the internal consistency of a document by providing the recipient with the structural information necessary to edit and/or lay out the document as intended by the originator;
- facilitate the creation of objects and documents by the recipient as prepared by the originator.

The generic structure consists of a generic logical structure and/or a generic layout structure.

Within the set of constituents representing the generic structure each object class description consists of attributes which parallel the attributes of object descriptions. These attributes can be used to determine the value(s) for the attributes of the corresponding object descriptions. The object class descriptions may also contain references to layout styles and presentation styles.

If an attribute value is specified explicitly in an object description then this overrides any value that may be derived from the corresponding attribute of an object class description.

Each object class in a generic structure specifies the object type of all the objects that are derived from the object class. There may be many object classes of the same object type. The object types that can occur are specified in 7.5.6 and 7.5.7.

Each object class in a generic structure is uniquely identified within that structure.

Object classes for basic objects may specify content, either in the form of generic content portions or by specifying an expression to generate content.

7.5.2 Generator and factor sets of object class descriptions

A complete generator set of logical object class descriptions can be used to control the editing process. In a document with such a complete generator set every logical object description references a logical object class description.

In addition, the generation of immediate subordinates is completely derived from the referenced logical object class descriptions.

A partial generator set of logical object class descriptions can be used to guide the generation of sub-hierarchies of the specific logical structure during the editing process. In a partial generator set, some composite logical object class descriptions may guide the possible subordinates for objects of the class, while others do not.

A factor set of logical object class descriptions can be used to guide the creation of logical objects during the editing process. In a factor set, the composite logical object class descriptions do not constrain the possible subordinates for objects of the class. Such a factor set need not include an object class description corresponding to the document logical root.

A complete generator set of layout object class descriptions can be used to control the layout process. In a document with such a complete generator set every reference from the layout directive attributes, either by layout reference or layout category, is satisfied in the layout object class descriptions.

If the constituents representing the specific layout structure are present together with such a complete generator set then every composite layout object description references a layout object class description, and the generation of immediate subordinates is completely derived from the referenced layout object class descriptions. A block may, but need not, reference a layout object class description.

A partial generator set of layout object class descriptions can be used to guide the generation of sub-hierarchies of the specific layout structure. In a partial generator set, some composite layout object class descriptions may guide the possible subordinates for objects of the class, while others do not.

A factor set of layout object class descriptions can be used to guide the creation of layout objects. In a factor set, the composite layout object class descriptions do not constrain the possible subordinates for objects of the class. Such a factor set need not include an object class description corresponding to the document layout root.

In any of the preceding cases, object class descriptions achieve factorisation by holding information that is common to a number of object descriptions. Such factorisation of attribute values, by avoiding replication, can result in improved transmission efficiency and can also facilitate the creation of objects by the recipient. Such factorisation applies to content when an object class description either specifies generic content portion description(s) or specifies an expression to generate content.

7.5.3 Complete generator sets of object class descriptions

A complete generator set of layout object class descriptions form the nodes of a single directed graph.

A complete generator set of logical object class descriptions form the nodes of a set of directed graphs. This set consists of a primary graph and, optionally, one or more secondary graphs, each corresponding to the logical object class descriptions referenced by the attribute "logical source" (see 9.4.2.5).

Each directed arc of each graph connects the node from which it starts with each of the nodes which represent a possible class of immediately subordinate objects that can be generated. For any node each possible class of immediately subordinate objects is represented by a directed arc starting from the node and ending on a node representing the class of the immediately subordinate objects.

The rules describing the possible hierarchical relationships between a composite object and its immediate subordinates, which correspond to these arcs, are defined in the object class description of the composite object.

Each graph is connected such that there is a single root node which is such that:

- it is possible to reach every other node on a path through the graph following the directed arcs;
- there is no path through the graph following the directed arcs from which it is possible to reach this single root node.

The single root node of the graph represents the object class description of the object type document layout root in the set of constituents representing the generic layout structure.

The single root node of the primary graph represents the object class description of the object type document logical root in the set of constituents representing the generic logical structure.

The single root node of each secondary graph represents an object class description which is identified by the attribute "logical source" of one or more layout object class descriptions.

Some paths may pass through a particular node a multiple number of times. Some of the directed arcs may start and end on the same node.

The non-terminal nodes, that is the nodes from which some directed arc starts, which in all cases include the document root itself, represent object classes for composite objects.

In the case of the terminal nodes, that is the nodes from which no directed arc starts, the situation depends on which generic structure is represented by the complete generator set of object class descriptions.

If the generic logical structure is represented, then the terminal nodes represent the object classes for basic logical objects.

If the generic layout structure is represented, then the terminal nodes represent object classes for basic pages, for blocks, or for frames which will always be the lowest level frame in the particular branch of the hierarchy (see 7.5.7).

Generic content portions are referenced only from object classes for basic objects and each generic content portion shall be referenced by just one object class.

7.5.4 Generation of structures

A complete generator set of object class descriptions and associated generic content portions representing a generic structure is used to maintain the internal consistency of a document. This is achieved by defining which specific structures are possible in a given document class and which object classes are possible in parts of a document (see 9.3.2.1). When creating, editing or laying out a document, a generic structure can be used to control the generation and modification of the specific structure and thereby preserve the intentions of the originator, in this case the creator of the document class description.

Generation and modification of the specific structure is controlled by ensuring that the attributes of each object class description are used to specify default values for attributes in the corresponding object descriptions.

Recursion is permitted in the generic logical structure and in the generic layout structure, thus objects of the same object class may be hierarchically related.

The ITU-T Rec. T.410-Series | ISO/IEC 8613 does not define any particular generic structures or object classes. However, it allows the definition of a potentially infinite variety of document classes.

7.5.5 Content in generic structures

An object class description for a basic object can specify content in one of three ways:

- by specifying one or more generic content portions;
- by specifying a content generator;
- by referencing an object class in a resource-document which has content specified.

A content generator allows the content information to be specified by an expression which is evaluated during the layout process. Together with specification of a content generator, a content portion description may also be present, in order to allow for specification of attributes of the content portion.

When an object class description specifies content, content portion descriptions can be derived for the specific structure.

A basic object class description with a generic content portion description may be referred to by more than one basic object description of the same specific structure and of the same object class, which permits content sharing.

7.5.6 Generic logical structure

The object classes that can occur within the generic logical structure of a document are for objects of the following object types:

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

The possible hierarchical relationships between logical object classes are as follows:

- the object class for the document logical root can specify that there shall be any number and combination of immediately subordinate composite logical objects and basic logical objects;
- an object class for a composite logical object can specify that there shall be any number and combination of immediately subordinate composite logical objects and basic logical objects;
- an object class for a basic logical object can optionally have content specified; if content is specified then it may be in the form of one or more generic content portions, specification of a content generator, or a reference to an object class in a resource-document which has content specified.

Object classes of these object types serve as templates for objects of corresponding types in the specific logical structure.

7.5.7 Generic layout structure

The object classes that can occur within the generic layout structure of a document are for objects of the following object types:

- document layout root;
- page set;
- composite or basic page;
- frame;
- block.

The object classes for pages within a generic layout structure may either be basic pages or composite pages, but not a combination of these.

The possible hierarchical relationships between layout object classes are as follows:

- the object class for the document layout root can specify that there shall be any number and combination of immediately subordinate pages or page sets;
- an object class for a page set can specify that there shall be any number and combination of immediately subordinate pages or page sets;
- an object class for a composite page can specify that there shall be either any number of immediately subordinate frames or any number of immediately subordinate blocks;
- an object class for a frame can specify that there shall be either any number of immediately subordinate frames or any number of immediately subordinate blocks;
- an object class for a basic page or a block always has content specified; this may be by specifying one or more generic content portions, by specifying a content generator, or by referencing an object class in a resource document which has content specified.

Object classes of these object types serve as templates for the corresponding objects in the specific layout structure. Basic objects created by the layout process, for the content associated with the logical structures, do not reference layout object classes.

Object classes for basic pages and object classes for composite pages that specify immediately subordinate blocks shall only be included within documents of the formatted document architecture class.

7.5.8 Resource-document

A logical object class description may contain a reference to a logical object class description in the resource-document (see 6.3.12). A layout object class description may contain a reference to a layout object class description in the resource-document. In both cases, a subset or all of the attributes may be present in the object class description of the interchanged document only implicitly, by reference to the object class description of the resource-document. This may include the attribute “content portions”, in which case content portions may be present in the interchanged document only implicitly, by reference to the content portion descriptions of the resource-document.

The association between the interchanged document and the resource-document is achieved through the use of resource names. The resource-document includes, within the document profile, a table which maps resource names onto the object class identifiers of that resource-document. An object class description within the interchanged document may then refer to an object class description within the resource-document by using one of these resource names.

Attributes in the resource-document that are used by reference as attributes of object class descriptions in the interchanged document shall not make use of object class identifiers.

7.6 Colour

7.6.1 Overview

One goal of the ITU-T Rec. T.410-Series | ISO/IEC 8613 is to support efficient device independent interchange of documents containing coloured layout components or coloured contents. It enables the originator to transmit a precise specification of intent with respect to colour.

In order to achieve this goal, a colorimetrically precise reference colour space is used to specify interchange colour spaces.

7.6.2 Reference colour space

The reference colour space is based on the XYZ tristimulus values of the 1931 CIE Standard Colorimetric Observer, described in CIE Publication S002. All interchange spaces are defined in terms of this space.

NOTES

1 The reference colour space is colorimetrically precise and covers all perceivable colours. It is based on properties of the human visual system, determined by extensive experiments in colour matching, rather than on the properties of any particular device.

2 It is recognised that the general problem of appearance matching has not been completely solved. However, this Specification uses the best available and internationally recognised approach, which is the CIE system of colorimetry.

3 The reference colour space is normalised such that the Y tristimulus value is 1 for the reference white to allow for simplicity of conversion from colorimetric values to other colour spaces. This differs from the CIE recommendation of normalising Y of the perfect white reflecting diffuser (reference white) to exactly 100. See H.1 and H.2.

7.6.3 Interchange colour spaces

The interchange colour spaces for document level attributes as well as for character content architecture, raster graphics content architecture and geometric graphics content architecture are RGB and CMY(K). CMY(K) references both the CMY colour space and the CMYK colour space. The CMY colour space does not include a black component. The CMYK colour space has a place holder for a black value which may be zero. RGB allows the use of the reference colour space as well as other colour spaces based on self-luminous primary colours. In addition to these colour spaces, CIELUV and CIELAB are permitted for raster graphics content architecture.

Multiple interchange colour spaces are allowed to accommodate many different sources of colour data. All allowed interchange colour spaces have the common feature that they are defined in terms of the reference colour space. Transformations from each interchange colour space to the reference space are documented in Annex H. These transformations are defined in the data stream by additional interchange data called calibration data.

NOTE – The capability for more than one interchange space is required because:

- content architectures have different requirements;
- different applications have different requirements for representations of colour;
- conformance is required with other standards that are related to a content architecture.

7.6.4 Colour imaging model

The following is a conceptual model of one way in which interchanged colour content may be imaged.

An imaging system converts a colour value specified by one of the interchange colour spaces to the reference space. It then converts it from the reference space to the device space for its imaging device. This applies for both input and output devices. An imaging system may also utilize the colour values directly without conversion. See Figure 12 for a description of these colour space relationships at the document architecture level. This Figure also represents the common colour spaces used across all content architectures.

NOTES

1 Although this is conceptually two operations, it can be implemented as one transformation.

2 One practical difficulty which can arise during this process is the need to produce colours which are outside the gamut of the imaging device. In addition, an imaging device may have a limited ability to reproduce colour. The methods for addressing these problems are left for future study. See also Annex J.

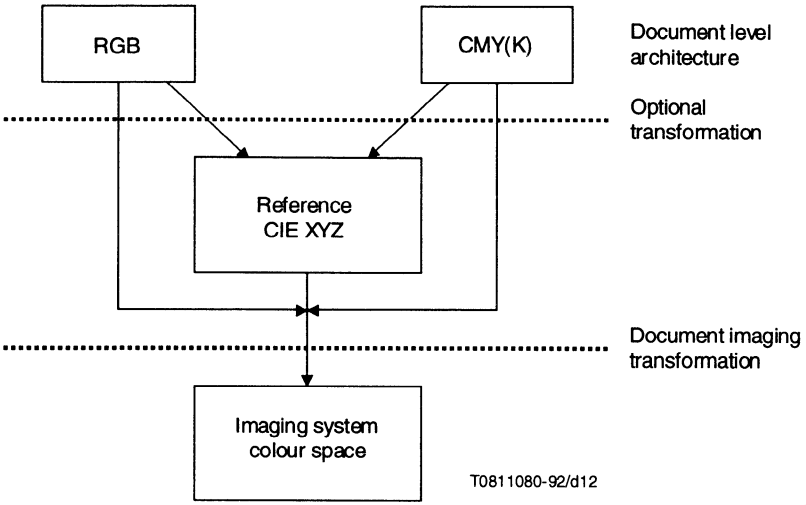

RGB
CMY(K)
Document level architecture
Optional transformation
Reference CIE XYZ
Document imaging transformation
Imaging system colour space
T0811080-92/d12

Figure 12 – Relationships among the colour spaces for the document level architecture

7.6.5 Colour tolerance

In some instances, the originator of a colour description may specify some tolerance or permit some variability in the interpretation of colour values. An example is the traditional usage within CGM of RGB values which do not refer to any particular set of RGB primaries or to any particular reference white. (See H.3.1 and H.3.2). Therefore, this Recommendation | International standard uses the attribute "colour tolerance" to allow the originator to specify colour differences, the amount by which colour values can vary during presentation (or processing) and still satisfy the original intent. The colour differences are specified using one of the two CIE-recommended uniform colour spaces – CIELUV and CIELAB (see Annex H and CIE Publication, 15.2, 2nd Edition, 1986).

8 Interface between the document architecture and content architectures

8.1 General concept

A document is defined by means of:

- The *document architecture* – A finite set of constituents together with their characteristics and rules that establish their relationships.
- The *content architectures* – A finite set of content elements, (for example, graphic characters, pels), together with their characteristics and rules that establish their relationships.

Content elements and control functions make up content portions governed by a content architecture. Every content portion is associated with a basic component governed by the document architecture.

One content element is thus specified within the document through a specific content architecture together with the document architecture.

The terms basic values and non-basic values are used in this clause with the following meanings:

- *basic values* of attributes, control function parameters and other capabilities are those that are unconditionally allowed in document interchange in the context of a particular document application profile;
- *non-basic values* of attributes, control function parameters and other capabilities are only allowed in document interchange in the context of a particular document application profile, if their use is declared in the document profile.

The document architecture and any content architectures are connected through an interface, as shown in Figure 13.

The interface is specified by describing which attributes defined in a content architecture have influence on constituents of the document architecture (document profile, basic component descriptions, content portions).

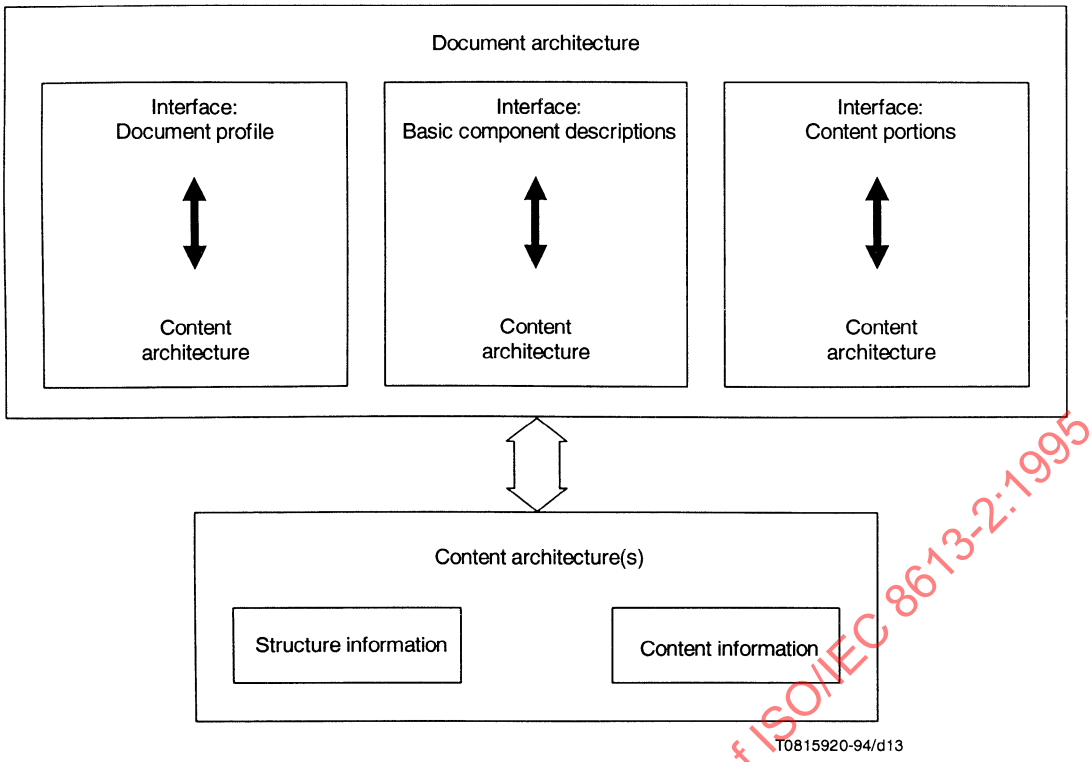


Figure 13 – Document architecture – Content architecture interface

8.2 Specification of a content architecture

The specification of a content architecture consists of three categories of information:

- structure information that identifies the class of content architecture, its internal structure rules and the positioning and imaging rules;
- content information that determines what information comprises the content portion. It includes a set of content element repertoires along with a default repertoire and the set of control functions available in the content architecture;
- information that specifies the basic values, default values and non-basic values of presentation attributes, coding attributes and control function parameters.

This information is required in order to link the document architecture to content architecture(s), through the interface information.

8.3 Interface information

A content architecture has the following three interfaces to the document architecture:

- the *document profile*, which includes attributes that identify the content architecture class(es) used within the document and attributes that specify the use of any non-basic content architecture features;
- an *object description* or an *object class description* for a basic object, which include attributes that identify the content architecture class and presentation attributes;
- a *content portion description*, which includes attributes that identify the type of coding and coding attributes.

The definition of each content architecture specifies the information associated with these attributes as described in the following sub-clauses.

8.3.1 Interface between a content architecture and the document profile

The definition of a content architecture shall include the following information:

- the value(s) of the attribute used to identify the content architecture class in the document profile;
- the format of the attribute used to specify non-basic values of presentation attributes in the document profile;
- the format of the attribute used to specify default values of presentation attributes in the document profile;
- the format of the attribute used to specify non-basic values of coding attributes in the document profile;
- the format of the attribute used to specify default values of coding attributes in the document profile.

The distinction between basic and non-basic values of presentation attributes and coding attributes is not specified as a part of the definition of a content architecture. This distinction is made in the definition of a document application profile in accordance with ITU-T Rec. T.411 | ISO/IEC 8613-1.

8.3.2 Interface between a content architecture and a basic component description

A definition of a content architecture shall include the following information:

- the value(s) of the attribute used to identify the content architecture class in a basic component description;
- the formats, permissible values and recommended default values of the presentation attributes;
- the presentation attributes for which a document application profile may define non-standard default values;
- possible interactions between presentation attributes and document architecture attributes.

8.3.3 Interface between a content architecture and a content portion

The definition of a content architecture shall include the following information:

- the value(s) of the attribute used to identify the type of coding in a content portion;
- the formats, permissible values and recommended default values of the coding attributes;
- the coding attributes for which a document application profile may define non-standard default values;
- possible interactions between control functions and presentation attributes.

9 Attribute definitions

9.1 General principles of attributes

Characteristics of structural elements of a document and relationships between structural elements are represented by constituents which are sets of attributes. Each attribute is identified by a name and has a value that describes the characteristic or relationship. Attributes are also used to identify constituents.

This clause contains definitions of all attributes defined in the document architecture. The attribute definitions specify the range of values that each attribute can assume.

9.1.1 Attribute categorisation

Attributes are categorised according to the constituent to which they apply, as follows:

- document profile attributes;
- component description attributes;
- layout style attributes;
- presentation style attributes;
- content portion description attributes;
- protected part attributes.

The constituents termed descriptions are those which are counterparts of the structural elements (objects, object classes and content portions).

Document profile attributes are defined in ITU-T Rec. T.414 | ISO/IEC 8613-4. The general principles of other attribute categories are described in this sub-clause and the attributes are defined in 9.3 to 9.10.

NOTE – The tables in Annex E summarise all attributes defined in this clause, in accordance with their categorisation.

9.1.1.1 Attributes of components

Attributes of component descriptions are further categorized as follows:

- *Shared attributes* – These can be included in both logical and layout component descriptions (see 9.3).
- *Layout attributes* – These can be included in layout component descriptions only (see 9.4).
- *Logical attributes* – These can be included in logical component descriptions only (see 9.5).

Some attributes of components can be included in composite component descriptions only, some in basic component descriptions only and some only in component descriptions of a particular object type.

In addition, some attributes may be included in object class descriptions only, some in object descriptions only and some in both object class descriptions and in object descriptions.

All attributes, except the presentation attributes (see 9.1.1.4), are independent of the content architecture pertaining to any component descriptions.

9.1.1.2 Layout style attributes

A layout style consists of the attributes:

- layout style identifier;
- user-visible name;
- user-readable comments;
- application comments;
- sealed;
- a set of layout directive attributes.

Layout styles are referred to from logical component descriptions. Such a reference is made using the attribute "layout style" with a value equal to the value of the layout style identifier. The layout style identifier uniquely identifies the layout style within the document.

A layout style that explicitly specifies all of the appropriate attributes that relate to that layout style is termed a root layout style. Any number of additional layout styles may be derived from a root layout style. The derived layout styles will specify only the attributes and/or attribute values that differ from the root layout style. This provides a means for factoring attributes thus preventing the necessity of copying the same attributes in similar layout styles. Any number of levels of derived layout styles may be provided by first specifying a layout style derived from the root layout style and then specifying other layout styles derived from derived layout styles.

A layout style may be referenced by more than one logical component description. The effect of a reference to a layout style is to apply its layout directive attributes to that logical component description which contains the reference. The layout directive attributes are a set of attributes which specify information for the document layout process (see clause 10). Precedence rules are specified in 9.1.2.4, 9.1.2.6 and 9.7.16.

Layout directive attributes are defined in 9.7.

Thus, layout styles affect the layout of objects, not their content and are independent of particular content architectures. Presentation styles affect the layout and imaging of the content associated with basic objects and hence are content architecture specific. Therefore, there is no conflict between the two.

9.1.1.3 Presentation style attributes

A presentation style consists of the attributes:

- presentation style identifier;
- user visible name;
- user-readable comments;
- application comments;

- transparency;
- border;
- sealed;
- colour;
- colour of layout object;
- object colour table;
- content background colour;
- content foreground colour;
- content colour table;
- sets of presentation attributes particular to each content architecture.

A presentation style may be referred to from a basic logical or basic layout component description. Such a reference is made using the attribute "presentation style" with a value equal to the value of the presentation style identifier. The presentation style identifier uniquely identifies the presentation style within the document.

A presentation style may be referenced by more than one component description.

A presentation style that explicitly specifies all of the appropriate attributes that relate to that presentation style is termed a root presentation style. Any number of additional presentation styles may be derived from a root presentation style. The derived presentation styles will specify only the attributes and/or attribute values that differ from the root presentation style. This provides a means for factoring attributes thus preventing the necessity of copying the same attributes in similar presentation styles. Any number of levels of derived presentation styles may be provided by first specifying a presentation style derived from the root presentation style and then specifying other presentation styles derived from derived presentation styles.

The effect of a reference to a presentation style is to apply its presentation attributes, transparency, colour and border to the basic component which contains the reference. In addition, presentation attributes can be specified for basic layout components. Precedence rules are specified in 9.1.2.4 and 9.1.2.6.

Each set of presentation attributes corresponds to a particular content architecture class.

Presentation styles affect the layout and imaging of the content associated with basic objects and hence are content architecture specific. Layout styles affect the layout of objects, not their content. Therefore, there is no conflict between the two.

9.1.1.4 Presentation attributes

Presentation attributes, which can be applied to basic components only, are used to specify the properties of the content portion(s) associated with that component. The presentation attributes specify information for the content layout process and content imaging process.

The presentation attributes specify the initial conditions concerned with the presentation of the content portion(s). The presentation attributes are specified either in a presentation style or, for layout components, in a basic layout component description.

The set of presentation attributes that is applicable depends upon the particular content architecture class specified for the basic component. The content architecture class attributes specify the content architecture class of the associated content portion(s), and therefore which set of presentation attributes are applicable to the content (see 9.3.4).

The presentation attributes are defined in those Specifications of the ITU-T Rec. T.410-Series | ISO/IEC 8613 that are concerned with individual content architectures.

9.1.1.5 Content portion description attributes

A content portion description consists of the attributes:

- content identifier-logical;
- content identifier-layout;
- type of coding;
- alternative representation;
- coding attributes;
- content information.

These attributes uniquely identify the content portion description, specify the type of coding used to code the content information and specify an alternative representation that may be imaged in lieu of content information when a recipient is not capable of decoding and/or imaging the content portion.

Content portion identifiers and alternative representation are fully specified in 9.9. Other attributes of content portions depend on the content architecture and details are specified in those Specifications of the ITU-T Rec. T.410-Series | ISO/IEC 8613 that deal with individual content architectures.

9.1.1.6 Protected part attributes

There are four kinds of descriptions of protected parts of a document, one for sealed information and three for enciphered information. They are:

A sealed document profile description consists of the attribute "sealed document profile information", which is a document profile, with a similar structure to a regular document profile. The only differences are that:

- every attribute is optional;
- only the attributes that are sealed shall be present.

It is also possible to seal absent attributes.

An enciphered document profile description consists of two attributes:

- *Protected part identifier.*
- *Enciphered information* – The value of this attribute is the result of an encipherment of the confidentially protected part of the document profile. The confidentially protected part of the document profile has a structure similar to a regular document profile. The differences are that:
  - every attribute is optional;
  - only the attributes that are confidential shall be present.

A pre-enciphered document body part description consists of two attributes:

- *Protected part identifier.*
- *Enciphered information* – The value of this attribute is the result of encipherment of the confidential part of the document body applied before the layout process has been performed.

A post-enciphered document body part description consists of two attributes:

- *Protected part identifier.*
- *Enciphered information* – The value of this attribute is the result of encipherment of the confidential part of the document body applied after the layout process has been performed.

9.1.2 Attribute values

9.1.2.1 General principles

Each constituent consists of a set of attributes; the attributes in the set are said to be *specified* for the particular constituent.

In the case of object descriptions, the complete set of attributes describing the characteristics of the object need not be specified for the constituent; such attributes can also be derived from attributes specified for other constituents, by using defaulting rules specified in 9.1.2.4. In such cases, the attributes which are specified for the constituent, together with those derived for the constituent, are said to form the set of attributes which *apply* to the constituent (or that are applied to, or are applicable to, the constituent).

Individual attributes may be structured into parameters, sub-parameters, sub-sub-parameters, etc., each attribute definition includes a specification of any such structuring.

Attributes may be formed from a set consisting of a list of attributes. At each point in this specification when an attribute is formed from a set of attributes, a list of the attributes that may be included in the set is explicitly specified.

The definition of an attribute also indicates whether the value(s) of that attribute may consist of one or more elements of data or expression(s).

Attribute values in the document profile, presentation styles and content portion descriptions can only consist of data elements of defined types; attribute values in component descriptions and layout styles may consist of either data elements or expressions.

If the value of an attribute consists of one or more data elements, then each element will be of a certain type and will be taken from a pre-defined set of values.

An attribute, parameter, sub-parameter, etc., value may consist of:

a) a sequence of numeric values;

b) a string of characters from a defined character set;

c) an element from a set of data elements defined for that attribute;

d) a reference to another constituent that exists within the document;

e) a reference to a value of an attribute of a constituent that exists in the document.

An example of a value of type c) is the sub-parameter "fill order" of the attribute "position" (see 9.4.1.1) which can have a value equal to a member of the set {'normal order', 'reverse order'}. An example of a value of type d) is the attribute "presentation style" (see 9.3.3.5). An example of a value of type e) is the parameter "binding name" of the attribute "bindings" (see 9.3.5.4).

Alternatively, if a value consists of an expression, then the expression will need to be evaluated in order for the value to be interpreted. This value may depend upon the values of attributes in other constituents (see 9.1.3).

9.1.2.2 Attribute classification

Attributes can be classified as mandatory, defaultable or non-mandatory.

The classifiers are M, D, NM, as follows:

- M – Mandatory attribute: The attribute shall be specified for the constituent.
- D – Defaultable attribute: The attribute need not be specified for the constituent; the value can be derived using the defaulting mechanisms specified in 9.1.2.4 and 9.1.2.5.
- NM – Non-mandatory attribute: The attribute need not be specified for the constituent. If the attribute is not specified for a constituent, the attribute does not apply to that constituent.

Attributes of object descriptions, content portion descriptions and styles when applied to object descriptions, can be classified as either mandatory, defaultable or non-mandatory; attributes of all other constituents can only be classified as mandatory or non-mandatory.

In the case of defaultable attributes, the attribute definition also specifies the standard default value (see 9.1.2.4 and 9.1.2.5).

A document application profile may specify non-standard default values for attributes (see ITU-T Rec. T.414 | ISO/IEC 8613-4). If this is the case, they are declared in the document profile by means of the document profile attribute "document application profile defaults".

9.1.2.3 Default value lists

Certain attributes that are classified as defaultable may be specified in a default value list. Default value lists may be specified for composite component descriptions.

The purpose of a default value list is to allow the values of some attributes, which describe characteristics of objects, to be specified in constituents that correspond to objects at higher levels of the same specific structure. The use of a default value list forms part of the defaulting mechanism defined in 9.1.2.4.

A composite component description may specify one or more default value lists. If more than one list is specified, each list applies to a different object type that may be applicable to subordinate objects. Each default value list applies to all subordinate objects of the relevant object type.

For example, a default value list specified for a page may apply to subordinate frames or to subordinate blocks within the page.

9.1.2.4 Determining values for defaultable attributes of objects

The values of defaultable attributes of objects may be derived from:

1) within the object description;

2) within the object class description;

3) within a style;

4) within a default value list at a higher level of the hierarchical structure;

5) within an object class description in a referenced resource-document;

6) within a list of default values in the document profile, representing default values specified by a document application profile;

7) within the ITU-T Rec. T.410-Series | ISO/IEC 8613.

In case 4) the attributes are interpreted as default values for the lower levels. They may be overridden by attributes of object descriptions, styles or object class descriptions specified at the lower levels.

For example, using default value lists, it is possible to specify:

- at document layout root level, the default page size;
- at page level, the default line spacing for blocks containing character content.

The default values for attributes applicable to logical object descriptions are determined in the sets of constituents representing the specific and generic logical structures and referenced presentation and layout styles.

The default values for attributes applicable to layout object descriptions are determined in the sets of constituents representing the specific and generic layout structures and referenced presentation styles.

To determine the value of an attribute of an object that is classified as defaultable, the value is determined by the first of the following rules which is applicable.

When an attribute is structured into two or more parameters then it may be specified that the defaulting rules are to apply to each parameter independently. When parameters are structured into sub-parameters, sub-sub-parameters, etc, it may further be specified that the defaulting rules are to apply to each of these substructures independently. In all cases, when this is possible it is explicitly specified in the attribute definition, in the part of the definition which specifies the default values, by explicit identification of the parameters, sub-parameters, sub-sub-parameters, etc., for which independent defaulting rules are to apply:

a) If a value for the attribute is specified for the object description concerned, then that value is used.

b) If the object description concerned refers to a style and a value of the attribute is specified or derived for that style (see 9.1.2.6), then that value is used.

c) If the object description concerned refers to an object class description. and that object class description specifies a value for the corresponding attribute, then the value of the attribute is derived from that corresponding attribute.

d) If the object description concerned refers to an object class description, which specifies a reference to a style. and a value of the attribute in specified or derived for that style (see 9.1.2.6), then that value is used.

e) If the object description concerned refers to an object class description which refers to an object class description in the resource-document, and the object class description in the resource-document specifies a value for the corresponding attribute, then the value of the attribute is derived from that corresponding attribute.

f) If the object description concerned refers to an object class description which refers to an object class description in the resource-document which specifies a reference to a style, and a value of the attribute is specified or derived for that style (see 9.1.2.6), then that value is used.

g) The attribute value may be determined from a default value list according to the set of rules specified below. The value is determined by the first of the rules that returns a value.

   The set of rules is to be applied for each object description at each hierarchical level in the structure, starting with the hierarchical level immediately superior to the object description for which the attribute value is required and then applying the rules to each higher level superior object description in turn. If the document root is reached and no value is found, then rule g) does not determine a value.

   i) If the object description contains a default value list which is applicable to the object description for which the attribute is required and if the default value list contains a value for the attribute, then the value of the attribute is derived from the value specified in that default value list.

ii) If the object description contains a default value list which is applicable to the object description for which the attribute value is required and if the default value list refers to a style, and a value of the attribute is specified or derived for that style (see 9.1.2.6), then that value is used.

iii) If the object description refers to an object class description containing a default value list which is applicable to the object description for which the attribute is required and if the default value list contains a value for the attribute, then the value of the attribute is derived from the value specified in that default value list.

iv) If the object description refers to an object class description containing a default value list which is applicable to the object description for which the attribute value is required and if the default value list refers to a style, and a value of the attribute is specified or derived for that style (see 9.1.2.6), then the value of that attribute is used.

v) If the object description refers to an object class description which refers to an object class description in a resource document containing a default value list which is applicable to the object description for which the attribute is required and if the default value list contains a value for the attribute, then the value of the attribute is derived from the value specified in that default value list.

vi) If the object description refers to an object class description which refers to an object class description in a resource document containing a default value list which is applicable to the object description for which the attribute value is required and if the default value list refers to a style, and a value of the attribute is specified or derived for that style (see 9.1.2.6), then the value of that attribute is used.

h) If a default value is defined for this attribute by the document profile attribute “document application profile defaults” (see ITU-T Rec. T.414 | ISO/IEC 8613-4), then that value is used.

j) The default value defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613 is used.

NOTE – Annex D is a non-integral annex which summarizes some aspects of the defaulting mechanism. In particular, it includes a table summarizing the steps of the defaulting mechanism described in 9.1.2.4 as these apply to the various attributes.

9.1.2.5 Determining values of attributes of content portions

The content associated with a basic object in a specific structure is determined by the first of the following to specify either any content portions or a content generator:

- the basic object description;
- an object class description referenced from the basic object description;
- an object class description in the resource-document referenced from an object class description referenced from the basic object description.

To determine the value of an attribute of a content portion that is classified as defaultable, the value is determined by the first of the following rules which is applicable.

a) If a value for the attribute is specified for the content portion description concerned, then that value is used.

b) If a default value is defined for this attribute by the document profile attribute “document application profile defaults” (see ITU-T Rec. T.414 | ISO/IEC 8613-4) then that value is used.

c) The default value defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613 is used.

9.1.2.6 Determining values for attributes of styles

To determine the value of an attribute of a layout style or presentation style, the value is determined by the first of the following rules which is applicable.

a) If a value for the attribute is specified for the style concerned, then that value is used.

b) If the style concerned is derived from another style, and that style specifies a value for the attribute then that value used.

c) If the style concerned is derived from another style, and that style is derived from other styles at any number of levels including the root style and one or more of the styles specifies a value for the attribute, then the attribute value determined from the lowest level style is used.

d) No value is determined for the attribute. (For defaultable attributes, see 9.1.2.4).

9.1.3 Expressions

The value of some attributes can be specified by an expression. These attributes are:

- generator for subordinates;
- content generator;
- bindings, for the parameter "binding value";
- same layout object, for the first parameter;
- synchronization.

The expression permitted in the attribute "generator for subordinates" is a *construction expression,* as defined in 9.3.2.1. There are three other types of expression:

a) string expressions (see 9.1.3.1);

b) numeric expressions (see 9.1.3.2);

c) object identifier expressions (see 9.1.3.3).

A string expression or a numeric expression may refer to a binding value (see 9.1.3.4).

9.1.3.1 String expressions

A *string expression* within an attribute value specification consists of either an atomic string expression or a sequence of two or more atomic string expressions.

An atomic string expression is one of the following:

a) a string literal;

b) a reference to a binding value (see 9.1.3.4);

c) a string function application (see below).

A string literal is an arbitrary octet string.

NOTE – This octet string may be interpreted in various ways, depending on the application. For example, as a character string of a particular character repertoire, as a facsimile image or as a geometric picture.

A string function application is an application of one of the functions:

- MAKE-STRING – This function produces a character string consisting of the decimal numeral representing the numeric value of the parameter, which may be any integer (negative, zero or positive integers are permitted).
- UPPER-ALPHA – This function produces a character string consisting of one of the upper case letters A to Z, corresponding to the numeric value (1 to 26) of the parameter.
- LOWER-ALPHA – This function produces a character string consisting of one of the lower case letters a to z, corresponding to the numeric value (1 to 26) of the parameter.
- UPPER-ROMAN – This function produces a character string consisting of the Roman numeral representing the numeric value of the parameter, which may be any positive integer, using the upper case letters: C,D,I,L,M,V and X.
- LOWER-ROMAN – This function produces a character string consisting of the Roman numeral representing the numeric value of the parameter, which may be any positive integer, using the lower case letters: c,d,i,l,m,v and x.

In each case, the character string returned consists of characters taken from the minimum subrepertoire of ISO/IEC 6937.

Each of these functions has a parameter consisting of a numeric expression (see 9.1.3.2).

A string function application produces an empty string when the value of the parameter is out of range.

When two or more atomic string expressions occur in sequence each expression is evaluated independently and their results are concatenated.

9.1.3.2 Numeric expressions

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

a) a numeric literal;

b) a reference to a binding value (see 9.1.3.4);

c) a numeric function application (see below).

A numeric literal is any integer (i.e. negative, zero or positive integers are permitted).

A numeric function application is an application of one of the functions:

- INCREMENT – This function has one parameter, consisting of a numeric expression. The result of the function is a numeric value which is one greater than the value of the parameter.
- DECREMENT – This function has one parameter, consisting of a numeric expression. The result of the function is a numeric value which is one less than the value of the parameter.
- ORDINAL – This function has one parameter, consisting of a reference to an object, represented by either an object identifier or an object identifier expression.

  The result of the function is equal to the sequence number of the referenced object, within the set of objects that are immediately subordinate to the immediate superior of the referenced object and belong to the same object class as the referenced object. Within the set of objects that satisfy this criterion, the objects are ordered according to the sequential order and are numbered using the positive integers 1, 2, 3, etc.

The numeric value range for numeric expressions is the set of all integers (i.e. negative, zero and positive integers are permitted).

9.1.3.3 Object identifier expressions

Some attributes that apply to object descriptions have values that contain references to other object descriptions.

In an object class description or a layout style, the value of such an attribute can be represented by an object identifier expression. This is an expression that, when it is evaluated in the proper context, produces the value of an object identifier.

An *object identifier expression* consists of a reference to an object selection function and the specification of one or two parameters if required by the object selection function concerned.

Object selection functions are:

- CURRENT-OBJECT – A function without a parameter, which produces the identifier of the object description to which the attribute applies whose value is represented by the object identifier expression.
- CURRENT-INSTANCE – A function with two parameters, which produces the identifier of that instance of an object of the object class or object type specified by the first parameter that is current relative to the position in the specific structure of the object specified by the second parameter. (This function is explained further in 9.1.3.5).
- SUPERIOR-OBJECT – A function with one parameter, which produces the identifier of the object description that is immediately superior to the object description identified by the parameter. The parameter consists of an object identifier expression.
- PRECEDING-OBJECT – A function with one parameter, which produces the identifier of the object description that immediately precedes the object description specified by the parameter in the sequential order (see 7.1.2). The parameter consists of an object identifier expression.

Any object selection function will produce a null object identifier when an attempt is made to reference a non-existent object (for example, the object superior to the document layout root, or the document logical root).

9.1.3.4 References to binding values

Within string and numeric expressions it is possible to refer to the value of a binding within the attribute "bindings" of either the object description to which the attribute applies, in which the expression occurs, or any other specified object description. Such a reference is represented by the combination of a reference to an object description and a binding name (see 9.3.5.4).

The reference to the object description is represented by an expression which when evaluated produces an object identifier.

Such an expression is called a *binding reference expression*. Its format is the same as that of an object identifier expression, defined in 9.1.3.3, but its semantics are as described below.

A binding reference expression consists of either an object identifier or one of the following four functions which, in this context, are called *binding selection functions*:

- CURRENT-OBJECT;
- CURRENT-INSTANCE;

- SUPERIOR;
- PRECEDING.

The parameters of *binding selection functions* are as defined in 9.1.3.3; in particular object identifier expressions used as parameters have the semantics described there and are not in turn interpreted as binding selection function applications.

The semantics of the binding selection functions CURRENT-OBJECT and CURRENT-INSTANCE are identical to those of the corresponding object selection functions defined in 9.1.3.3 and 9.1.3.5.

The semantics of the binding selection functions SUPERIOR and PRECEDING are similar to those of the object selection functions SUPERIOR-OBJECT and PRECEDING-OBJECT defined in 9.1.3.3.

The differences are that, if the result of the function is the identifier of an object description to which no binding applies that has the specified binding name, then the invocation of the function is repeated, using the identifier of the failing object description as the function parameter, until an object description is found to which a binding applies that has the specified binding name.

In the case that the binding selection function is SUPERIOR, the invocation of the function is repeated on increasingly superior object descriptions, starting from the object description immediately superior to the object description specified by the parameter, until either a binding of the specified binding name is located or the document logical root or document layout root is reached.

In the case that the binding selection function is PRECEDING, the invocation of the function is repeated through the object descriptions in the reverse of the sequential order of the objects in the structure, starting from the object description immediately preceding the object description specified by the parameter, until either a binding of the specified binding name is located or the document logical root or document layout root is reached.

If the specified object identifier or the result of the binding reference expression does not pertain to an object to which a binding with the specified binding name applies, then the result of the reference to the binding value is either an empty string, the integer zero, or a null object identifier, depending on whether a string, a numeric value or an object identifier expression is expected in the context concerned.

The value of the binding which is referenced may contain an expression, which may itself refer to further expressions.

9.1.3.5 Current instance function

The CURRENT-INSTANCE function may be used in object identifier expressions (see 9.1.3.3) and in binding reference expressions (see 9.1.3.4). It has two parameters. The first parameter is either an object class identifier or an object type.

The second parameter consists of a reference to a logical object or layout object and is represented by either an object identifier or an object identifier expression.

The result of the function is the object identifier of that instance of the object class or object type specified by the first parameter, that is current relative to the position in the document corresponding to the logical or layout object specified by the second parameter.

To determine the result of the function, four cases are distinguished:

a) the first parameter specifies a logical object class or logical object type, and the second parameter refers to an object that is part of the specific logical structure (that is, excluding any logical objects generated as a result of the attribute "logical source");

b) the first parameter specifies a layout object class or layout object type, and the second parameter specifies a logical object that either has or has not been generated as a result of the attribute "logical source";

c) the first parameter specifies a logical object class or logical object type, and the second parameter specifies a temporary logical object that has been generated as a result of the attribute "logical source";

d) the first parameter specifies a logical object class or logical object type, and the second parameter refers to a layout object that is of a class that is referenced by at least one basic layout object without generic content.

   In case a), the logical object specified by the second parameter is called the "reference logical object" and

   - if the reference logical object belongs to the object class or object type specified by the first parameter, then the result of the function is the identifier of that logical object;
   - otherwise, the result of the function is the identifier of the nearest superior of the reference logical object that belongs to the object class or object type specified by the first parameter.

In case b), the "reference layout object" is defined as the first basic layout object in which is laid out any content of the logical object specified by the second parameter and

- if the reference layout object belongs to the object class or object type specified by the first parameter, then the result of the function is the identifier of that layout object;
- otherwise, the result of the function is the identifier of the nearest superior of the reference layout object that belongs to the object class or object type specified by the first parameter.

In case c), the "reference layout object" is defined as the first basic layout object in which is laid out any content of the logical object specified by the second parameter and

- the "reference logical object" is defined as the first basic object of the specific logical structure (that is, excluding any temporary logical objects generated as a result of the attribute "logical source") of which any content is laid out in a layout object that follows the reference layout object in the sequential order;
- if the reference logical object belongs to the object class or object type specified by the first parameter, then the result of the function is the identifier of that logical object;
- otherwise, the result of the function is the identifier of the nearest superior of the reference logical object that belongs to the object class or object type specified by the first parameter.

In case d), the "reference logical object" is defined as the first logical object of which any content is laid out in the layout object referenced by the second parameter and

- if the reference logical object belongs to the object class or object type specified by the first parameter, then the result of the function is the identifier of that logical object;
- otherwise, the result of the function is the identifier of the nearest superior of the reference logical object that belongs to the object class or object type specified by the first parameter.

Any current instance function will produce a null identifier when an attempt is made to reference a non-existent object. [(For example, in case a)] if neither the reference logical object nor any of its superiors are of the object class or object type specified by the first parameter.

9.1.4 Colour attribute values

Attributes for colour may be applied to both objects and to content. Within these attributes, colour is specified either directly or indirectly. Direct specification is done by specifying three or four colour component values. Indirect specification is done by specifying an index into a colour table. The characteristics of the interchange colour space that is referenced by the colour attributes are defined in the attribute "colour spaces list" which is contained in the document profile. Each entry in the attribute "colour spaces list" has a unique identifier, and a user specified name (without any semantics). The important information is the type of colour space, data scaling parameters and calibration specification. RGB, CMYK and CMY are the colour space types supported for document architecture components and for all content types.

Colour data scaling permits the range of colour data to be adjusted by means of a multiplicative scale factor and an additive offset. Calibration data can be provided for the different schemes defined for RGB, CMYK and CMY colour spaces.

Colour is applied to objects by the attribute "colour of layout object". This attribute is applied to layout object descriptions and may be applied to logical object descriptions by presentation styles. In order to use the attribute "colour of layout object", the attribute "transparency" must have the value 'opaque' and the attribute "colour" must have the value 'coloured'. The colour specified through the attribute "colour of layout object" may have a parameter "colour tolerance". The parameter "colour tolerance" specifies to what degree of precision an originator wishes to have a particular colour imaged, on a colour by colour basis.

If object colours are referenced by index, they must be defined by the attribute "object colour table". The characteristics of the interchange colour space are defined by an entry in the attribute "colour spaces list".

Colour is applied to content by the attributes "foreground content colour" and "background content colour". They provide the initial foreground and background colours for character content (in ITU-T Rec. T.416 | ISO/IEC 8613-6) and bi-level or bit mapped image (in ITU-T Rec. T.417 | ISO/IEC 8613-7). Both types of content colour point at an entry in the attribute "colour spaces list" to define the characteristics of the interchange colour space. Transparency within content colour is indicated on a colour by colour basis, unlike object colour.

If content colour is referenced by index, the content colour must be defined by the attribute "content colour table".

For describing colour, various attributes make use of *colour expressions*, the structure, permissible values, default value and the meaning of which are described in 9.1.4.1. Two forms of colour expressions are to be distinguished: *direct colour expressions* and *indexed colour expressions* which are selected by the subparameter "colour access mode". Indexed colour expressions refer to a colour specification in a relevant *colour table* (which depends on the attribute making use of the colour expression). The structure, permissible values, default value and meaning of colour tables are described in 9.1.4.2 below.

9.1.4.1 Colour expressions

Colour expressions occur as values for parameters, sub-parameters, etc., of various attributes specifying colour.

Structure:

A parameter "colour expression" is either a direct colour expression or an indexed colour expression. In either case, a sub-parameter "colour access mode" indicates whether a direct or indexed colour expression is used.

In the case of a direct colour expression, there are three optional sub-parameters:

colour space id;

colour specification;

colour tolerance.

For the sub-parameter "colour tolerance" one of the sub-sub-parameters "specified tolerance" or "unspecified tolerance" shall be specified. The sub-sub-parameter "specified tolerance" consists of two sub-sub-sub-parameters, "tolerance value" and "tolerance space". The sub-sub-parameter "unspecified tolerance" is not substructured.

In the case of an indexed colour expression, there is one optional sub-parameter "index" which references a colour within the applied colour table.

Permissible values:

colour access mode:	direct, indexed
a) When the value of "colour access mode" is 'direct':	
colour space id:	a non-negative integer
colour specification:	a sequence of three or four colour coordinate values (integers or real numbers)
colour tolerance:	
specified tolerance:	
tolerance value:	either a positive integer or a positive
real	number
tolerance space:	either CIELUV or CIELAB
unspecified tolerance:	infinite
b) When the value of "colour access mode" is 'indexed':	
index:	a non-negative integer

Default values:

The sub-parameters of a parameter colour expression are independently defaultable.

colour access mode:	direct

If the sub-parameter "colour access mode" has the value 'direct' then the default values are:

colour space id:	0
colour specification:	1,1,1
colour tolerance:	
unspecified tolerance:	infinite

If the sub-parameter “colour access mode” has the value 'indexed' then the default value is:

index: 0

NOTE – The default value for a colour expression is a white colour in RGB space without limit on tolerance (value 'infinite'). This value was chosen for compatibility with the previous edition of this Specification.

Definition:

In the case of a *direct colour expression* the colour is specified by identification of a colour space and of colour coordinates within that colour space. The sub-parameter “colour space id” indicates the colour space by specifying the unique identifier of the colour space. The sub-parameter “colour specification” specifies the colour coordinate values according to the colour space specified by the sub-parameter “colour space id”.

The sub-parameter “colour tolerance” defines the allowed difference between the colour as it is specified and all other colours that would satisfy the originator's intent. It either has the sub-sub-parameter “unspecified tolerance” with the value 'infinite' (no limit on tolerance) or it is given in terms of the sub-sub-parameter “specified tolerance” in which case the sub-sub-sub-parameter “tolerance value” specifies the number of colour difference units in the CIE uniform colour space specified by the sub-sub-sub-parameter “tolerance space”. In the case of RGB, CMYK and CMY colour specifications, transformation to the specified tolerance space can be achieved by means of the calibration mechanism and the formulae in Annex H.

In the case of an *indexed colour expression* colour is specified by a non-negative integer, which references a colour within the applied colour table.

9.1.4.2 Colour tables

Colour tables occur as values for various attributes specifying colour.

Structure:

An attribute whose value is a colour table has two parameters, “colour space id” and “colour table entries”.

The parameter “colour table entries” is a table that consists of one or more entries. Each entry consists of three sub-parameters,

index;

colour specification;

colour tolerance.

The sub-parameter “colour tolerance” is optional and is a choice between two sub-sub-parameters: “specified tolerance” and “unspecified tolerance”. The sub-sub-parameter “specified tolerance” consists of two sub-sub-sub-parameters, “tolerance value” and “tolerance space”. The sub-sub-parameter “unspecified tolerance” is not substructured.

Permissible values:

colour space id:	a non-negative integer
colour table entries:	
for each entry:	
index:	a non-negative integer
colour specification:	a sequence of three or four colour coordinate values (either real numbers or integers)
colour tolerance:	
specified tolerance:	
tolerance value:	either a positive integer or a positive real number
tolerance space:	either CIELUV or CIELAB
unspecified tolerance:	infinite

Default values:

colour space id:			0
colour table entries:			
entry:	index:		0
	colour specification:		1.0, 1.0, 1.0
	colour tolerance:		
		unspecified tolerance:	infinite
entry:	index:		1
	colour specification:		0.0, 0.0, 0.0
	colour tolerance:		
		unspecified tolerance:	infinite
entry:	index:		2
	colour specification:		1.0, 0.0, 0.0
	colour tolerance:		
		unspecified tolerance:	infinite
entry:	index:		3
	colour specification:		0.0, 1.0, 0.0
	colour tolerance:		
		unspecified tolerance:	infinite
entry:	index:		4
	colour specification:		0.0, 0.0, 1.0
	colour tolerance:		
		unspecified tolerance:	infinite
entry:	index:		5
	colour specification:		1.0, 1.0, 0.0
	colour tolerance:		
		unspecified tolerance:	infinite
entry:	index:		6
	colour specification:		1.0, 0.0, 1.0
	colour tolerance:		
		unspecified tolerance:	infinite
entry:	index:		7
	colour specification:		0.0, 1.0, 1.0
	colour tolerance:		
		unspecified tolerance:	infinite

Definition:

Colour tables can be referenced to specify the colour of objects and content to which they apply.

The parameter “colour space id” identifies the colour space.

Each entry in the parameter “colour table entries” specifies the colour and the tolerance associated with the index value.

The sub-parameter “index” specifies the index value associated with that entry, and is the means of indexing into the colour table. The particular values used are arbitrary, and there need be no ordering within the colour table.

The sub-parameter “colour specification” specifies the colour coordinate values according to the colour space. The default values for the sub-parameter “colour specification” are in terms of the colour space with the identifier 0 which is an RGB space pre-defined in the ITU-T Rec. T.414 | ISO/IEC 8613-4.

The “colour tolerance” sub-parameter defines the allowed difference between the colour as it is specified and all other colours that would satisfy the originator's intent. It either has the value 'infinite' (no limit on tolerance) or it is given in terms of a “tolerance value”, which is the number of colour difference units in the CIE uniform colour space specified by the “tolerance space”. In the case of RGB, CMYK and CMY colour specifications, transformation to the specified tolerance space can be achieved by means of the calibration mechanism and the formulae in Annex H.

9.2 Attribute definition format

The attribute definitions in this clause are structured as follows:

Constituents:

States the types of constituents for which the attribute may be specified. For layout directive attributes, also states the types of logical components to which the attribute may be applied.

Classification:

States whether the attribute is classified as mandatory, non-mandatory or defaultable (see 9.1.2.2), and for which types of constituent.

Structure:

States the structuring of the attribute into parameters, sub-parameters, sub-sub-parameters, etc., if any. This entry is omitted if not applicable.

Permissible values:

States the permissible values of the attribute. If the attribute is structured into parameters, sub-parameters, sub-sub-parameters, etc., the permissible values are specified for these.

Representation:

States the representation of the values in the interchange format. Only stated in certain cases; in general this information is defined in ITU-T Rec. T.415 | ISO/IEC 8613-5. This entry is omitted if not applicable.

Default values:

States the default values of the attribute (see 9.1.2.4j). If the attribute is structured into parameters, sub-parameters, sub-sub-parmaters, etc., the default values are specified for these. This entry is omitted if not applicable.

Definition:

Textual description of the semantics of the attribute.

Exceptions:

States any exceptional cases that do not follow the general rules specified for the attribute definition. These exceptional cases occur when there are particular optimisations available with restricted usage of the attribute and to provide compatibility with previous standards.

9.3 Shared Attributes

The attributes defined in this sub-clause may be specified for more than one type of constituent. Attributes that may be specified only for logical components, or only for layout components, or only for one type of constituent, are described in subsequent subclauses.

9.3.1 Identification attributes

These attributes are used to identify uniquely the component to which they apply.

9.3.1.1 Object type

Constituents:

Component descriptions.

Classification:

Mandatory for all object class descriptions.

Defaultable for an object description which refers to an object class description, otherwise mandatory.

Permissible values:

A set of data elements defined for the attribute.

In the case of a layout component description:

document layout root;
page set;
composite or basic page;
frame;
block.

In the case of a logical component description:

document logical root;
composite logical object;
basic logical object.

Default value:

Never applies, since the value is always determined by one of the rules a) or c) in 9.1.2.4.

Definition:

This attribute specifies the object type. The object type determines the attributes that may be specified for the object description or object class description.

In the case of a layout object description, the attribute specifies whether the object is of object type:

- document layout root;
- page set;
- composite or basic page;
- frame;
- block.

A page is a composite page if it has any subordinates (see 9.3.3.2). It is a basic page if it has no subordinates.

In the case of a logical object description, the attribute specifies whether the object is of object type:

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

In the case of an object class description, the attribute specifies the object type of the objects in the object class.

9.3.1.2 Object identifier

Constituents:

Object descriptions.

Classification:

Mandatory, unless the exceptional case described below applies, in which case the attribute is non-mandatory.

Permissible values:

A sequence of non-negative integers. The value of the first integer is:

- 1, if the constituent is a layout object description;
- 3, if the constituent is a logical object description.

Representation:

A character string consisting of decimal numerals and space characters. The decimal numerals are in one to one correspondence with the integers constituting the identifier: a space character is used as a separator between successive numerals.

Definition:

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

An object identifier consists of a sequence of integers. Each integer in this sequence corresponds to a hierarchical level of the specific layout structure or specific logical structure and identifies one particular object description representing an object at that level.

The integers in this sequence start with the integer corresponding to the object description of the document layout root or document logical root. This is followed by each of the integers corresponding to the object descriptions on the path through the hierarchical structure from the document layout root or document logical root to the object description.

The first integer in the sequence indicates whether the identifier pertains to a layout object description or a logical object description. An object identifier consisting of just this first integer identifies the object description of the document layout root or document logical root.

The actual value of each subsequent integer is not significant; however the sequence of integers allocated to each object description shall be chosen so that each object description can be uniquely distinguished from all other object descriptions in the document.

Exceptions:

This attribute is non-mandatory in certain documents. These documents are those which have all of the following characteristics:

- The interchange format class used for document interchange is class B (see ITU-T Rec. T.415 | ISO/IEC 8613-5); consequently the only specific structure present is the specific layout structure.
- The only object types present in the document are document layout root, pages and blocks.
- There is no use of object identifiers in attributes.

In documents adhering to these rules any two consecutive objects of the same object type in the data stream have the same immediate superior. Thus, under these conditions, the semantics of the attribute "object identifier" can be transmitted to the recipient implicitly and the attribute need not be explicitly specified.

NOTE – This exceptional case is provided for compatibility with ITU-T Recommendations.

9.3.1.3 Object class identifier

Constituents:

Object class descriptions.

Classification:

Mandatory.

Permissible values:

A sequence of non-negative integers. The value of the first integer is:

- 0, if the constituent is a layout object class description;
- 2, if the constituent is a logical object class description.

Representation:

A character string consisting of decimal numerals and space characters. The decimal numerals are in one to one correspondence with the integers constituting the identifier: a space character is used as a separator between successive numerals.

Definition:

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

An object class identifier consists of a sequence of integers.

The first integer in this sequence indicates whether the identifier pertains to a layout object class description or a logical object class description. An object class identifier consisting of just this first integer identifies an object class description for the document layout root or document logical root.

The allocation of the other integers is not constrained, other than that the identifier of each object class shall be unique.

9.3.2 Construction attributes

These attributes specify rules for controlling the generation of object descriptions from object class descriptions and for controlling the generation of content.

9.3.2.1 Generator for subordinates

Constituents:

Composite object class descriptions.

Classification:

Non-mandatory.

In the case of a complete generator set of logical object class descriptions, this attribute is mandatory for composite logical object class descriptions. In the case of a complete generator set of layout object class descriptions, this attribute is mandatory for all composite layout object class descriptions except those for lowest level frames, for which it is non-mandatory.

In the case of a factor set of object class descriptions, this attribute shall not be specified.

Permissible values:

A construction expression (see definition below):

Definition:

This attribute specifies which objects, and which combinations of objects, may be immediately subordinate to an object of the class. In addition, this attribute specifies an ordering among these immediately subordinate objects.

The value of this attribute is an expression which can be evaluated in a number of ways to yield a set of possible values. Each value is a sequence of object class identifiers representing a sequence of object classes.

If the attribute is present in a composite object class description within a complete generator set of object class descriptions, then its set of possible values specify a constraint for all objects of the class, restricting the permissible immediately subordinate objects.

If the attribute is present in a composite object class description within a partial generator set of object class descriptions, then it does not constrain the immediately subordinate objects for objects of the class. However, it can be used as a guide for creating or editing the specific structure.

NOTE 1 – A partial generator set may specify the attribute "generator for subordinates" for all composite object class descriptions, but this need not be interpreted by a recipient as a constraint on generation of specific structures.

If the attribute is specified, then the constraint, in the case of a complete generator set of object class descriptions, or the guide, in the case of a partial generator set of object class descriptions, is as follows.

Each member of a set of object descriptions which has a common immediate superior has a value for the attribute "object class". If a sequence is formed consisting of the values of the attribute "object class" for all object descriptions in the set, in the order specified among those object descriptions by the attribute "subordinates" of their common immediate superior, then this sequence shall be one of the values that can be generated by the attribute "generator for subordinates" of the object class description of their common immediate superior.

If the attribute is absent in a composite object class description, then no constraints are specified for the set of immediately subordinate objects of objects of the class.

The value of this attribute consists of a construction expression. A *construction expression* specifies the identifiers of object class descriptions that can be used to generate immediately subordinate object descriptions of the object description being generated.

A construction expression is either a construction term (see below) or a construction type. A construction type is one of the following:

- a sequence construction, which consists of one or more construction terms, which are to be evaluated in the order specified;
- an aggregate construction, which consists of one or more construction terms, which are to be evaluated in an arbitrary order;
- a choice construction, which consists of one or more construction terms, 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.

Each construction factor is either an object class identifier or a construction type. In the former case, the value of the construction factor is the object class identifier. In the latter case, the value of the construction factor is derived by evaluation of the construction type. Evaluation of the construction type may produce either an empty sequence or a sequence of one or more object class identifiers.

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 need not be evaluated, when the containing construction 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 need not be evaluated, when the containing construction term is evaluated.

The rules for evaluation of construction terms and construction factors specify all the possible ways in which the construction expression may be evaluated, in order to define the set of all possible values referred to above. The rules for evaluation do not constrain the sequence of events within the document processing reference models.

NOTE 2 – Specifically, a subordinate may be introduced at any time in the reference editing process or reference layout process provided that its position amongst other subordinates follows the constraint or guide defined.

9.3.2.2 Content generator

Constituents:

Basic object class descriptions, basic logical object descriptions.

Classification:

Non-mandatory.

Permissible values:

A string expression.

Definition:

The value of this attribute is a string expression, which, when evaluated, produces the content associated with the object. String expressions are defined in 9.1.3.1.

When a content generator is specified for a logical object class, it is interpreted as providing the default value for this attribute of the logical object descriptions for objects of that object class.

For any component description, this attribute is ignored if there is more than one content portion or if there is a single content portion which specifies the attribute “content information”.

The attribute “content generator” is evaluated during the layout process and specifies a value for the attribute “content information”. The content portion description, if any, is used to specify other content portion attributes.

When a content generator is evaluated, the resulting content is laid out and presented in accordance with the layout directive attributes and presentation attributes applicable.

The content architecture class of the basic component together with the attributes of the content portion, if any, determine how to interpret the string expression. The string expression may represent character content, raster graphics content or geometric graphics content, with type of coding, coding attributes and alternative representation as defined. Any character string literals in a content generator shall pertain to the character set and control functions specified for the particular content architecture class.

9.3.3 Relationship attributes

These attributes specify the relationships between objects, between objects and object classes, between objects and content portions, between objects and presentation styles and between styles.

9.3.3.1 Object class

Constituents:

Object descriptions.

Classification:

Non-mandatory.

In the case of a complete generator set of logical object class descriptions this attribute is mandatory for logical object descriptions. In the case of a complete generator set of layout object class descriptions this attribute is mandatory for composite layout object descriptions.

Permissible values:

The identifier of an object class.

Definition:

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

The value of this attribute is the identifier of the corresponding object class description (see 9.3.1.3).

9.3.3.2 Subordinates

Constituents:

Composite object descriptions.

Classification:

Mandatory, unless the exceptional case described below applies, in which case the attribute is non-mandatory.

Permissible values:

A sequence of one or more non-negative integers.

Definition:

This attribute identifies the set of objects immediately subordinate to the object for which this attribute is specified. In the case of subordinate basic objects this attribute identifies their primary descriptions.

The value of this attribute is a sequence of one or more integers. Each integer corresponds to an immediately subordinate object description and consists of the last integer in the identifier of that subordinate object description (see 9.3.1.2). The sequence contains integers corresponding to each immediately subordinate object description and the same integer shall not occur more than once in the sequence.

The order of appearance of the integers in the sequence (not the order of their numeric values) defines the sequential order among the immediately subordinate objects.

In logical object descriptions, the sequential order is interpreted as determining the sequential layout order in which the objects are handled by the layout process. For layout object descriptions, the sequential order is interpreted as determining the imaging order, which is the order in which the immediately subordinate layout objects are overlaid during the imaging process (see 11.1 and 11.2), unless overridden by the attribute “imaging order” (see 9.4.3.1).

Exceptions:

This attribute is non-mandatory in certain documents. These documents are those which have all of the following characteristics:

- The interchange format class used for document interchange is class B (see ITU-T Rec. T.415 | ISO/IEC 8613-5); consequently the only specific structure present is the specific layout structure.
- The only object types present in the document are document layout root, pages and blocks.
- There is no use of object identifiers in attributes.

In documents adhering to these rules any two consecutive objects of the same type in the data stream have the same immediate superior. Thus, under these conditions the semantics of the attribute "subordinates" can be transmitted to the recipient implicitly and the attribute need not be explicitly specified.

The sequential order is defined by the order of appearance in the interchange format, as defined in ITU-T Rec. T.415 | ISO/IEC 8613-5.

NOTE – This exceptional case is provided for compatibility with ITU-T Recommendations.

9.3.3.3 Content portions

Constituents:

Basic component descriptions.

Classification:

Non-mandatory.

Permissible values:

A sequence of one or more non-negative integers.

Definition:

This attribute identifies the set of content portions associated with a component. If more than one content portion is associated with a component then this attribute specifies an ordering among these content portions.

The value of this attribute is a sequence of one or more integers. Each integer corresponds to a content portion of the component concerned and consists of the last integer in the identifier of the content portion description (see 9.9.1). The sequence contains integers corresponding to each content portion of the component concerned and the same integer shall not occur more than once in the sequence.

The order of appearance of the integers in the sequence (not the order of their numeric values) defines the sequential order among the content portions.

The sequential order is interpreted as determining the order in which the content portions are handled by the layout and imaging processes.

The attribute shall be specified for a basic object description unless at least one of the following applies:

a) The basic object description refers to an object class description which specifies content by one of the following means:

   - by having associated generic content portion descriptions;
   - by specifying a content generator;
   - by referencing an object class description in a resource-document which either has associated generic content portion descriptions or which specifies a content generator.

b) The basic object description specifies a content generator; this is possible only in the case of basic logical objects.

Exceptions:

In certain documents this attribute need not be specified for a basic component description even when content portions are associated with the component. These documents are those which have all of the following characteristics:

- The interchange format class used for document interchange is class B (see ITU-T Rec. T.415 | ISO/IEC 8613-5), consequently the only specific structure present is the specific layout structure.
- The only object types present in the document are document layout root, pages and blocks.
- There is no use of the content portion identifier attributes.

In documents adhering to these rules any two consecutive content portions in the data stream are associated with the same basic object. Thus, under these conditions the semantics of the attribute "content portions" can be transmitted to the recipient implicitly and the attribute need not be explicitly specified.

The sequential order is defined by the order of appearance in the interchange format, as defined in ITU-T Rec. T.415 | ISO/IEC 8613-5.

NOTE – This exceptional case is provided for compatibility with ITU-T Recommendations.

9.3.3.4 Resource

Constituents:

Object class descriptions.

Classification:

Non-mandatory.

Permissible values:

A string of characters from the minimum subrepertoire of ISO/IEC 6937.

Definition:

This attribute is used to establish a relationship between an object class description in the interchanged document and an object class description in the resource-document (see 6.3.12 and 7.5.8).

The value of this attribute represents the name of an object class description in the resource-document.

The mapping between these names and the object class identifiers within the resource-document is specified by the attribute "resources" in the document profile of the resource-document.

9.3.3.5 Presentation style

Constituents:

Basic component descriptions.

Classification:

Non-mandatory.

Permissible values:

Either a presentation style identifier or 'null'.

Definition:

This attribute is used to establish a relationship between a basic component description and a presentation style.

If this attribute has the value 'null' then no presentation style is referenced from this basic component description.

9.3.3.6 Alternative

Constituents:

Basic object descriptions.

Classification:

Non-mandatory.

Permissible values:

The identifier of an alternative description of a basic object.

Definition:

For a primary description of an object this attribute refers to the first alternative description, in the order of preference. For an alternative description this attribute refers to the next alternative description, in the order of preference.

NOTE – Thus, the specification of this attribute establishes a chain of alternative descriptions to a primary description of a basic object in the order of decreasing preference.

9.3.3.7 Primary

Constituents:

Basic object descriptions.

Applicable only to alternative descriptions.

Classification:

Mandatory for alternative descriptions.

Permissible values:

The identifier of a basic object.

Definition:

This attribute refers from an alternative description to its primary description.

9.3.3.8 Derived from

Constituents:

Layout styles and presentation styles.

Classification:

Non-mandatory.

Permissible values:

A presentation style identifier or a layout style identifier.

Definition:

This attribute is used to establish a relationship either between a layout style and another layout style or between a presentation style and another presentation style. Attributes and their values from the referenced style are used along with the directly specified attributes. Values of directly specified attributes take precedence over values obtained from referenced styles (see 9.1.2.6).

9.3.4 Content architecture class

Constituents:

Basic component descriptions.

Classification:

Non-mandatory for object class descriptions.
Defaultable for object descriptions.

Permissible values:

An identification of a content architecture class.

Representation:

An ASN.1 object identifier.

For basic layout component descriptions, the value 'formatted raster graphics content architecture' as defined in ITU-T Rec. T.417 | ISO/IEC 8613-7 may have an additional representation in ODIF interchange, see ITU-T Rec. T.415 | ISO/IEC 8613-5. Document Application Profiles may refer to this attribute under the name "content type". In this case, and only in this case, this special representation is to be used.

Default value:

'formatted character content architecture', as defined in ITU-T Rec. T.416 | ISO/IEC 8613-6.

NOTE – In certain cases an implied value is assumed for the document application profile defaults of this attribute, see ITU-T Rec. T.414 | ISO/IEC 8613-4.

Definition:

This attribute specifies the content architecture class of the content associated with the basic component.

This attribute identifies the sets of presentation attributes, control functions and coding attributes which are applicable to the content.

9.3.5 Miscellaneous attributes

9.3.5.1 User-readable comments

Constituents:

Component descriptions and styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for styles.
Defaultable for object descriptions.

Permissible values:

A string of characters from a defined character set. This set consists of SPACE, CARRIAGE RETURN, LINE FEED and a set of graphic characters. The graphic character set is that specified in the document profile attribute "comments character sets". In the absence of the document profile attribute "comments character sets", this set of graphic characters consists of the 73 graphic characters of the minimum subrepertoire of ISO/IEC 6937.

Default value:

empty string.

Definition:

This attribute consists of a sequence of characters that is to be interpreted as comments relevant to the constituent and to any associated content portions. This character sequence is not part of the document content.

This sequence of characters is intended for use in presentation to humans. The attribute has no significance for the reference models of the layout or imaging processes defined in this Specification, nor for any content layout or imaging processes defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613.

9.3.5.2 Application comments

Constituents:

Component descriptions and styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for styles.
Defaultable for object descriptions.

Permissible values:

An octet string.

Default value:

empty string

Definition:

This attribute shall be used for application specific information. The value of this attribute is not part of the document content. It shall be possible to process the document ignoring the value of this attribute.

The attribute has no significance for the reference models of the layout or imaging processes defined in this Specification, nor for any content layout or imaging processes defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613.

9.3.5.3 User-visible name

Constituents:

Component descriptions and styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for styles.
Defaultable for object descriptions.

Permissible values:

A string of characters from a defined character set. This set consists of SPACE, CARRIAGE RETURN, LINE FEED and a set of graphic characters. The graphic character set is that specified in the document profile attribute "comments character sets". In the absence of the document profile attribute "comments character sets", this set of graphic characters consists of the 73 graphic characters of the minimum subrepertoire of ISO 6937-2.

Default value:

empty string.

Definition:

This attribute consists of a sequence of characters that can be used to identify the constituent within the document structure. This character sequence is not part of the document content.

This sequence of characters is intended for use in presentation to humans. The attribute is intended to assist in the editing of documents, for example to enable a user to access an object directly by name. The attribute has no significance for the reference models of the layout or imaging processes defined in this Specification, nor for any content layout or imaging processes defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613.

This attribute is not intended to be used as an alternative to the identification attributes "object identifier", "object class identifier", "layout style identifier" or "presentation style identifier".

For example, in the case of a logical object class description the value of this attribute may be a name which serves to indicate semantics of the object class to a human, such as "chapter", "section", "paragraph" or "footnote", sometimes termed markup. However, such values are not defined by the ITU-T Rec. T.410-Series | ISO/IEC 8613.

9.3.5.4 Bindings

Constituents:

Component descriptions.

Classification:

Non-mandatory for object class descriptions.
Defaultable for object descriptions.

Structure:

A set of pairs of parameters, each pair consisting of:

- a binding name, with a value unique within the set;
- a binding value.

Permissible values:

For the parameter "binding name", a string of characters from the minimum subrepertoire of ISO/IEC 6937.

For the parameter "binding value":

- in the case of object class descriptions or logical object descriptions, an expression, which may be a string expression, a numeric expression, or an object identifier expression, as defined in 9.1.3;
- in the case of layout object descriptions, a string literal, a numeric literal, or an object identifier.

Default values:

Each pair of parameters is independently defaultable. For each possible value of the parameter "binding name", the default is that no binding is specified.

Definition:

This attribute specifies a means for determining attribute values. The names specified by the parameter "binding name" are assigned by the application.

The use of this attribute is restricted to relate ultimately to the value of an attribute.

In the ITU-T Rec. T.410-Series | ISO/IEC 8613 this attribute shall be used only in the evaluation of the content specified by the attribute "content generator".

9.3.5.5 Default value lists

Constituents:

Composite component descriptions.

Classification:

Non-mandatory.

Structure:

This attribute consists of a set of one or more lists of attributes.

Permissible values:

One or more lists of attributes, each list applicable to one particular object type.

Within such a set of lists, there shall be zero or one list that pertains to any particular object type.

Definition:

This attribute specifies default attribute values for subordinate object descriptions.

Table 1 lists the attributes that may be included in a list for each object type.

In the case of a page, if the list applies to a composite page, the attributes "content architecture class", "presentation style", and presentation attributes are not applicable.

When a list applies to a basic object description, then that list may contain one or more presentation styles or sets of presentation attributes, each corresponding to a different content architecture class.

Table 1 – Defaultable attributes that can be specified in default value lists

Object type	Defaultable attributes that can be specified
Document Layout Root Document Logical Root Page Set	(No attributes can be specified)
Composite or Basic Page	presentation style content architecture class dimensions transparency colour page position medium type presentation attributes sealed colour of layout object object colour table content background colour content foreground colour content colour table
Frame	position dimensions border layout path permitted categories transparency colour sealed colour of layout object object colour table
Block	presentation style content architecture class position dimensions border transparency colour presentation attributes sealed colour of layout object object colour table content background colour content foreground colour content colour table
Composite Logical Object	protection layout style sealed
Basic Logical Object	presentation style content architecture class protection layout style sealed

9.3.6 Security attributes

9.3.6.1 Enciphered

Constituents:

Basic component descriptions and composite object descriptions.

Classification:

Non-mandatory.

Structure:

The parameters, enciphered subordinates and protected part identifier. The parameter "enciphered subordinates" either has a value or is structured into a sub-parameter "partial".

Permissible values:

For the parameter "enciphered subordinates", the values 'none' or 'all', or a sub-parameter "partial" whose value is a sequence of one or more non-negative integers. The sub-parameter "partial" is applicable only to a basic object description in a document of the formatted processable form.

For the parameter "protected part identifier", which is optional, a sequence of two non-negative integers representing the identifier of a pre- or a post-enciphered document body part description (see 9.10.1).

Definition:

This attribute specifies whether the component descriptions and the content portions that are subordinate to this component description are enciphered or not, and if they are enciphered where the enciphered parts are to be found.

This attribute specifies the current encipherment status. The relevant instance of this attribute shall be updated at every encipherment and every decipherment of a document body part.

In the case of a composite object description, if the parameter "enciphered subordinates" has the value 'all', then all subordinate object descriptions and all content portions associated with any of them are enciphered.

In the case of a composite object description, if this attribute is not specified or if the parameter "enciphered subordinates" has the value 'none', then none of its immediate subordinate object descriptions is enciphered.

In the case of a basic component description, if the parameter "enciphered subordinates" has the value 'all', then all content portions associated with the component description are enciphered.

In the case of a basic component description, if this attribute is not specified or if the parameter "enciphered subordinates" has the value 'none', then none of the content portions associated with the component description is enciphered.

In the case of a basic object description in a formatted processable form document, if the parameter "enciphered subordinates" has the sub-parameter "partial", then this sub-parameter is in the form of a sequence of content portion identifiers. In this case only the specified content portions are enciphered. Each content portion identifier is represented by its last integer.

If the parameter "enciphered subordinates" has a value different from 'none', the parameter "protected part identifier" shall specify the identifier of either the pre-enciphered document body part or the post-enciphered document body part where the enciphered part is to be found.

The layout process shall ignore the subtree of the logical structure for which the parameter "enciphered subordinates" has a value different from 'none'. The imaging process shall ignore the subtree of the layout structure for which the parameter "enciphered subordinates" has a value different from 'none'.

9.3.6.2 Sealed

Constituents:

Component descriptions and styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for styles.
Defaultable for object descriptions.

Structure:

The parameter "sealed status" and an optional parameter "seal identifiers".

Permissible values:

sealed status: 'no' or 'yes'.

seal identifiers: a list of integers identifying the seals in which this component or style is incorporated.

Default value:

Only the parameter “sealed status” has a default value.

The default value is 'no'.

Definition:

This attribute specifies if this component description or style is incorporated in a seal (see 6.6.3).

This attribute specifies the current sealing status. The relevant instance of this attribute shall be updated at every sealing or deletion of a seal of a document body part.

The value 'no' of the parameter “sealed status” implies that this component description or style is not incorporated in any seal.

If, for a composite object description, the parameter “sealed status” has the value 'yes', then it and all its subordinates and all content portions associated with them are included in one or more seals.

If, for a composite object class description or style, the parameter “sealed status” has the value 'yes', then it is included in one or more seals.

If for a basic component description the parameter “sealed status” has the value 'yes' then it and all content portions associated with it are included in one or more seals.

The parameter “seal identifiers” specifies the seals in which the specified part of the document body is included.

9.4 Layout attributes

The attributes defined in this sub-clause are applicable to layout components and are not applicable to logical components.

9.4.1 Property attributes

These attributes specify the characteristics that are applicable to layout components.

9.4.1.1 Position

Constituents:

Frame and block component descriptions.

Classification:

Non-mandatory for object class descriptions.
Defaultable for object descriptions.

Structure:

Either, the parameter “fixed position”, which has two sub-parameters:

horizontal position; and

vertical position,

or the parameter “variable position”, which has four optional sub-parameters:

offset,

separation,

alignment; and

fill order.

The sub-parameter “offset” is structured into four optional sub-sub-parameters:

leading offset;

trailing offset;

left-hand offset;

right-hand offset.

The sub-parameter “separation” is structured into three optional sub-sub-parameters:

leading edge;

trailing edge;

centre separation.

Permissible values:

For the parameter “fixed position”:

– horizontal position:	a non-negative integer;
– vertical position:	a non-negative integer.

For the parameter “variable position”:

- For the sub-parameter “offset”, zero or more of “leading offset”, “trailing offset”, “left-hand offset”, “right-hand offset” can be specified in any instance of this sub-parameter. For each the permissible value is a non-negative integer.
- For the sub-parameter “separation”, zero or more of “leading edge”, “trailing edge”, “centre separation” can be specified in any instance of this sub-parameter. For each the permissible value is a non-negative integer.
- For the sub-parameter “alignment”, the permissible values are 'right-hand aligned', 'centred', 'left-hand aligned'.
- For the sub-parameter “fill order”, the permissible values are 'normal order', 'reverse order'.

Default values:

If no value is specified, the default value is the parameter “fixed position” with the sub-parameters:

– horizontal position:	0
– vertical position:	0

Definition:

This attribute specifies the position of the object relative to the object at the next higher level in the hierarchical structure (i.e. the immediately superior page or frame).

Two cases are to be considered, that of fixed position and that of variable position.

In the case of fixed position, the sub-parameters “horizontal position” and “vertical position” correspond to the horizontal and the vertical distances from the reference point of the immediately superior layout object to the reference point of the layout object to which this attribute applies. The position specified shall be within the immediately superior layout object. If the immediately superior layout object has a border then the position is further constrained not to fall within that border.

The sub-parameter “horizontal position” specifies the horizontal distance, the sub-parameter “vertical position” specifies the vertical distance. Each of these sub-parameters consists of a non-negative integer, representing the distance concerned in scaled measurement units.

The case of variable position, may only be specified for frame class descriptions that are referred to in construction expressions (see 9.3.2.1) only from other frame class descriptions. Consequently, block component descriptions, frame descriptions and frame class descriptions referred to in construction expressions specified for page class descriptions may only specify fixed position.

In the case of variable position, zero or more of the sub-parameters "offset", "separation" "alignment" and "fill order" are specified, as follows:

a) *Offset*

This sub-parameter constrains the area within the immediately superior layout object in which the frame can be placed.

The sub-parameter specifies minimum amounts of offset between the boundary of the frame and the boundary of the immediately superior layout object.

The sub-parameter is structured into four sub-sub-parameters, "leading offset", "trailing offset", "left-hand offset", "right-hand offset". These specify the minimum distance between the corresponding edge of the frame and the leading, trailing, left-hand and right-hand edge of the immediately superior layout object, respectively.

(The names of the edges are as defined in 7.3.3.)

For each edge, the sub-sub-parameter specifies the amount of offset for that edge in scaled measurement units.

If the parameter "variable position" does not specify a value for this sub-parameter, or for any of its sub-sub-parameters, then the value zero is assumed for each sub-sub-parameter for which a value is not specified.

b) *Separation*

This sub-parameter specifies minimum amounts of separation between this frame and the nearest adjacent frame immediately subordinate to the same immediate superior layout object. The sub-parameter is structured into three sub-sub-parameters, "leading edge", "trailing edge", "centre separation".

The sub-sub-parameter "leading edge" specifies the minimum separation from the leading edge of the frame and the trailing edge of the next frame laid out in the same fill order.

The sub-sub-parameter "trailing edge" specifies the minimum separation from the trailing edge of the frame and the leading edge of the preceding frame laid out in the same fill order.

Thus, a constraint on the separation of two adjacent frames which both have the same fill order is that the separation shall be equal to, or greater than, the greater of the value of the sub-sub-parameter "leading edge" for the first of the frames in the direction of layout path and the value of the sub-sub-parameter "trailing edge" for the second of the frames.

The sub-sub-parameter "centre separation" specifies the minimum separation between two frames that are laid out with different values for the sub-parameter "fill order".

Thus, a constraint on the separation of two adjacent frames which have different fill orders is that the separation shall be equal to, or greater than, the greater of the values of the sub-sub-parameter "centre separation" specified for the two frames.

For each edge, the sub-sub-parameter specifies the amount of separation for that edge in scaled measurement units.

(The names of the edges are as defined in 7.3.3.)

If the parameter "variable position" does not specify a value for this sub-parameter, or for any of its sub-sub-parameters, then the value zero is assumed for each sub-sub-parameter for which a value is not specified.

c) *Alignment*

This sub-parameter specifies the alignment of the frame within the area available for positioning the frame inside the immediately superior layout object. The alignment is in the direction orthogonal to that specified by the attribute "layout path" of the immediately superior layout object.

This sub-parameter takes one of three values, 'right-hand aligned', 'centred', 'left-hand aligned'.

Subject to satisfying constraints on placement specified by the sub-parameter "offset" of the frame and the attribute "border" of the immediately superior layout object, the values of alignment are defined as follows:

1) if the value is 'right-hand aligned' then this frame is to be positioned as close as possible to the right-hand edge of the immediately superior layout object;

2) if the value is 'centred' then this frame is to be centred, in the direction orthogonal to the layout path of the immediately superior layout object, within the area of the immediately superior layout object which is available for positioning the frame;

3) if the value is 'left-hand aligned' then this frame is to be positioned as close as possible to the left-hand edge of the immediately superior layout object.

(The names of the edges are as defined in 7.3.3.)

It is possible to align independently each of a set of frames that have a common immediately superior layout object.

If the parameter "variable position" does not specify a value for this sub-parameter then the value 'right-hand aligned' is assumed.

d) *Fill order*

The sub-parameter "fill order" specifies how a frame is to be positioned in its immediately superior layout object relative to the direction of layout path of that object.

The sub-parameter "fill order" takes one of two values, 'normal order' and 'reverse order', defined as follows:

1) If the value is 'normal order' then this frame is grouped together with any other frames having the same immediately superior layout object and specifying this value. These frames are positioned after each other in the direction of the layout path of their common immediately superior layout object. The frames in the group are positioned in their sequential layout order, starting at the distance specified by the trailing offset of the first of these frames from the trailing edge of the immediately superior layout object.

2) If the value is 'reverse order' then this frame is grouped together with any other frames having the same immediately superior layout object and specifying this value. These frames are positioned after each other in the direction of the layout path of their common immediately superior layout object. The frames in the group are positioned in their sequential layout order, ending at the distance specified by the leading offset of the last of these frames from the leading edge of the immediately superior layout object.

The sub-parameter is applied subject to constraints specified by the parameters "offset" and "separation".

(For the definition of leading and trailing edges see 7.3.3.)

If the parameter "variable position" does not specify a value for this sub-parameter then the value 'normal order' is assumed.

This attribute is subject to the precedence rule that satisfying constraints on the positioning of other frames having the same immediately superior layout object and which contain any content laid out earlier in sequential logical order than any content contained in this frame, has precedence over satisfying constraints on the positioning of this frame.

9.4.1.2 Dimensions

Constituents:

Page, frame or block component descriptions.

Classification:

Non-mandatory for object class descriptions.
Defaultable for object descriptions.

Structure:

One of two cases applies:

- *Case 1* – The attribute consists of two parameters: "horizontal dimension", "vertical dimension".

  The parameter "vertical dimension" consists of one of the sub-parameters: "fixed dimension" or "variable page height".

  The parameter "horizontal dimension" is represented by a sub-parameter "fixed dimension".

- *Case 2* – The attribute consists of two parameters: "horizontal dimension", "vertical dimension".

  Each parameter includes one of the four sub-parameters: "fixed dimension", "Rule A", "Rule B" and "maximum size".

  The sub-parameters "Rule A" and "Rule B" include two optional sub-sub-parameters "minimum dimension", "maximum dimension".

Permissible values:

- *Case 1*:
  - horizontal dimension:
    - fixed dimension: positive integer;
  - vertical dimension: one of two sub-parameters:
    - fixed dimension: positive integer;
    - variable page height: any integer.
- *Case 2*:
  - horizontal dimension: one of four sub-parameters:
  - vertical dimension: one of four sub-parameters:
    - fixed dimension: a positive integer;
    - Rule A: two optional sub-sub-parameters:
      - minimum dimension: a positive integer,
      - maximum dimension: a positive integer;
    - Rule B: two optional sub-sub-parameters:
      - minimum dimension: a positive integer,
      - maximum dimension: a positive integer;
    - maximum size: applies.

Default values:

Each of the parameters is independently defaultable.

For a frame or block: horizontal dimension, vertical dimension both with the sub-parameter "fixed dimension", with a value that is the maximum size that can be achieved for the position within the area of the immediately superior object.

For a page: horizontal dimension, vertical dimension both with the sub-parameter "fixed dimension", with a value that is the assured reproduction area for ISO A4 (see 11.3.3)

Definition:

One of two cases applies:

- *Case 1* shall apply to either objects or object classes;
- *Case 2* shall apply only to object classes.

In either case, this attribute consists of a pair of parameters, corresponding to the dimensions in the horizontal and vertical directions of the component in scaled measurement units. The parameter "horizontal dimension" specifies the dimension in the horizontal direction, the parameter "vertical dimension" specifies the dimension in the vertical direction. (For the definition of horizontal direction and vertical direction, see 7.3.2.1.)

Case 1:

The parameter "horizontal dimension" is represented by a sub-parameter "fixed dimension", the parameter "vertical dimension" is either represented by a sub-parameter "fixed dimension" or by a sub-parameter "variable page height", defined as follows:

- fixed dimension: specifies the dimension concerned in scaled measurement units;
- variable page height: specifies that the vertical dimension is undetermined.

Case 2:

Each parameter includes one of the four sub-parameters, "fixed dimension", "Rule A", "Rule B" and "maximum size", defined as follows:

a) *Fixed dimension* – Specifies the dimension concerned in scaled measurement units.

b) *Rule A* – Specifies that the dimension concerned is to be the minimum size necessary for the frame to contain that immediately subordinate frame or block which has the associated content that is earliest in sequential logical order. The dimension acts as a constraint on the dimensions in the same direction of all other immediately subordinate frames or blocks.

c) *Rule B* – Specifies that the dimension concerned is to be the minimum size necessary for the frame to contain all the immediately subordinate frames or blocks.

d) *Maximum size* – Specifies that the same constraints apply as for a dimension specified as "Rule B", with the additional constraint that it is intended to obtain the maximum dimension possible in the dimension specified as "maximum size". This additional constraint has less precedence than the constraints imposed by either superior frames or frames having the same immediately superior layout object. Competition between constraints specified by variably positioned frames having the same immediately superior layout object that specify "maximum size" for their dimension in the direction of the layout path of the immediately superior frame is resolved by evenly expanding all these frames by the same amount.

By the use of the sub-parameters in Case 2, each dimension of a frame can have either a fixed size or a variable size.

The sub-parameters "Rule A" and "Rule B" may be specified only for frame class descriptions that are referred to in construction expressions (see 9.3.2.1) only from other frame class descriptions. Consequently, page component descriptions, block component descriptions, frame descriptions and frame class descriptions referred to in construction expressions specified by page class descriptions shall not specify these rules. Further, the sub-parameter "Rule A" may be specified only for the dimension in same direction as the layout path of the immediately superior frame.

In the case of the sub-parameters "Rule A" "Rule B", two further sub-sub-parameters may optionally be specified:

- minimum dimension;
- maximum dimension.

These sub-sub-parameters specify a constraint on the dimensions determined by the rule. If the value determined by the rule would otherwise be smaller than the value of the sub-sub-parameter "minimum dimension" then the value will be that specified by that sub-sub-parameter. If the value determined by the rule would otherwise be larger than the value of the sub-sub-parameter "maximum dimension" then the value will be that specified by that sub-sub-parameter.

The values of the sub-sub-parameters "minimum dimensions" and maximum dimension" are specified in scaled measurement units.

In all cases the dimensions for frame and block components are constrained to be no greater than the maximum size which can be contained within the immediately superior frame or page (given the position of the frame or block and the direction of the layout path). If the immediately superior layout object is a frame which has a border then the dimensions are further constrained such that no part of the frame or block falls within the border. Note that this maximum size is the default value for the attribute.

NOTE – The sub-parameter "variable page height" is provided for compatibility with existing ITU-T Recommendations. The value of this sub-parameter has no significance.

9.4.1.3 Border

Constituents:

Frame and block component descriptions, presentation styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for presentation styles.
Defaultable for object descriptions.

Structure:

Four optional parameters: left-hand-edge, right-hand-edge, trailing-edge and leading-edge. Zero or any combination of these may be specified.

Each parameter either has the value 'null' or consists of zero or more of the optional sub-parameters "border line width", "border line type", "border line colour", "border freespace width". The sub-parameter "border line colour", if specified, is a choice between two sub-sub-parameters: "implementation-defined" and "colour-expression". The sub-sub-parameter "implementation-defined" is not substructured. The sub-sub-parameter "colour-expression" is structured as defined for the colour expressions in 9.1.4.1.

Permissible values:

For each of the four parameters

- null;

or, a combination of

– border line width:	a non-negative integer;
– border line type:	one of the values 'solid', 'dashed', 'dot', 'dash-dot', 'dash-dot-dot', 'invisible';
– border freespace width:	a non-negative integer;
– border line colour:	
– implementation-defined:	implementation defined;
– colour expression:	a colour expression as defined in 9.1.4.1.

Default values:

The four parameters and each of the sub-parameters are independently defaultable.

For each of the four parameters

– border line width:	0;
– border line type:	solid;
– border freespace width:	0;
– border line colour:	
– implementation defined:	implementation-defined.

NOTE – The default value of "border line colour" was chosen for compatibility with the previous edition of this Specification.

Definition:

This attribute specifies a border, consisting of a border line and border freespace, for the edges of a frame or block component (see 7.3.5). Each of the four parameters determines that the corresponding block or frame edge is to be highlighted by the presence of a border.

The border line is described by specifying its width in scaled measurement units and its line type. The border freespace is described by specifying its width in scaled measurement units.

The sub-parameter "border line colour" specifies the colour of the border line. It either has the value 'implementation-defined' (a value that is distinguishable from the colour visible in the layout object) or is given in terms of a colour expression as defined in 9.1.4.1. In the case of an indexed colour expression the relevant colour table is specified in the attribute "object colour table" applicable to the object.

In the case of a block, the border is outside the edges of the block and the border freespace provides a region which surrounds the block between the block boundary and the border line.

In the case of a frame, the border is inside the edges of the frame and the border freespace provides a region which is within the frame between the frame boundary and the border line.

If the parameter for a particular edge has the value 'null', then no border shall be drawn for that edge.

In the case of the attribute being specified for an object class, if any of the parameters or sub-parameters is not specified, then for the sub-parameters "border line width", "border freespace width" the value zero is assumed. For the sub-parameter "border line type" the value 'solid' is assumed.

9.4.2 Formatting attributes

These attributes specify the information applicable to the formatting of the document.

9.4.2.1 Balance

Constituents:

Composite layout component descriptions without immediately subordinate blocks.

Classification:

Non-mandatory for layout object class descriptions.
Defaultable for layout object descriptions.

Permissible values:

In the case of a layout object description, either 'null' or a sequence of two or more layout object identifiers.

In the case of a layout object class description, either 'null' or a sequence of two or more layout object class identifiers.

Default value:

null.

Definition:

This attribute specifies that the leading edges of a set of immediately subordinate layout objects shall, as far as possible, be aligned along a line orthogonal to the direction of the layout path.

In the case of a layout object description, the value of this attribute is either 'null' or a sequence of two or more identifiers of immediately subordinate object descriptions.

In the case of a layout object class description, the value of this attribute is either 'null' or a sequence of two or more identifiers of layout object class descriptions for composite layout objects. The layout object class description is also required to specify the attribute "generator for subordinates" which shall be able to generate a sequence of object class identifiers which correspond one-for-one and in sequence to the sequence of object class identifiers specified by this attribute. If this requirement is not satisfied then the attribute balance shall be ignored.

All object descriptions or object class descriptions identified by this attribute shall have the same value for layout path (see 9.4.2.2) and the same stream categories (see 9.7.1.2) and the same layout categories (see 9.4.2.6, 10.3). If the values for layout path are not the same or the same set of stream categories and layout categories are not specified then the attribute "balance" shall be ignored.

The value 'null' indicates that no constraints on the subordinate layout objects are specified by this instance of this attribute.

Further detail on this attribute is included in the definition of the reference model of the layout process (see clause 10, in particular 10.4.6).

9.4.2.2 Layout path

Constituents:

Frame component descriptions.

Classification:

Non-mandatory for object class descriptions.
Defaultable for object descriptions.

Permissible values: A set of data elements defined for the attribute:

0°,

90°,

180°,

270°.

Representation:

In the interchange format, 0°, 90°, 180°, 270° are represented by the character strings d0, d90, d180, d270, respectively.

Default value:

270°.

Definition:

In the case of lowest-level frames this attribute specifies the direction of progression of the allocation of any immediately subordinate blocks during the layout process, relative to the horizontal direction.

In the case of higher-level frames this attribute specifies the direction of progression of the allocation of any immediately subordinate frames with variable positions during the layout process, relative to the horizontal direction.

This attribute has no meaning in the case of immediately subordinate frames or blocks with fixed positions.

For a frame for which the object class defines variable dimensions, the layout path influences the determination of these dimensions, as described for the attribute "dimensions" (see 9.4.1.2).

9.4.2.3 Layout stream categories

Constituents:

Layout component descriptions, except the document layout root.

Classification:

Non-mandatory for object class descriptions.
Defaultable for object descriptions.

Permissible values:

Either 'null' or one or more strings of characters from the minimum subrepertoire of ISO/IEC 6937, each being the identifier of a stream root category. The implicit stream root category is represented by an empty string.

Default value:

Implicit stream root category.

Definition:

This attribute is applicable for an object only if no superior object has a non-null value for the attribute. If any superior object has a non-null value for the attribute then this instance of the attribute is ignored.

This attribute specifies the stream root categories applicable to the layout object and to all of its subordinate objects.

This attribute constrains the stream categories permitted for logical objects the content of which is to be laid out within the layout object and its subordinates.

A stream root category has an identifier which may be associated with layout component descriptions, and which applies to all subordinate layout component descriptions. One or more stream root categories may apply to a layout component.

The content associated with a logical object is constrained to be placed within layout objects to which a stream root category applies which has the same identifier as the stream root category that applies to the logical object. This provides for the layout process to consist of a number of hierarchically related layout streams (see 10.3).

The value of this attribute is a set of identifiers of stream root categories.

If the attribute value is 'null', the content associated with logical objects of any stream root category (including the implicit stream root category) is permitted to be placed in the layout object.

9.4.2.4 Layout stream sub-categories

Constituents:

Layout component descriptions, except the document layout root.

Classification:

Non-mandatory for object class descriptions.
Defaultable for object descriptions.

Permissible values:

Either 'null' or one or more strings of characters from the minimum subrepertoire of ISO/IEC 6937, each being the identifier of a stream sub-category.

Default value:

null.

Definition:

This attribute is applicable for an object only if a superior object has a non-null value for the attribute "layout stream categories". If no superior object has a non-null value for the attribute "layout stream categories", then this instance of this attribute is ignored.

This attribute specifies the stream sub-categories applicable to the layout object and to all of its subordinate objects.

This attribute constrains the stream categories permitted for logical objects the content of which is to be laid out within the layout object and its subordinates. Together with the stream root categories of the superior layout object to which the attribute "layout stream categories" applies and the stream sub-categories of intermediate superior layout objects to which the attribute "layout stream sub-categories" applies (if any) it forms a set of full hierarchic identifiers of the stream categories that apply to the layout object.

A stream sub-category has an identifier which may be associated with layout component descriptions, and which applies to all subordinate layout component descriptions, in order to specify and restrict the layout objects into which the content associated with logical objects may be placed. One or more stream sub-categories may apply to a layout component.

The content associated with a logical object is constrained to be placed within layout objects to which a stream category applies which has the same identifier as the stream category that applies to the logical object. This provides for the layout process to consist of a number of hierarchically related layout streams (see 10.3).

The value of this attribute is a set of identifiers of stream sub-categories.

If the attribute value is 'null', the effect is as if the attribute did not apply to the logical component.

9.4.2.5 Logical source

Constituents:

Frame object class descriptions.

Classification:

Non-mandatory.

Permissible values:

An object class identifier for a logical object class description.

Definition:

This attribute is specified for a layout object class if the content associated with each of the layout objects of that class is to be supplied by a logical object class, for example, the content associated with a header or footer frame on a page.

The attribute identifies the logical object class description concerned.

The effect of the attribute is that an instance of an object of the specified logical object class, and all its subordinates, if any, is created automatically whenever an instance of an object of the layout object class is generated during the layout process.

If the logical object class description specifies the attribute "generator for subordinates", the construction expression it contains is evaluated, causing the creation of one or more subordinate logical objects. This step is then repeated for the logical object class descriptions corresponding to these subordinate objects. If any of the logical object class descriptions specifies the attribute "content generator" then the expression in this attribute is evaluated.

The attribute "generator for subordinates" specified by the logical object class description, if any, shall contain only construction terms which consist of required construction factors, or sequence constructions which use only required construction factors. The same rule applies to logical object class descriptions for all subordinate composite objects.

The content associated with the created logical object(s) is then laid out entirely within the layout object, as if the logical object class identified had specified the attribute "layout object class" referring to the layout object class concerned.

The logical object that is automatically created and its subordinates, if any, are not added to the specific logical structure and are not interchanged as a part of the document. However, the layout object and its subordinates and content are added to the specific layout structure.

The content portion descriptions added to the specific layout structure shall contain the attribute "content identifier – layout" and shall not contain the attribute "content identifier – logical".

9.4.2.6 Permitted categories

Constituents:

Lowest level frame component descriptions.

Classification:

Non-mandatory for object class descriptions.
Defaultable for object descriptions.

Permissible values:

Either 'null' or one or more strings of characters from the minimum subrepertoire of ISO/IEC 6937, each being the identifier of a layout category. The implicit layout category is represented by an empty string.

Default value:

Implicit layout category.

Definition:

This attribute specifies the layout categories of the layout object. The attribute constrains the layout categories permitted for logical objects the content of which is to be laid out within the frame.

A layout category has an identifier which may be associated with lowest level frame component descriptions and with basic logical component descriptions in order to specify and restrict the layout objects into which the content associated with basic logical objects may be placed.

One or more layout categories may apply to a frame component. The content associated with a basic logical object is constrained to be placed within frame(s) that specify the same layout category as that which applies to the logical object. This provides for the layout process to consist of a number of layout streams (see 10.3).

The value of this attribute is the set of identifiers of the layout categories permitted.

This attribute is significant only for lowest level frames. If the attribute is specified for a frame which has frame(s) as subordinates then the attribute is ignored, i.e., permitted layout categories are not cumulative.

If the attribute value is 'null', the content associated with logical objects of any layout category (including the implicit layout category) is permitted to be placed in the frame.

NOTE – Layout streams require layout categories to match and also require stream categories to match (see 10.3). Layout categories and stream categories are provided in order to provide compatibility with the previous edition of this Specification.

9.4.3 Imaging attributes

These attributes specify the information applicable to imaging the document.

9.4.3.1 Imaging order

Constituents:

Object descriptions for composite pages or frames.

Classification:

Non-mandatory.

Permissible values:

A sequence of one or more non-negative integers.

Definition:

This attribute specifies the precedence for imaging the immediately subordinate layout objects.

The value of this attribute is a sequence of one or more integers. Each integer corresponds to an immediately subordinate object description and consists of the last integer in the identifier of that subordinate object description (see 9.3.1.2). The sequence contains integers corresponding to each immediately subordinate object description and the same integer shall not occur more than once in the sequence.

The order of appearance of the integers in the sequence (not the order of their numeric values) defines the imaging order among the immediately subordinate objects (see 11.1).

The imaging order determines how the image of the document is resolved for displaying on a presentation surface. In the definition of the reference imaging process it determines the order in which the immediately subordinate layout objects are overlaid during the imaging process. The use in combination of the attributes "imaging order", "transparency" and "colour" is described in 11.2.

If a value is not specified for this attribute, the imaging order is the same as the sequential layout order, as specified by the attribute "subordinates" (see 9.3.3.2).

9.4.3.2 Transparency

Constituents:

Page, frame or block component descriptions, presentation styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for presentation styles.
Defaultable for object descriptions.

Permissible values:

A set of data elements defined for the attribute:

transparent,

opaque.

Default value:

transparent.

Definition:

This attribute defines the transparency of a page, frame or block.

Transparency is in effect only when blocks and/or frames actually intersect. When two or more frames and/or blocks intersect, the effect of combination is determined from the imaging order, as described in 11.1. The use in combination of the attributes "imaging order", "transparency" and "colour" is described in 11.2.

9.4.3.3 Colour

Constituents:

Page, frame and block component descriptions, presentation styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for presentation styles.
Defaultable for object descriptions.

Permissible values:

A set of data elements defined for the attribute:

colour of media,

coloured

Default value:

colour of media

Definition:

This attribute indicates whether the colour of a block, frame or page is specified in the document or is taken from the media onto which the document is imaged.

If the value is 'coloured', the colour of the page, frame or block is determined by the attribute “colour of layout object”. The value 'coloured' shall be specified only for an attribute that applies to an object to which the attribute “transparency” does not apply with the value 'opaque'.

When two or more frames and/or blocks intersect, the effect of combination is determined from the imaging order, as described in 11.1. The use in combination of the attributes “imaging order”, “transparency” and “colour” is described in 1.2.

NOTES

1 The value 'colourless' in the previous edition of this Specification has been renamed to 'colour of media'.

2 The value 'white' in the previous edition of this Specification has been renamed to 'coloured'. In the attribute “colour of layout object” the default is the colour white, for backwards compatibility with the previous edition of this Specification.

9.4.3.4 Page position

Constituents:

Page component descriptions.

Classification:

Non-mandatory for object class descriptions.
Defaultable for object descriptions.

Structure:

Two parameters “horizontal position” and “vertical position”.

Permissible values:

For each parameter, a non-negative integer.

Default value:

Such that edge losses are minimised.

Definition:

This attribute specifies the position of the layout object page within a nominal page (see 11.3).

The value of this attribute is a pair of parameters that specify the horizontal and vertical distances from the top left corner of the nominal page to the reference point of the layout object page, in scaled measurement units.

The use of this attribute is further described in 11.3.

9.4.3.5 Medium type

Constituents:

Page component descriptions.

Classification:

Non-mandatory for object class descriptions.
Defaultable for object descriptions.

Structure:

Three parameters, “nominal page size”, “side of sheet”, and “colour of medium”. The parameter “nominal page size” consists of two sub-parameters: horizontal dimension, vertical dimension. The parameter “colour of medium” is structured as one of two sub-parameters: “specified colour” and “unspecified colour”. The sub-parameter “specified colour” is structured as defined for colour expressions in 9.1.4.1. The sub-parameter “unspecified colour” is not substructured.

Permissible values:

For the parameter “nominal page size”:

horizontal dimension:	a positive integer
vertical dimension:	a positive integer

For the parameter “side of sheet”, a set of data elements defined for the parameter:	recto, verso, unspecified.

For the parameter “colour of medium”:

specified colour:	a colour expression as defined in 9.1.4.1.
unspecified colour:	unspecified.

Default values:

The three parameters are independently defaultable.

The default values are:

– nominal page size:	dimensions for ISO A4 (see 11.3.3);
– side of sheet:	unspecified.
– colour of medium:	
unspecified colour:	unspecified.

NOTES

1 If the sub-parameter “specified colour” is specified then defaulting rules apply to the colour expression, as defined in 9.1.4.1.

2 The default value for “colour of medium” was chosen for compatibility with the previous edition of this Specification.

Definition:

This attribute defines the type of presentation media that is to be used for imaging the page.

The parameter “nominal page size” identifies the particular nominal page size that is to be used. This parameter specifies the horizontal and vertical dimensions of the nominal page, in scaled measurement units. The sub-parameter “horizontal dimension” specifies the dimension in the horizontal direction, the sub-parameter “vertical dimension” specifies the dimension in the vertical direction. (For the definition of horizontal direction and vertical direction, see 7.3.2.1.)

The parameter “side of sheet” indicates the side of a sheet on which the page is to be imaged or indicates that this is unspecified. (For the definition of 'recto' and 'verso', see 11.3.4.)

The parameter “colour of medium” specifies the intended colour of the presentation medium. The specification of the sub-parameter “unspecified colour” indicates that an arbitrary colour would satisfy the intention of the originator. If the sub-parameter “specified colour” is specified the intended colour of the medium is specified as a colour expression, as defined in 9.1.4.1. In the case of an indexed colour expression the relevant colour table is specified in the attribute “object colour table” applicable to the page.

The use of this attribute is described further in clause 11.

9.4.3.6 Colour of layout object

Constituents:

Page, frame or block component descriptions, presentation styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for presentation styles.
Defaultable for object descriptions.

Structure:

The attribute has the structure of a colour expression, as defined in 9.1.4.1.

Permissible values:

The permissible values are as defined in 9.1.4.1.

Default values:

The default values are as defined in 9.1.4.1.

Definition:

This attribute defines the colour of a page, frame or block.

The value of this attribute is ignored when applied to an object to which a “transparency” value other than 'opaque' or a “colour” value other than 'coloured' applies.

The value of this attribute is either a direct colour expression or an indexed colour expression. In the case of an indexed colour expression the relevant colour table is specified in the attribute “object colour table” applicable to the object.

When two or more frames and/or blocks intersect, the effect of combination is determined from the imaging order, as described in 11.1 and the intersection principles described in 11.2.

NOTE – The default colour is defined as white to allow compatibility with the previous edition of this Specification for the attributes “transparency” and “colour”.

9.4.3.7 Object colour table

Constituents:

Page, frame and block component descriptions, presentation styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for presentation styles.
Defaultable for object descriptions.

Structure:

This attribute has the structure of a colour table, as defined in 9.1.4.2.

Permissible values:

The permissible values are as defined in 9.1.4.2.

Default value:

The default values are as defined in 9.1.4.2.

Definition:

The value of this attribute defines a colour table, and can be referenced to specify the colour for layout components to which the attribute applies. Colour tables are further explained in 9.1.4.2

NOTE – This colour table may be referenced from indexed colour expressions (see 9.1.4.1) by the attributes “colour of layout object”, “border”, and “medium type” applying to the object.

9.4.3.8 Content background colour

Constituents:

Basic page component descriptions, block component descriptions, presentation styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for presentation styles.
Defaultable for object descriptions.

Structure:

Either the parameter “content background transparency” or the parameter “colour expression”. “The parameter “content background transparency” is not substructured. The parameter “colour expression” is structured as defined for colour expressions in 9.1.4.1.

Permissible values:

– content background transparency:	content background transparency
– colour expression:	as defined in 9.1.4.1.

Default values:

content background transparency:	content background transparency.

If the parameter "colour expression" is specified then defaulting rules apply to each of the subparameters of the parameter "colour expression" as defined in 9.1.4.1.

Definition:

This attribute specifies the initial value of the background colour of the content of the object to which the attribute applies.

If the parameter "content background transparency" is specified, the content background is transparent, i.e. the colour and transparency of the underlying block also applies to the content background (see 11.2). Otherwise, the content background occludes the underlying block.

If the parameter "colour expression" is specified, the value is a colour expression as defined in 9.1.4.1. In the case of an indexed colour expression the relevant colour table is specified in the attribute "content colour table" applicable to the object.

NOTE – It is suggested that when applying both "content background colour" and "content foreground colour" to the same object, the same values be used for "colour space id," "colour tolerance" and "content colour table" in each case.

9.4.3.9 Content foreground colour

Constituents:

Basic page component descriptions, block component descriptions, presentation styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for presentation styles.
Defaultable for object descriptions.

Structure:

A choice between three parameters: implementation-defined, content foreground transparency and colour expression. The parameters "implementation-defined" and "content foreground transparency" are not substructured.

The parameter "colour expression" is structured as defined for colour expressions in 9.1.4.1.

Permissible values:

– implementation-defined:	implementation defined
– content foreground transparency:	content foreground transparency
– colour expression:	as defined in 9.1.4.1

Default values:

implementation-defined:	implementation defined

If the parameter "colour expression" is specified then defaulting rules apply to each of the sub-parameters of the parameter "colour expression", as defined in 9.1.4.1.

Definition:

This attribute specifies the initial value of the foreground colour of the content of the object to which the attribute applies.

If the parameter "implementation-defined" is specified, the content foreground is an implementation-dependent colour which is distinguishable from the content background colour.

NOTE – If the content background is defined as transparent, the content foreground colour is distinguishable from the colour resulting from the colour and transparency specification of the underlying block (see 11.2).

If the parameter "content foreground transparency" is specified, the content foreground is defined as transparent, i.e. the colour and transparency specification of the underlying block also applies to the content foreground (see 11.2). Otherwise, the content foreground occludes the underlying block.

If the parameter "colour expression" is specified, the value is a colour expression as defined in 9.1.4.1. In the case of an indexed colour expression, the relevant colour table is specified in the attribute "content colour table" applicable to the object.

9.4.3.10 Content colour table

Constituents:

Basic page component descriptions, block component descriptions, presentation styles applicable to basic page descriptions or block descriptions.

Classifications:

Non-mandatory for object class descriptions.
Non-mandatory for presentation styles.
Defaultable for object descriptions.

Structure:

This attribute has the structure of a colour table, as defined in 9.1.4.2.

Permissible values:

The permissible values are as defined in 9.1.4.2.

Default values:

The default values are as defined in 9.1.4.2.

Definition:

The value of this attribute defines a colour table, and can be referenced to specify the colour for contents. Colour tables are further explained in 9.1.4.2.

NOTE – This colour table may be referenced from indexed colour expressions (see 9.1.4.1) by the attributes "content background colour" and "content foreground colour" applying to the object.

9.4.4 Presentation attributes

Constituents:

Basic page component descriptions, block component descriptions, presentation styles.

Classification:

Non-mandatory for object class descriptions.
Non-mandatory for presentation styles.
Defaultable for object descriptions.

Structure:

This attribute consists of one or more sets of attributes.

Permissible values:

A number of sets of presentation attributes, each set applicable to one particular content architecture class. These attributes are as defined in:

- Those Specifications of the ITU-T Rec. T.410-Series | ISO/IEC 8613 which specify the individual content architectures.
- Specifications of content architectures that are not included in the ITU-T Rec. T410-Series | ISO/IEC 8613. This is possible,
  - provided a content architecture is defined to the interface specified in clause 8;
  - provided a document application profile is applicable to the document,
  - provided that this content architecture is referenced by the document application profile applicable to the document.

NOTE – Document application profiles are defined as specified in ITU-T Rec. T.411 | ISO/IEC 8613-1.

Default values:

The individual presentation attributes are independently defaultable.

Definition:

A number of sets of presentation attributes may be specified. Which set applies to a given basic component depends on the content architecture class of the content associated with the component.

9.5 Logical attributes

The attributes defined in this sub-clause are applicable to logical components and are not applicable to layout components.

9.5.1 Protection

Constituents:

Logical component descriptions

Classification:

Non-mandatory for object class descriptions.
Defaultable for object descriptions.

Permissible values:

A set of data elements defined for the attribute:

protected,

unprotected.

Default value:

unprotected.

Definition:

This attribute specifies whether or not the logical object, and any associated content portions, are intended to be protected from having any attributes modified by the recipient.

If the attribute is specified for a composite logical component description, then it is applicable only to that description. If the attribute is specified for a basic component description, then it is applicable to that description and associated content portion descriptions.

9.5.2 Layout style

Constituents:

Logical component descriptions.

Classification:

Non-mandatory.

Permissible values:

A layout style identifier or 'null'.

Definition:

This attribute is used to establish a relationship between a logical component and a layout style.

If this attribute has the value 'null' then no layout style is referenced from the logical component description for which this attribute is specified.

9.6 Layout style attributes

9.6.1 Layout style identifier

Constituents:

Layout styles.

Classification:

Mandatory.

Permissible values:

A sequence of two non-negative integers, the first of which is always 4.

Representation:

A character string consisting of two decimal-coded numerals with a space character as a separator between the numerals.

Definition:

This attribute identifies a layout style uniquely within the context of the document.

9.6.2 Attributes which can be specified for layout styles

The following attributes can be specified for layout style descriptions:

- layout style identifier (see 9.6.1);
- user-readable comments (see 9.3.5.1);
- application comments (see 9.3.5.2);
- user-visible name (see 9.3.5.3);
- derived from (see 9.3.3.8);
- sealed (see 9.3.6.2);
- layout directive attributes (see 9.7).

Apart from the attribute "layout style identifier", which is mandatory for layout styles, the attributes are non-mandatory for layout styles.

The attributes "user-readable comments", "user-visible name" and "application comments" are used to describe the style itself and are not referenced by the defaulting mechanism for the purpose of determining values for attributes of the same name for object descriptions.

9.7 Layout directive attributes

9.7.1 General principles of layout directive attributes

9.7.1.1 Layout directive attributes

A layout directive attribute is an attribute of a layout style which guides the generation of a layout structure from a logical structure.

Layout directive attributes are characterised as follows:

- they apply to a logical component as a whole and cannot be changed within the content;
- they are content architecture independent;
- during the layout process, they affect the creation and position of layout objects (see clause 10 for a specification of the reference document layout process).

Some of the layout directive attributes may be applied only to basic logical component descriptions, some only to composite logical component descriptions and some in both cases. The applicability to logical object types and the default value for each attribute is specified in the individual attribute definitions under "constituents".

Layout directive attributes are applied subject to their conforming to the layout object class descriptions.

9.7.1.2 Stream categories

A *stream category* applies to logical objects and to layout objects in order to specify and restrict the layout objects into which the content associated with logical objects may be placed.

A stream category consists of a stream root category followed by a sequence of zero or more stream sub-categories. Each member in this sequence corresponds to a hierarchic level of the specific layout structure or specific logical structure superior to that layout or logical object to which the stream category applies. Each member in the sequence applies to an object description on the path through the hierarchic structure from the object to which the stream root category applies to this object. The stream sub-categories occur in the stream category in the same sequence as the object descriptions to which they apply occur on this path.

NOTE – The stream category is used only as a concept in this specification; it never occurs as a parameter value and is never represented in the data stream in the form of a hierarchic identifier.

Any number of stream categories can apply to a layout object (see 9.4.2.3, 9.4.2.4).

Only one stream category can apply to a logical object (see 9.7.9, 9.7.10)

The content associated with a logical object is constrained to be placed within layout objects to which a stream category applies that is the same as that which applies to the logical object.

The content associated with logical objects to which the same stream category applies is laid out such that the sequential layout ordering of the content is the same as its sequential logical ordering. The content associated with logical objects to which different stream categories apply may be laid out such that the sequential layout ordering of the content is different from its sequential logical ordering.

Further detail on streams is given in the reference model of the layout process (see clause 10, in particular 10.3).

9.7.2 Block alignment

Constituents:

May be specified for layout styles;
applicable only to basic logical component descriptions.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Permissible values:

A set of data elements defined for the attribute:

- right-hand aligned,
- left-hand aligned,
- centred,
- null.

Default value:

right-hand aligned.

Definition:

This attribute specifies the alignment of the block(s) used to present the content associated with the basic logical object to which it applies, within the available area(s) of their immediately superior layout object(s) (see 6.4.2), subject to satisfying constraints on placement specified by the attribute "offset" (see 9.7.12). The alignment specified by this attribute is in the direction orthogonal to that specified by the attribute "layout path" of the lowest level frame(s) containing the block(s).

The value of this attribute specifies the alignment relative to the direction of layout path.

When the attribute "layout path" specifies 270°, 'right-hand aligned' means that the block(s) will appear left aligned on the presentation medium within the available area(s), 'left-hand aligned' means that the block(s) will appear right aligned on the presentation medium within the available area(s), and 'centred' means that the block(s) will appear centred on the presentation medium, within the available area(s).

The value 'null' indicates that no constraints on the layout of the logical object are specified by this instance of this attribute.

9.7.3 Concatenation

Constituents:

May be specified for layout styles;
applicable only to basic logical component descriptions.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Permissible values:

A set of data elements defined for the attribute:

- concatenated,
- non-concatenated.

Default value:

non-concatenated.

Definition:

This attribute specifies whether or not the content associated with the basic logical object to which it applies and the content associated with the preceding basic logical object which has the same values for all of:

- content architecture class (see 9.3.4),
- layout category (see 9.7.7.),
- stream category (see 9.7.1.2),
- fill order (see 9.7.4),

are to be concatenated. That is, the attribute indicates whether the content associated with the object and the preceding object are to be treated as an unbroken stream. There may be other basic logical objects between the two in sequential logical order, but these must not have the same set of values for content architecture class, layout category, stream category and fill order.

The value 'concatenated' specifies that the layout of the content associated with the basic logical object shall if possible be continued in the same basic layout object as used with the content associated with this preceding basic logical object.

The value 'non-concatenated' specifies that the content associated with the basic logical object shall be laid out starting in a new basic layout object.

Each content architecture defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613 specifies whether or not the function of concatenation can be applied to content of that content architecture. In the case of any content architecture for which concatenation can be applied, the rules for concatenation as they affect presentation attributes are included in the definition of that content architecture.

9.7.4 Fill order

Constituents:

May be specified for layout styles;
applicable only to basic logical component descriptions.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Permissible values:

A set of data elements defined for the attribute:

- normal order,
- reverse order.

Default value:

'normal order'.

Definition:

This attribute specifies how the block(s) used to present the content associated with the basic logical object to which it applies are to be laid out within their immediately superior layout object(s), relative to the direction of layout path of these superior objects.

In the case of either 'normal order' or 'reverse order' the blocks are to be positioned in the direction of layout path, in the sequential logical order of the logical objects whose content they are used to present.

The value 'normal order' specifies that the blocks are positioned starting from the trailing edge of the immediately superior layout object, subject to constraints specified by other layout directive attributes.

The value 'reverse order' specifies that the blocks are positioned ending at the leading edge of the immediately superior layout object, subject to constraints specified by other layout directive attributes.

(For the definition of leading and trailing edges, see 7.3.3.)

9.7.5 Floatability range

Constituents:

May be specified for layout styles;
applicable to all logical component descriptions, except the document logical root.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Structure:

This attribute consists of two optional parameters: forward limit, backward limit.

The parameter "forward limit" consists of the sub-parameter "logical object" and an optional sub-parameter "layout object".

The parameter "backward limit" consists of the sub-parameter "logical object" and an optional sub-parameter "layout object".

Permissible values:

For each of the parameters "forward limit", "backward limit":

For the sub-parameter "logical object", one of:

a) the identifier of a logical object description;

b) an object identifier expression;

c) null.

When this attribute is specified for a layout style that is referred to from any logical object class description, case a) is not permissible.

For the sub-parameter "layout object":

- the identifier of a layout object class description;
- the identifier of a stream root category;
- the identifier of a layout category;
- object type page;
- null.

Default value:

The parameters and sub-parameters are independently defaultable.

The default is that each of "forward limit" and "backward limit" is present with sub-parameter values:

- logical object: null
- layout object: null.

Definition:

The content associated with the logical object to which this attribute applies and all of its subordinates (if any), is called the affected content. The purpose of this attribute is to limit how the affected content shall be laid out in sequential layout order with respect to the content associated with the limit objects.

If the sub-parameter "logical object" of the parameter "backward limit" is not 'null', this attribute specifies that none of the affected content is to be laid out earlier in the sequential layout order than any of the content associated with the logical object identified by "backward limit" and with all of its subordinates (if any), with the possible exception of a single layout object as specified by the sub-parameter "layout object", if that is non-null, in which both sets of content may be laid out. If the sub-parameter "logical object" has the value 'null' then no constraints on the layout process are specified by this instance of the parameter "backward limit".

If the sub-parameter "logical object" of the parameter "forward limit" is not 'null', this attribute specifies that all the content associated with the logical object identified by "forward limit" and all of its subordinates (if any), is to be laid out later in the sequential layout order that the affected content, with the possible exception of a single layout object as specified by the sub-parameter "layout object", if that is non-null, in which both sets of content may be laid out. If the sub-parameter "logical object" has the value 'null' then no constraints on the layout process are specified by this instance of the parameter "forward limit".

The two layout objects specified by the sub-parameters "layout object" may be identical or may be different.

NOTE – The default values for "backward limit" and for "forward limit" lead to no constraints on the layout process.

If it is not possible to satisfy the constraints imposed by both "backward limit" and "forward limit" the constraint imposed by "backward limit" shall take precedence.

Further detail on this attribute is included in the definition of the reference model of the layout process (see clause 10, in particular 10.4.2).

9.7.6 Indivisibility

Constituents:

May be specified for layout styles;
applicable to all logical component descriptions except the document logical root.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Permissible values:

One of the following:

a) the identifier of a layout object class description;

b) the identifier of a layout category;

c) object type page;

d) null.

Default value:

null.

Definition:

This attribute specifies that the content associated with the logical object to which it applies shall if possible be laid out within a single layout object which is of a specified object class, layout category or object type.

This attribute does not restrict the layout of other logical objects within the same layout object.

If the value is as in a) above, the layout object shall be of object type page set, page or frame and shall be of the specified layout object class.

If the value is as in b) above, the layout object shall be of the specified layout category.

If the value is as in c) above, the layout object shall be of object type page.

The value 'null' indicates that no constraints on the layout of the logical object are specified by this instance of this attribute.

Further detail on this attribute is included in the definition of the reference model of the layout process (see clause 10, in particular 10.4.4).

9.7.7 Layout category

Constituents:

May be specified for layout styles;
applicable only to basic logical component descriptions.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Permissible values:

Either ‘null’ or a string of characters from the minimum subrepertoire of ISO/IEC 6937, this being the identifier of a layout category. The implicit layout category is represented by the value ‘null’.

Default value:

null.

Definition:

This attribute specifies the layout category of the logical object. A layout category has an identifier which may be associated with basic logical component descriptions and with lowest level frame component descriptions in order to specify and restrict the layout objects into which the content associated with basic logical objects may be placed.

The value of this attribute is the identifier of a layout category. A logical component can specify only a single layout category. The content associated with the basic logical object is constrained to be placed within frame(s) that specify a matching layout category in the list of identifiers specified by the attribute “permitted categories”.

The content associated with logical objects to which the same layout category applies is laid out such that the sequential layout ordering of the content is the same as its sequential logical ordering. The content associated with logical objects to which different layout categories apply may be laid out such that the sequential layout ordering of the content is different from its sequential logical ordering.

Thus, the effect of using more than one layout category within a specific logical structure is to divide the content into different layout streams (see 10.3).

The content associated with more than one basic logical object may be laid out in the same frame, whether their layout categories are the same or different, provided that the frame specifies a matching layout category identifier for each of the layout categories that apply to the logical objects. Also, the content associated with basic logical objects which have the same layout category may be laid out in layout objects generated from different layout object classes, provided that each frame specifies a matching layout category.

If the attribute value is 'null' then the logical object specifies the implicit layout category and the associated content is permitted to be laid out only into frames for which the implicit layout category has been specified, i.e. for which the attribute “permitted categories” has a value identifying the implicit layout category.

9.7.8 Layout object class

Constituents:

May be specified for layout styles;
applicable to all logical component descriptions.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Permissible values:

An identifier of a layout object class description, or 'null'.

Default value:

null.

Definition:

This attribute specifies the class of a layout object into which the content associated with this logical object and all its subordinates is to be laid out. The content shall be laid out within a single instance of a layout object derived from the referenced layout object class, and no other part of the content of the document may be laid out within the same layout object, except for:

- content that results from one of the attributes "logical source", "content generator", or "content portions" in constituents of the generic layout structure referenced by the instance of the layout object, or its subordinates;
- layout object classes in a resource document referenced by the generic layout structure referenced by the instance of the layout object or its subordinates.

This attribute may be used only to specify layout object classes of object type document layout root, page set, page or frame.

This attribute may be applied to logical objects at any hierarchical level within the logical structure. It is valid for this attribute to apply to a logical object when the attribute also applies to one or more superior logical objects. The constraints specified by the attributes in such cases are cumulative. In every such case, the layout object into which the content associated with a subordinate logical object is placed shall be a subordinate of the layout object into which a superior logical object is placed. Consequently, the generic structure is required to be such that objects of the layout object class specified by the subordinate may be generated within objects of the layout object class specified by the superior.

If this attribute is not present on any superior logical object, then there is no restriction on the layout object class specified by this attribute.

The attribute "layout object class" takes precedence over "layout category". Thus, when a composite logical object has the attribute "layout object class", the layout categories applicable to subordinate basic logical objects are valid only within the layout sub-structure subordinate to the layout object corresponding to the specified "layout object class". However, the semantics of the attribute "layout category" are not overridden; it is required that the layout categories of basic logical objects match those of the frames into which they are placed.

A 'null' value indicates that no constraints on the layout of the logical object are specified by this instance of this attribute.

Further detail on this attribute is included in the definition of the reference model of the layout process (see clause 10, in particular 10.3.4).

9.7.9 Logical stream category

Constituents:

May be specified for layout styles;
applicable to all logical component descriptions, except the document logical root.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Permissible values:

Either ‘null’ or a string of characters from the minimum subrepertoire of ISO/IEC 6937, this being the identifier of a stream root category. The implicit stream root category is represented by the value ‘null’.

Default value:

null.

Definition:

This attribute is applicable for an object only if no superior object has a non-null value for this attribute. If any superior object has a non-null value for this attribute, then this instance of this attribute is ignored.

This attribute specifies the identifier of the stream root category applicable to the logical object and to all of its subordinate objects.

A stream root category has an identifier which may be associated with logical component descriptions and which applies to all subordinate logical component descriptions. A stream root category may also be associated with a layout component description (see 9.4.2.3). Stream root categories are used to specify and restrict the layout objects into which the content associated with logical objects may be placed.

In the case of a basic logical object, the content associated with the basic logical object is constrained to be placed within layout objects to which a stream root category applies which has the same identifier as the stream root category that applies to the basic logical object.

In the case of a basic logical object, if the attribute value is 'null', then the logical object specifies the implicit stream root category and the associated content is permitted to be laid out only into layout objects for which the implicit stream root category has been specified, i.e. for which the attribute “layout root categories” has a value identifying the implicit stream root category.

9.7.10 Logical stream sub-category

Constituents:

May be specified for layout styles;
applicable to all logical component descriptions, except the document logical root.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical component descriptions.
Defaultable when applied to logical object descriptions.

Permissible values:

Either ‘null’ or a string of characters from the minimum subrepertoire of ISO/IEC 6937, this being the identifier of a stream sub-category.

Default value:

null.

Definition:

This attribute is applicable for an object only if a superior object has a non-null value for the attribute “logical stream category”. If no superior object has a non-null value for the attribute “logical stream category”, then this instance of this attribute is ignored.

This attribute specifies the identifier of a stream sub-category which is applicable to the logical object and to all of its subordinate objects. Together with the stream root category of the superior logical object to which the attribute "logical stream category" applies and the stream sub-categories of intermediate superior logical objects to which the attribute "logical stream sub-category" applies (if any) it forms a full hierarchic identification of the stream category that applies to the logical object (see 9.7.1.2).

The stream category specifies and restricts the layout objects into which the content associated with logical objects may be placed. In the case of a basic logical object, the content associated with the basic logical object is constrained to be placed within layout objects that specify the same stream category.

9.7.11 New layout object

Constituents:

May be specified for layout styles;
applicable to all logical component descriptions except the document logical root

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Permissible values:

One of the following:

a) the identifier of a layout object class description;

b) the identifier of a layout category;

c) object type page;

d) null.

Default value:

null.

Definition:

This attribute specifies that the content associated with the logical object to which it applies shall be laid out starting within the next layout object (from a current layout position, see below) which does not contain any content associated with preceding logical objects, and which is of a specified layout object class, layout category or object type.

The set of layout categories that apply to the object to which this layout directive applies, if it is a basic logical object, or any of its subordinates, if it is a composite logical object, is called the "affected set".

The current layout position in the specific layout structure is the position in the sequential layout order of that basic layout object that is latest in the sequential layout order of the set of all basic layout objects in which content of preceding basic logical objects to which a layout category from the affected set applies was laid out. If there is no such basic object, then the current layout position is the document layout root.

If the value is as in a) above, the next layout object shall be of object type page set, page or frame and shall be of the specified layout object class.

If the value is as in b) above, the next layout object shall be of the specified layout category.

If the value is as in c) above, the next layout object shall be of object type page.

The value 'null' indicates that no constraints on the layout of the logical object are specified by this instance of this attribute. That is, the content associated with the logical object is to be laid out starting at the current layout position, if possible.

Further detail on this attribute is included in the definition of the reference model of the layout process (see clause 10, in particular 10.4.1).

9.7.12 Offset

Constituents:

May be specified for layout styles;
applicable only to basic logical component descriptions.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Structure:

Four optional parameters, leading offset, trailing offset, left-hand offset, right-hand offset. One or more of the parameters shall be specified in any instance of this attribute.

Permissible values:

For each parameter, a non-negative integer.

Default value:

Each of the four parameters is independently defaultable.

The default for each parameter is: 0.

Definition:

This attribute constrains the available area (see 6.4.2) within the immediately superior frame or page in which the content associated with the basic logical object may be placed.

This attribute specifies minimum amounts of offset between the boundary of a block used to present the content associated with this logical object and the boundary of the immediately superior layout object.

The parameters "leading offset", "trailing offset", "left-hand offset", "right-hand offset" specify the minimum distance between the corresponding edge of the block containing the content associated with this logical object and the leading, trailing, left-hand and right-hand edge of the immediately superior layout object, respectively. (The names of the edges are as defined in 7.3.3.)

For each edge, the corresponding parameter specifies the amount of offset for that edge in scaled measurement units.

9.7.13 Same layout object

Constituents:

May be specified for layout styles;
applicable to all logical component descriptions, except the document logical root.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Structure:

Two optional parameters: logical object, layout object.

Permissible values:

For the parameter "logical object", one of:

a) the identifier of a logical object description;

b) an object identifier expression;

c) null.

When this attribute is specified for a layout style that is referred to from any logical object class description, case a) is not permissible.

For the parameter "layout object":

If the first parameter has the value 'null', then this parameter is ignored, otherwise the value shall be one of:

a) the identifier of a layout object class description;

b) the identifier of a stream root category;

c) the identifier of a layout category;

d) object type page;

e) null.

The value 'null' is permitted only when the value 'null' is also specified by the parameter “logical object”.

Default value:

The parameters are independently defaultable.

The default values are:

– logical object:	null
– layout object:	null.

Definition:

This attribute specifies that the start of the content associated with the logical object to which it applies and the end of the content associated with another specified logical object identified by the parameter “logical object” shall be laid out, if possible, within a single layout object, as specified by the parameter “layout object”.

The block or blocks containing the two content portions concerned are:

- the block that is least advanced in the sequential layout order and that receives content associated with the logical object to which this attribute applies or with its subordinates;
- the block that is most advanced in the sequential layout order and that receives content associated with the logical object identified by “logical object” or with its subordinates.

If the value of the parameter “layout object” is as in a) above, the layout object containing these block(s) shall be of object type page set, page or frame and shall be of the specified layout object class.

If the value of the parameter “layout object” is as in b) or c) above, the layout object containing these block(s) shall be of the specified category.

If the value of the parameter “layout object” is as in d) above, the layout object containing these block(s) shall be of object type page.

A 'null' value of the parameter “logical object” indicates that no constraints on the layout of the logical object are specified by this instance of this attribute.

Further detail on this attribute is included in the definition of the reference model of the layout process (see clause 10, in particular 10.4.3).

9.7.14 Separation

Constituents:

May be specified for layout styles;
applicable only to basic logical component descriptions.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Structure:

Three optional parameters, leading edge, trailing edge, centre separation. One or more of the parameters shall be specified in any instance of this attribute.

Permissible values:

For each parameter, a non-negative integer.

Default values:

Each of the three parameters is independently defaultable.

The default value for each parameter is: 0.

Definition:

This attribute specifies minimum amounts of separation between the block(s) used to lay out the content associated with the basic logical object to which the attribute applies and the nearest adjacent block(s) immediately subordinate to the same immediate superior layout object.

The parameter "leading edge" specifies the minimum separation from the leading edge of the last block containing content associated with this logical object and the trailing edge of the next block laid out in the same fill order.

The parameter "trailing edge" specifies the minimum separation from the trailing edge of the first block containing content associated with this logical object and the leading edge of the closest preceding block laid out in the same fill order.

Thus, a constraint on the separation of two adjacent blocks which both have the same fill order is that the separation shall be equal to, or greater than, the greater of the value of the parameter "leading edge" for the logical object with content laid out in the first of the blocks in the direction of layout path and the value of the parameter "trailing edge" for the logical object with content laid out in the second of the blocks.

The parameter "centre separation" specifies the minimum distance between two objects within a frame that are laid out with opposite directions of fill order.

Thus, a constraint on the separation of two adjacent blocks which have different fill orders is that the separation shall be equal to, or greater than, the maximum of the value of the parameter "centre separation" specified for the logical objects with content laid out in the two blocks.

For each edge, the parameter specifies the amount of separation for that edge in scaled measurement units. (The names of the edges are as defined in 7.3.3.)

9.7.15 Synchronisation

Constituents:

May be specified for layout styles;
applicable to all logical component descriptions, except the document logical root.

Classification:

Non-mandatory when specified for layout styles.
Non-mandatory when applied to logical object class descriptions.
Defaultable when applied to logical object descriptions.

Permissible values:

One of:

a) the identifier of a logical object description;

b) an object identifier expression;

c) null.

When this attribute is specified for a layout style that is referred to from any logical object class description, case a) is not permissible.

Default value:

null.

Definition:

The attribute specifies that content associated with the component to which the attribute applies and content associated with another specified logical component are to be laid out aligned along a line orthogonal to the direction of the layout path.

Specifically two blocks shall be laid out, if possible, aligned along a line orthogonal to the direction of the layout path. These blocks are:

- the block that is least advanced in the sequential layout order and that receives content associated with the logical object to which this attribute applies and with all of its subordinates;
- the block that is least advanced in the sequential layout order and that receives content from the other specified logical component and with all of its subordinates.

The trailing edges of both these blocks shall be synchronised, i.e. the lines along the trailing edges are aligned.

The two blocks involved shall be placed into distinct lowest level frames. The direction of fill order for the two blocks shall be the same, and the direction of layout path shall be the same in the two lowest level frames for this attribute to have in effect. If any of these conditions are not satisfied this attribute shall be ignored. The frames may have the same or different layout categories and may be on the same or different pages.

The value 'null' indicates that no constraints on the layout of the content associated with the logical object are specified by this instance of this attribute.

Further detail on this attribute is included in the definition of the reference model of the layout process (see clause 10, in particular 10.4.5).

9.7.16 Interactions and precedences among the layout directive attributes

This clause describes the order in which the different layout directive attributes are to be taken into account in the layout process.

These rules on precedence specify additional definition material on the use of the attributes in combination. However, the interactions and precedences described here only provide guidelines, for clarifying the semantics of the attributes. These rules are not intended to be taken as a complete formal specification of the interaction resolution mechanism. They are not intended to represent an actual implementation, nor to restrict in any way the processing that may be applied to an interchanged document.

It is assumed that a document which is defined in accordance with the ITU-T Rec. T.410-Series | ISO/IEC 8613 can be laid out in accordance with the rules described below. That is, it is assumed that each attribute pertaining to the layout process can be interpreted in accordance with the definition of that attribute together with these rules on precedence. It is outside the scope of the ITU-T Rec. T.410-Series | ISO/IEC 8613 to indicate how a document containing conflicting or inconsistent information is to be laid out.

For composite logical objects descriptions the following layout directive attributes are applicable, listed in the order of decreasing precedence:

- layout object class;
- logical stream category;
- logical stream sub-category;
- new layout object;
- floatability range;
- same layout object;
- synchronization;
- indivisibility.

This set of layout directive attributes shall not only be taken into account at a particular level but also at hierarchically related levels. This means that each layout directive attribute applicable to a subordinate logical object description shall result in a valid layout as defined by the layout directive attributes applicable to all superior logical object descriptions. That is, all layout directive attributes applicable to a logical object description have precedence over any layout directive attributes applicable to logical object descriptions subordinate to that object description.

For basic logical objects descriptions the following layout directive attributes are applicable, listed in the order of decreasing precedence:

- layout object class;
- logical stream category;
- logical stream sub-category;
- layout category;
- new layout object;
- floatability range;
- same layout object;
- fill order;
- concatenation;
- offset;
- separation;
- synchronization;
- indivisibility;
- block alignment.

As specified above, the layout directive attributes applicable to basic logical object descriptions shall also be valid within the set of layout directive attributes applicable to all superior logical object descriptions.

The rules for the individual layout directive attributes, in order of decreasing precedence are:

a) *Layout object class*

The specification made by this attribute shall be fulfilled if this is possible within the constraints imposed by the attribute "layout object class" of any superior component.

When this attribute specifies a lowest level frame, then the following attributes shall be ignored:

- new layout object;
- same layout object;
- indivisibility.

b) *Logical stream category*

The specification made by this attribute shall be fulfilled if this is possible within the constraints imposed by the attribute "layout object class" of this or any superior components. This attribute shall be ignored if a value for this attribute applies to any superior component.

c) *Logical stream sub-category*

The specification made by this attribute shall be fulfilled if this is possible within the constraints imposed by the attribute "layout object class" of this or any superior components, by the attribute "logical stream category" of a superior component and by the attribute "logical stream sub-category" of any superior components subordinate to a component to which the attribute "logical stream category" applies.

d) *Layout category*

The specification made by this attribute shall be fulfilled.

e) *New layout object*

The specification made by this attribute shall be fulfilled if the attribute "layout object class" does not apply to this logical object.

When the attribute "layout object class" is applied to this logical object and the layout object of that class is of the kind specified by the attribute "new layout object", then the specification made by this attribute is automatically fulfilled. Otherwise a subordinate of the object of that class shall fulfil this specification.

If the attribute "layout object class" has specified a lowest level frame for this or a superior logical object, then this attribute shall be ignored.

f) *Floatability range*

This attribute shall be fulfilled if this is possible within the constraints imposed by the attributes "layout object class", "logical stream category", "logical stream sub-category", "layout category" and "new layout object".

g) *Same layout object*

The specification made by this attribute shall be fulfilled, except in the following cases:

- the end of the content associated with the referenced logical object is laid out in a layout object not belonging to the class or category specified;
- the attribute "layout object class" has specified a lowest level frame for this or a superior logical object:
- the attribute "new layout object" has specified an object class, object type or layout category which can not be a subordinate of the layout object specified by this attribute.
- it would require content to be laid out outside a layout object specified by the attribute "layout object class" applicable to this object or one of its superiors;
- it would require content to be laid out within a layout object which is specified by the attribute "layout object class" applicable to an object which is neither a superior nor a subordinate of the object for which the attribute is specified.

h) *Fill order*

The specification made by this attribute shall be fulfilled.

j) *Concatenation*

When this attribute specifies the value 'non-concatenated', the specification made by this attribute shall be fulfilled.

When this attribute specifies the value 'concatenated', the specification made by this attribute shall be ignored and the content associated with the logical component shall not be concatenated if either of the following conditions apply:

- the closest preceding basic logical object in logical sequential order which has the same layout category, stream category and fill order does not have the same content architecture class as this logical object;
- the attribute "layout object class" or the attribute "new layout object" also applies to the component.

In addition, the content need not be concatenated if this is necessary in order to satisfy the attribute "balance"

If none of the preceding conditions applies then the content associated with the logical object shall be concatenated if there is adequate available area.

The attributes "separation", "offset", "border", "colour", "transparency", "colour of layout object", "object colour table", "content colour table" and "block alignment", applicable to the same logical component shall be ignored when the content associated with the logical object is concatenated.

The attribute "indivisibility" and the attribute "same layout object" may be used in conjunction with the attribute "concatenation".

k) *Offset*

The specification made by this attribute shall be fulfilled except when the attribute is applied to basic logical objects that are concatenated to preceding logical objects.

l) *Separation*

The specification made by this attribute shall be fulfilled except when the attribute is applied to basic logical objects that are concatenated to preceding logical objects.

m) *Synchronization*

When this attribute specifies a value other than 'null', the specification shall be fulfilled provided that all conditions for synchronization are possible without violating any layout directive attributes of higher precedence.

n) *Indivisibility*

When this attribute specifies a value other than 'null', the specification shall be fulfilled provided that all conditions for indivisibility are possible without violating any layout directive attributes of higher precedence.

o) *Block alignment*

The specification made by this attribute shall be fulfilled except for the part of the content associated with the logical object which is concatenated with the content associated with another logical object.

The attribute is applied subject to satisfying constraints on placement specified by the attribute "offset".

9.7.17 Interactions among attributes affecting the layout process

All layout directive attributes affect the document layout process. This is also true for some of the attributes of the generic layout structure.

Similarly to the layout directive attribute "layout category" the specifications made by the attribute "permitted categories" shall be fulfilled.

When the layout process requires the creation of a new composite layout object, its creation is controlled by the attribute "generator for subordinates".

The specification made by the attribute "balance" shall be fulfilled subject to satisfying any constraints specified by any layout directive attributes.

The specification made by the attribute "layout path" is used for determining the sides specified in the attributes "fill order", "offset", "separation" and "block alignment". It is also used by the layout directive attribute "synchronization" as well as the layout attributes "position", "dimensions" and "balance".

The attribute "border" may also constrain the position and dimensions of layout objects.

The attributes "transparency" and "imaging order" have no effect on the creation of layout objects, nor do they affect their position and dimensions.

9.8 Presentation style attributes

9.8.1 Presentation style identifier

Constituents:

Presentation styles.

Classification:

Mandatory.

Permissible values:

A sequence of two non-negative integers, the first of which is always 5.

Representation:

A character string consisting of two decimal-coded numerals with a space character as a separator between the numerals.

Definition:

This attribute identifies a presentation style uniquely within the context of the document.

9.8.2 Attributes which can be specified for presentation styles

The following attributes can be specified for presentation style descriptions:

- presentation style identifier (see 9.8.1);
- user-readable comments (see 9.3.5.1);
- user-visible name (see 9.3.5.3);
- application comments (see 9.3.5.2);
- border (see 9.4.1.3);

- transparency (see 9.4.3.2);
- colour (see 9.4.3.3);
- content colour table (see 9.4.3.10);
- content background colour (see 9.4.3.8);
- content foreground colour (see 9.4.3.9);
- colour of layout object (see 9.4.3.6);
- object colour table (see 9.4.3.7);
- derived from (see 9.3.3.8);
- sealed (see 9.3.6.2);
- presentation attributes (see 9.4.4).

Apart from the attribute "presentation style identifier", which is mandatory for presentation styles, the attributes are non-mandatory for presentation styles.

The attributes "user-readable comments", "user-visible name" and "application comments" are used to describe the style itself and are not referenced by the defaulting mechanism for the purpose of determining values for attributes of the same name for object descriptions.

All attributes in presentation styles apply only to basic component descriptions. The attributes "border", "transparency" and "colour" may only apply to blocks when specified in a presentation style. (Such a reference can be either direct, if the presentation style is referred to by a layout component, or indirect, if the presentation style is referred to by a logical component.)

9.9 Content portion attributes

The attributes defined in this sub-clause are applicable to content portions only.

The attributes are as defined in:

- Those Specifications of the ITU-T Rec. T.410-Series | ISO/IEC 8613 which specify the individual content architectures.
- Specifications of content architectures that are not included in the ITU-T Rec. T410-Series | ISO/IEC 8613. This is possible
  - provided a content architecture is defined to the interface specified in clause 8;
  - provided a document application profile is applicable to the document;
  - provided that this content architecture is referenced by the document application profile applicable to the document.

  NOTE – Document application profiles are defined as specified in ITU-T Rec. T.411|ISO/IEC 8613-1.

9.9.1 Identification attributes
Content identifier – logical;
Content identifier – layout

Constituents:

Content portion descriptions.

Classification:

Each of the attributes individually is non-mandatory. At least one of the attributes shall be specified for every content portion description, unless the exceptional case described below applies.

Structure:

A pair of attributes.

Permissible values:

For each attribute, a sequence of non-negative integers.

Representation:

For each attribute, a character string consisting of decimal numerals and space characters. The decimal numerals are in one to one correspondence with the integers constituting the identifier: a space character is used as a separator between successive numerals.

Definition:

These attributes identify a content portion description uniquely within the context of the document and are used to refer to that content portion description.

These attributes are used in the context of relationships to content portions (see 9.3.3.3).

The value of each attribute consists of a sequence of integers, as defined in 9.3.1.2 and 9.3.1.3 for an identifier of a basic component, with an additional integer at the end of the sequence to identify the content portion uniquely among the set of content portions that are associated with the relevant basic component.

A content portion description in the specific structure can specify one or both attributes.

The attribute "content identifier – layout" is specified when the content portion is associated with a basic layout object. The attribute "content identifier – logical" is specified when the content portion is associated with a basic logical object.

A generic content portion description shall have only one identifier attribute, according to whether the content portion is associated with a layout object class or a logical object class.

When a document is reformatted any content portion descriptions that are associated with the specific layout structure only are deleted. The content portions which are deleted are any that specify the attribute "content identifier – layout" and do not specify the attribute "content identifier – logical".

Exceptions:

This attribute is non-mandatory in certain documents. These documents are those which have all of the following characteristics:

- The interchange format class used for document interchange is class B (see ITU-T Rec. T.415 | ISO/IEC 8613-5), consequently the only specific structure present is the specific layout structure.
- The only object types present in the document are document layout root, pages and blocks.
- There is no use of identifiers of content portions in attributes.

In documents adhering to these rules any two consecutive content portions in the data stream are associated with the same basic object.

Thus, under these conditions the semantics of the content identifier attributes can be transmitted to the recipient implicitly and the attribute need not be explicitly specified.

NOTE – This exceptional case is provided for compatibility with ITU-T Recommendations.

9.9.2 Common coding attributes – Type of coding

Constituents:

Content portion descriptions.

Classification:

Defaultable.

Permissible values:

ASN.1 object identifier.

Representation, Default values:

The definition of particular values is included in the specification of individual content architectures.

Definition:

This attribute specifies the coding used to represent the content, and designates any set of additional coding attributes applicable to the content portion concerned (see 9.9.4).

Exceptions:

The value is also permitted to be an integer in the case of the formatted raster graphics content architecture.

NOTE – This exceptional case is provided for compatibility with ITU-T Recommendations.

9.9.3 Content information attributes

9.9.3.1 Content information

Constituents:

Content portion descriptions

Classification:

Non-mandatory

Permissible values:

An octet string.

Representation:

Defined in the specification of individual content architectures.

Definition:

This attribute specifies that part of the content portion description which is composed of content elements (for example, graphic characters, pels, control functions) governed by a content architecture.

9.9.3.2 Alternative representation

Constituents:

Content portion descriptions

Classification:

Non-mandatory.

Permissible values:

A string of characters from a defined character set.

Definition:

This attribute specifies a sequence of characters that may be imaged in lieu of the attribute “content information” when a receiver of the document is not capable of decoding and/or imaging that content information.

The character set to be used in this attribute is that specified in the document profile attribute “alternative representation character sets”.

The default character set is the minimum subrepertoire of ISO/IEC 6937.

In addition to the graphic character set, the control functions carriage return and line feed may be included in the character string.

9.9.4 Coding attributes

Constituents:

Content portion descriptions

Structure:

This attribute consists of a set of one or more attributes, as defined in the specification of individual content architectures.

Classification, Permissible values, Representation, Default values, Definition:

These attributes are related to the type of coding of the content portion and provide additional parametric information used in encoding/decoding the content portion. The attributes are as defined in the specification of individual content architectures.

The attributes are as defined in:

- Those Specifications of the ITU-T Rec. T.410-Series | ISO/IEC 8613 which specify the individual content architectures.
- Specifications of content architectures that are not included in the ITU-T Rec. T.410-Series | ISO/IEC 8613. This is possible
  - provided a content architecture is defined to the interface specified in clause 8,
  - provided a document application profile is applicable to the document,
  - provided that this content architecture is referenced by the document application profile applicable to the document.

  NOTE – Document application profiles are defined as specified in ITU-T Rec. T.411|ISO/IEC 8613-1:

9.10 Protected part attributes

9.10.1 Protected part identifier

Constituents:

Protected part descriptions.

Classification:

Mandatory

Permissible values:

A sequence of two non-negative integers.

The value of the first integer is:

- 6, if the protected part is a sealed document profile description;
- 7, if the protected part is an enciphered document profile description.
- 8, if the protected part is a pre-enciphered document body part description;
- 9, if the protected part is a post-enciphered document body part description.

Representation:

A character string consisting of decimal numerals and space characters. The decimal numerals are in one to one correspondence with the integers constituting the identifier, a space character is used as a separator between successive numerals.

Definition:

This attribute identifies a protected part description uniquely within the context of the document.

9.10.2 Sealed document profile information

Constituents:

Sealed document profile description.

Classification:

Non-mandatory.

Permissible values:

Any subset of a document profile.

Representation:

A document profile description, with the additional property of allowing the value 'null' for any attribute in the document profile that is not classified as mandatory.

Definition:

This attribute consists of the subset of the attributes of a document profile that is sealed for integrity, authenticity or non-repudiation of origin. A value 'null' in an attribute of the sealed document profile indicates that this attribute is sealed as absent.

9.10.3 Enciphered information

Constituents:

Enciphered document profile description, pre-enciphered document body part description, post-enciphered document body part description.

Classification:

Non-mandatory.

Permissible values:

Enciphered information.

Representation:

An octet string.

Definition:

For an enciphered document profile description, this attribute contains the result of a cryptographic algorithm applied to a confidential part of the document profile.

For a pre-enciphered document body part description, this attribute contains the result of a cryptographic algorithm applied to a sequence of constituents of the document body before the layout process has been performed.

For a post-enciphered document body part description, this attribute contains the result of a cryptographic algorithm applied to a sequence of constituents of the document body after the layout provess has been performed.

10 Reference model of the document layout process

This clause provides a description of the document layout process as applicable to documents which contain a specific logical structure, a complete generic layout structure and optionally layout styles, presentation styles, and/or a generic logical structure.

The purpose of the reference model specified in this clause is to aid the understanding of the semantics of the attributes affecting the layout process and in particular the creation of a specific layout structure. It provides additional normative information on the semantics of the attributes defined in clause 9 but the reference model is not intended to specify any process that might be carried out in a particular implementation.

The content layout process, which controls the layout of content portions within basic layout objects, is not described here but is included in the specification of individual content architecures.

10.1 Introduction

10.1.1 Overvew

The document layout process defines the automatic generation of a specific layout structure for a document and the layout of the content of basic logical objects into blocks within lowest level frames in this specific layout structure. During this process the basic logical objects are considered in their sequential logical order in the specific logical structure (see 7.1.2). If necessary, the logical structure is derived from the set of descriptions, including alternative descriptions present in the document, by a process called initialization (see 10.1.2).

The reference model of the document layout process handles only layout into frames, it does not handle the case of documents containing basic pages nor the case of composite pages whose immediate subordinates are blocks.

NOTE – Basic pages and composite pages whose immediate subordinates are blocks are permitted to be included only in documents of the formatted document architecture class (see 7.3.1.3). Although not described by this Specification, the generation by an implementation of the layout process of basic pages or composite pages where immediate subordinates are blocks to produce a document of the formatted document architecture class is not precluded.

The document layout process is carried out in accordance with the values of the layout directive attributes applicable to the logical object descriptions representing the specific logical structure. In effect, layout directive attributes express relationships between objects in the specific logical structure and object classes in a generic layout structure. The specific layout structure which is generated is consistent with the generic layout structure and is in accordance with the layout directive attributes applicable to the logical object descriptions and the logical object class descriptions.

In all cases when attributes of logical component descriptions or layout component descriptions have values which are specified by expressions these values are evaluated by the layout process. Each time a logical object is considered for layout, any applicable attributes which are specified by expressions are evaluated. Each time a layout object is created, any applicable attributes which are specified by expressions are evaluated.

The document layout process involves the creation of a sequence of page sets, pages and frames into which the content of the sequence of basic logical objects is to be laid out. The document layout process controls the allocation of the areas within a frame or sequence of frames into which the content of each basic logical object is to be placed and defines constraints on the area(s) into which the content may be laid out. The document layout process determines when the layout objects which have been created are closed from further use for layout.

The content layout process is responsible for formatting the content into the allocated area taking into account the constraints imposed by the document layout process. The content layout process and document layout process are together responsible for the creation of basic layout objects.

The content layout process determines the dimensions of the basic layout objects. The document layout process is responsible for determining the position of these basic layout objects within their immediately superior layout objects. The document layout process is also responsible for determining the dimensions and positions of frames.

This can be performed by two different mechanisms. When frames have fixed dimensions and positions, a top down approach is made resulting in areas available for positioning blocks. When values for dimensions and positions of frames are specified by rules or expressions, i.e. non-fixed values, a bottom up approach is taken in defining the dimensions and positions, based on the dimensions of basic layout objects. This latter approach is constrained from a top down specification of permissible ranges.

All frames and blocks immediately subordinate to a page are specified with fixed positions and dimensions.

All frames with the same immediate superior frame are either all specified with fixed positions or all specified with non-fixed positions.

All blocks with the same immediate superior frame are either all specified with fixed positions and dimensions, i.e. from the generic layout structure, or are all specified with non-fixed positions and dimensions, i.e. content dependent positions and dimensions.

It is assumed that a document which is defined in accordance with the ITU-T Rec. T.410-Series | ISO/IEC 8613 can be laid out in accordance with the rules described below. That is, it is assumed that each attribute pertaining to the layout process can be interpreted in accordance with the definition of that attribute together with the precedence rules (see 9.7.16). It is outside the scope of the ITU-T Rec. T.410-Series | ISO/IEC 8613 to indicate how a document containing conflicting or inconsistent information is to be laid out.

10.1.2 Initialization

In the case of a document containing any alternative descriptions, then before the layout process commences, the logical structure of the document is derived conceptually from the set of primary descriptions and alternative descriptions in the document by the following initialization of the layout process.

First, a logical structure is created from the primary descriptions in the document, i.e. the root logical object description and all those descriptions which are referred to by the attribute "subordinates" of composite object descriptions. If any of the primary descriptions cannot be decoded by the recipient, this first phase of initialization can substitute one or more primary descriptions by one or more alternative descriptions.

If the resulting logical structure cannot be processed by the recipient (e.g. edited) or may result in layout that cannot be processed by the recipient, this second phase of initialization can substitute one or more primary or alternative descriptions by one or more alternative descriptions.

These substitutions take account of the preference order for substitution specified by the attribute "alternative". That is, selecting in each case the first alternative description in preference order that can be decoded and processed by the recipient.

NOTE – For a more implementation-oriented description of the initialization see Annex F (where the information is informative and is non-integral to this Specification).

10.2 Content and layout structure generation

The generation of the specific layout structure is controlled by the complete generic layout structure. The construction rules for creation of page sets, pages and frames that are required for the layout of a particular specific logical structure are specified in the generic layout structure.

The only basic layout objects for an automatic layout process are blocks. These are created in one of two ways:

- Firstly, blocks may be created as a result of a layout process laying out the content associated with basic logical components. In this case there shall be no layout object class description of object type block.
- Secondly, blocks may be created from a layout object class description of object type block; such a block class description shall specify content, either in the form of generic content portion(s) or by use of the attribute "content generator".

The layout process creates a specific layout structure, which conforms to the complete generic layout structure and which accommodates all the content of the document.

10.2.1 Laying out content of a document

For the layout process, content of a document may be related either to the specific logical structure or to the generic layout structure.

In the generic structures, the content associated with an object class description may be specified by the attribute "content portions" and contained in one or more generic content portions identified by that attribute. Alternatively, the content may be specified by the attribute "content generator", in which case the content is derived from the string expression that is the value of that attribute.

The value of a content generator is determined during the layout process. The evaluation of the expression which specifies the value of the attribute occurs when the content portion is laid out. If the expression refers to other expressions, then these are also evaluated at this point.

10.2.1.1 Content related to the specific logical structures

The content related to the specific logical structure can occur in the following forms:

a) content in a content portion that is associated with a basic logical object;

b) content in a generic content portion that is associated with a basic logical object class description in the generic logical structure or in the resource-documents;

c) content specified by the attribute "content generator" that is specified for a basic logical object;

d) content specified by the attribute "content generator" that is specified for a basic logical object class in the generic logical structure or in the resource-document.

In all cases, the content may be in processable, formatted or formatted processable form.

If case a) applies, the document layout process may cause the creation of a basic layout object that references the same content portion referenced by the basic logical object. This is achieved by adding the attribute "content identifier – layout" to the content portion description. As a result, the content portion will be common to both the specific logical structure and the specific layout structure.

In some cases, the content portion associated with the single basic logical object will cause the generation of two or more basic layout objects. For example, a part of the content may be laid out at the end of one frame and the remainder of the content in the next frame. In this case, the content shall be divided into two or more content portions such that the basic logical object now has associated a sequence of two or more content portions, each of which is referenced by only one of the basic layout objects which have been created.

On the other hand, in some cases a new layout object need not be created if the attribute “concatenation” has been specified for the specific logical object. In this case, the corresponding basic layout object references a sequence of two or more content portions.

If case b) applies, the basic logical object derives its content information from content information associated with a logical object class. In this case the document layout process causes the generation of one or more new content portion(s) that are associated only with the specific layout structure. That is, these new content portion(s) are only referenced by the basic layout object produced during the document layout process.

In cases c) and d), the attribute “content generator” has to be evaluated before being laid out by the document layout process. As in case b), the document layout process results in the creation of one or more new content portion(s) that are associated only with the specific layout structure. If in these cases the basic logical object directly references a content portion (containing no content information) then no change is made to that content portion.

In cases b), c), d), a sequence of content portions may be created, each associated with a different basic layout object as for the examples in case a). Also a new basic layout object need not be created if concatenation has been specified.

10.2.1.2 Content related to the generic layout structure

The content related to the generic layout structure can occur in the following forms:

a) content in a generic content portion that is associated with a basic layout object class description in the generic layout structure or in the resource-document;

b) content specified by the attribute “content generator” that is specified for a basic layout object class description in the generic layout structure or in the resource-document;

c) content specified in the generic logical structure that is referred to from the generic layout structure (by use of the attribute “logical source”).

In case a), the content portion is already formatted and the position and dimensions of the block with which the content portion is associated are specified in the layout object class of object type block. The document layout process will use these. The specific layout objects containing this content, which can be many, will have a reference only to the layout object class description with which the generic content portion is associated and no additional content portions will be generated.

In case b), the attribute has to be evaluated by the layout process before being laid out by the layout process. The evaluated content may be in processable, formatted or formatted processable form. The evaluation will result in a content portion that is associated only with the specific layout structure.

In case c), on the creation of a layout object of a class that has the attribute “logical source” (see 9.4.2.5), an instance of the logical object and its content, if any, of the class specified by that attribute, and its subordinates if it is a composite object, is generated.

In this case, in order for the layout process to be deterministic, the attribute “generator for subordinates” for this logical object class is allowed to specify only a construction expression composed from construction terms consisting only of required construction factors and/or sequence constructions, which use only required construction factors. The same rule applies to the logical object class descriptions for all subordinate composite objects.

The resulting logical objects and associated content portions are then laid out in accordance with the document layout process. In order to lay out this content, one or more layout objects and their associated content portions are added to the specific layout structure being generated by the document layout process.

This is done in such a way that the created logical objects are laid out entirely within the layout object that caused their creation (as if the attribute “layout object class” had been specified for the root of the created tree of logical objects).

The created content portions are associated only to the specific layout structure. The created logical object, or hierarchy of logical objects, is used only for the purpose of the layout process, and the object(s) are not added to the specific logical structure.

10.2.1.3 Reformatting of content of a formatted processable form document

If a document is reformatted then all components in the specific layout structure are first deleted, together with all its content portions that are associated with layout object descriptions and that are not common to both the specific logical and the specific layout structures. All occurrences of the attribute “content identifier – layout” are also removed from the content portions associated with the logical object descriptions. In addition, any content portions associated with logical object descriptions that were divided as a result of the document layout process [see 10.2.1.1., case a)] are recombined into a single content portion.

The reformatting of the content is then performed as described for initial formatting in 10.2.1.1 and 10.2.1.2.

10.2.2 Specific layout structure generation

The specific layout structure generated by the layout process is constrained by the attribute "generator for subordinates" (see 9.3.2.1) which is applicable to layout object class descriptions and which constrains the layout sub-structures, if any, that may be immediately subordinate to layout objects of that class.

This attribute contains a construction expression which specifies the object classes permitted for the immediately subordinate objects to an object of that layout object class, and defines the set of possible combinations and possible orderings of such subordinate objects in the sequential layout order. The construction expression indicates whether or not each subordinate object is required, is optional, has a choice, and if it can be repeated more than once. Construction expressions are defined in 9.3.2.1.

The result of the layout process will be a hierarchic specific layout structure with a well defined sequential layout order that can serve as input to an imaging process.

10.3 Layout references and categories

There are two further principal methods of controlling the generation of a specific layout structure from the specific logical structure, and constraining the allocation of the content of basic logical objects to layout objects within that layout structure; one is provided by stream categories, the other by reference to layout object classes from the specific logical structure.

10.3.1 Stream categories

A stream category can apply to:

- logical components, by use of the attributes "logical stream category", "logical stream sub-category" (see 9.7.9, 9.7.10), which apply to all logical component descriptions except the document logical root;
- layout components, by use of the attributes "layout stream categories", "layout stream sub-categories" (see 9.4.2.3, 9.4.2.4), which apply to all layout component descriptions, except the document layout root.

The effect of the attributes "logical stream category" and "layout stream categories" is that the object to which the attribute applies and all subordinate objects, if any, belong to one or more streams which have a stream root identified by the stream root category identifier(s) specified.

The logical structure and layout structure can be regarded as being segmented into different streams such that each stream in the logical structure is laid out exclusively in a particular matching stream in the layout structure. Thus a stream in the logical structure can be viewed as being laid out in a 'tunnel' in the layout structure.

Only one stream root category can apply to a logical object and by this means the logical structure of a document can be divided into a set of streams.

When a stream root category applies to a logical object then the subordinates of this logical object can not have a different stream root category that applies to them.

A stream in the logical structure may also contain various specialist types of structure and sub-structure, such as tables and composite figures. From the point of view of the logical structure, these structures and their sub-structures can be regarded as sub-streams and sub-sub-streams, etc. within a higher level stream. Similarly, from the layout point of view, each such sub-stream and sub-sub-stream, etc. will be laid out in a part of the layout structure that is specially reserved for that purpose, i.e., in sub-streams and sub-sub-streams, etc. within the layout structure.

This subdivision into sub-streams, sub-sub-streams, etc., is achieved by the use of the attributes "logical stream sub-category" and "layout stream sub-categories".

The layout of a sub-stream in the logical structure is done in the context of its immediately superior stream (or sub-stream) and the layout of sub-streams at the same hierarchical level are independent of one another.

A stream in the logical structure is laid out in a matching stream in the layout structure and sub-streams are laid out in matching sub-streams and so on. This provides the means to create a hierarchical structure of streams.

The naming conventions used for stream categories and stream sub-categories need careful consideration. The stream sub-categories that are immediately subordinate to a given stream root may have different names, the same names or a mixture of both. Stream root categories that have different names are regarded as belonging to different streams in the layout structure. These will be laid out in different streams. Stream root categories that have the same name are regarded as belonging to the same stream and will be laid out in the same stream. Stream sub-categories that are immediately subordinate to different stream root categories may have the same name but such streams are not to be regarded as belonging to the same stream.

10.3.2 Layout categories

A layout category can be specified by the attribute "layout category" (see 9.7.7) which applies to basic logical component descriptions, and by the attribute "permitted categories" (see 9.4.2.6) which applies to frames at the lowest level in any branch of the layout structure. The layout process ensures that the content of basic logical objects of any layout category is placed only in frames which specify a matching layout category.

The effect of specifying different layout category identifiers for different basic logical objects is to cause them to be laid out into sequences of frames having appropriate layout category identifiers.

10.3.3 Basic principles of layout streams

The attributes "layout stream categories", "logical stream category", "layout stream sub-categories", "logical stream sub-category", "layout category" and "permitted categories" are handled as a set of constraints that shall all be satisfied in the layout process. Together they define a hierarchic set of identifiers that guide the construction of a specific layout structure to hold the content associated with the specific logical structure.

The layout process places the content associated with logical objects into layout objects specifying matching stream categories and matching layout categories.

The effect of specifying different stream category and/or different layout category identifiers for different logical objects is to partition those logical objects into different *layout streams*. Each layout stream corresponds to a particular stream category and layout category. These different layout streams are laid out into objects within the layout structure to which matching stream category and layout category identifiers apply.

The content associated with logical objects to which the same layout stream applies shall be laid out such that the sequential layout ordering of the content is the same as its sequential logical ordering.

The content associated with logical objects to which different layout streams apply may be laid out such that the sequential layout ordering of the content is different from its sequential logical ordering.

The content associated with more than one logical object may be laid out in the same layout object, whether their layout streams are the same or different, provided that matching layout streams apply to the layout object.

Also, the content associated with logical objects which have the same layout stream may be laid out in layout objects generated from different layout object classes, providing that each layout object specifies a matching layout stream.

However, the order in which the basic logical objects are laid out by the layout process is still in accordance with their sequential logical order, irrespective of the stream categories and layout categories that apply.

As the layout is created by sequentially processing the logical objects the layout process maintains a *current layout position*, which identifies an object of the specific layout structure, for each layout stream which occurs.

When the layout process commences, the current layout position of all layout streams is at the root of the specific layout structure.

When laying out the first content of a particular layout stream, and whenever a new layout object is needed for the content of a particular layout stream, then the layout process searches the layout structure for the earliest layout object(s) in sequential layout order which already exist, or which can be created, and to which matching stream category and layout category identifiers apply and that also satisfy other constraints that may be present due to attributes of the logical and layout structures. The search for such layout object(s) starts from the layout object identified by the current layout position of the particular layout stream. When suitable layout object(s) have been identified, the current layout position of the layout stream is moved forward to the lowest level frame into which the content of the layout stream is to be placed.

The layout process processes the basic logical objects of the document in the order in which they occur in the sequential logical order by placing the content of each such object into block(s) within the lowest level frame at the current layout position of the layout stream that applies to that logical object.

Whenever the content of a logical object is split by the layout process into several layout objects, the above search process is repeated for each layout object to contain a content portion. Thus this search is performed for each part of the content associated with a layout stream, with intervening searches for a new current layout position when necessary.

The current layout position of one or more layout streams may also be moved forward in the sequential layout order in order to satisfy a layout directive such as "new layout object". This is described in more detail in 10.4.1.

The current layout position of a layout stream is never moved backwards in the sequential layout order. In cases in which the content of a basic logical object is concatenated to that of another basic logical object which is not its immediate predecessor within the layout stream (for example, where an intermediate object specifies different fill order or content architecture attributes), then some of this content may be laid out within existing layout objects from which the current layout position has previously been moved forward.

However, backtracking to satisfy layout directive attributes such as "balance", "synchronization", "indivisibility" and "same layout object" may cause the layout for the part of the layout structure that is in the scope of the directive and of interacting directives to be reconsidered, moving all the current layout positions that were in this part back to their values at the beginning of the scope of the directive. Each of the cases in which this may occur is explicitly mentioned in the description of the layout process for the layout directive that may cause this backtracking.

Content that is not derived from content associated with basic logical objects (i.e. content specified by the attribute "logical source" or by a content rule of the layout structure) does not form part of the layout streams. This content is laid out when the current layout position of any layout stream passes the layout object in sequential layout order to which the content rule or the attribute "logical source" applies. If necessary, this content is laid out when the end of the layout process is reached.

The layout process places no additional constraints on the layout object class descriptions into which content of layout streams may be placed. Thus the content of logical objects may be laid out in layout objects generated by different layout object class descriptions, provided that each layout object specifies a matching layout stream.

In the case that layout objects specify multiple stream category and/or layout category identifiers then the current layout position of more than one layout stream can identify the same lowest level frame, and the layout process places no additional constraint in these cases, so in such cases content associated with logical objects of different layout streams may be laid out in the same lowest level frame.

If a lowest level frame has no layout stream identified (i.e. attribute values are 'null'), then the layout process treats it as if it had specified all stream categories and all layout categories, in the sense that such a frame will satisfy the search for a new current layout position for any layout stream. Thus the content associated with basic logical objects of any layout stream may be laid out in such a frame.

If a basic logical object has no layout stream identified then it is allocated to a separate implicit layout stream maintained for this purpose; the layout process will lay it out only in lowest level frames for which the implicit layout stream is identified.

The use of different layout streams is illustrated in Figure 14.

In this example, each page contains frames defined to accept layout streams termed 'A' and 'B'. If the next basic logical object has layout stream 'A' then its content will be laid out in the left hand frame of page 3. The content associated with the next basic logical object of layout stream 'B' will be laid out in the right hand frame of page 1.

From this example it can be seen that the document layout process does not close portions of the specific layout structure for further layout when creating a new layout object. That is, the creation of pages 2, 3, etc., and the frames within those pages, does not prevent frame 'B' in page 1 from being used for the layout of subsequent basic logical objects of layout stream 'B'.

Thus frame 'B' in page 1 is the current layout position for laying out the content of the next basic logical object of layout stream 'B'; and frame 'A' in page 3 is the current layout position for basic logical objects of layout stream 'A'.

In subsequent sub-clauses it is shown how some additional attributes impose additional constraints concerning the frame in which a basic logical object is laid out.

Both a layout object class (see 10.3.4) and a layout stream may apply to a basic logical object and in this case both attributes shall be satisfied when carrying out the layout process.

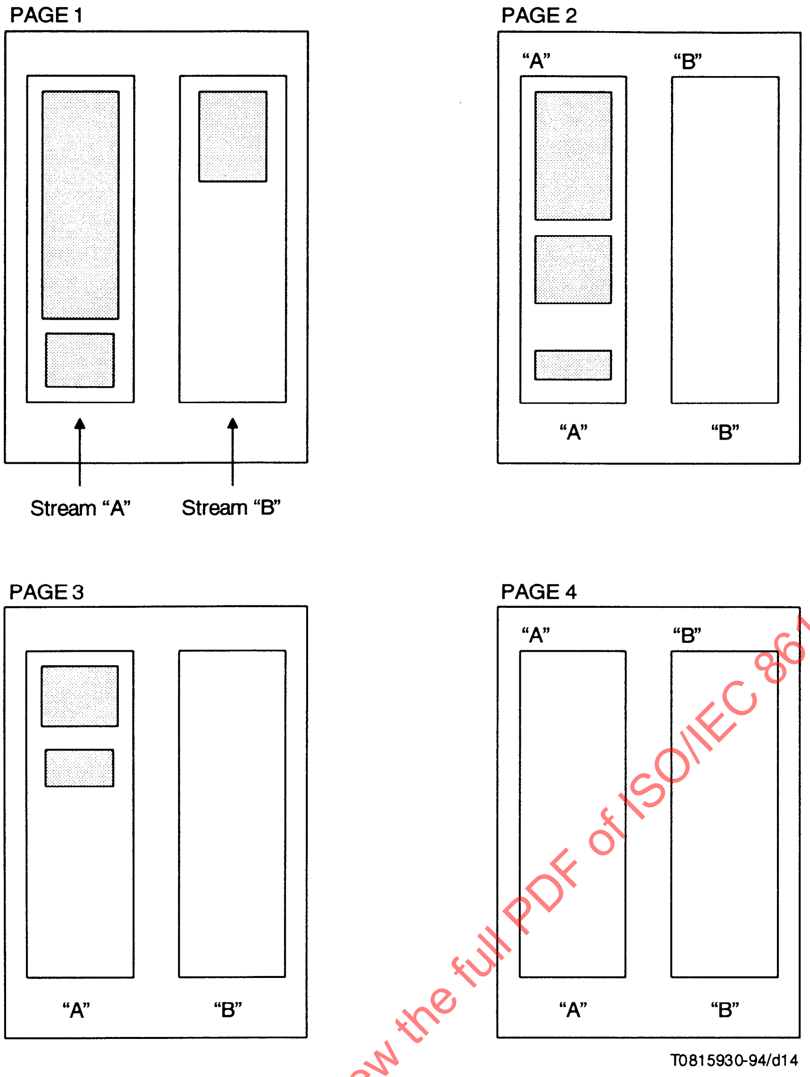


Figure 14 – Illustration of layout streams

10.3.4 Layout object class

The reference to a layout object class from the specific logical structure is specified by the layout directive attribute "layout object class" (see 9.7.8) which identifies a layout object class description.

The purpose of this attribute is to indicate that the content of the logical object, or, if it is a composite logical object, the content of all its subordinate basic logical objects, is to be laid out entirely within a single instance of the specified layout object class. No other part of the document is laid out within the layout object which is created, except for content that results from attributes of the generic layout structure, including "logical source".

If a layout object class has been specified for a basic logical object or any of its superiors, then the constraint specified by the layout object class shall be satisfied when laying out a basic logical object into a lowest level frame.

When a composite logical object specifies a layout object class, then a layout object, of the object class identified by the attribute "layout object class", shall be created by the layout process to contain all the content associated with all the subordinates of the composite logical object.

Thus, when processing a logical object description specifying this attribute, the layout process creates a layout object of the specified layout object class. Layout streams applicable to the logical object, and, if it is a composite logical object, also to all of its subordinates, are valid only within the layout sub-structure subordinate to this layout object. The layout process closes this layout object and all of its subordinates from use for further layout as soon as the logical object that specified the attribute, and all of its subordinates, have been laid out.

This attribute could be used, for example, to indicate that each chapter in a document is to be laid out in an instance of a particular page set.

10.4 The effect of some attributes on the document layout process

In addition to the constraints imposed on the document layout process by the use of layout streams and the use of layout object class, a number of additional constraints are imposed on the layout process by other attributes. This clause describes the effect of some of these attributes, namely:

- new layout object;
- floatability range;
- same layout object;
- indivisibility;
- synchronization;
- balance.

10.4.1 New layout object

The purpose of the attribute "new layout object" (see 9.7.11) is to indicate that the content associated with the logical object to which the attribute applies is to be laid out starting at the beginning of a particular instance of a layout object class. This layout object is one in which no preceding content has been laid out. In the case of a composite logical object, this function applies to all subordinate logical objects.

Examples of use are that a section shall start on a new page, a figure at the beginning of a frame, or a particular paragraph at the top of a column.

In the case of a basic logical object, the layout process determines the current layout position of the layout stream applying to the logical object to which the attribute applies. In the case of a composite logical object, multiple layout streams may apply to the subordinates. In this case, the current layout position is that of the layout stream that is the most advanced in the sequential layout order of all those streams in the logical object for which the attribute "new layout object" applies.

Starting from this current layout position, the layout process determines or creates the next layout object in the sequential layout order of the specified object class, layout category or object type 'page', which does not contain any content associated with any logical objects preceding, in the sequential logical order, the logical object with the content to be laid out.

If the current layout position is found to be in a layout object fulfilling these criteria, no new layout object of the specified layout object class, layout category or object type 'page' need be created. Otherwise the layout process shall create such an object according to the rules of the generic layout structure.

If a layout object is found or created fulfilling the criteria of the attribute "new layout object", the current layout positions for all layout streams applicable to the logical object or its subordinates are moved forward in the sequential layout order to this layout object. These current layout positions are moved before any content belonging to the logical object or its subordinates is laid out. Any layout stream with current layout position within or after the specified layout object is not affected.

In acting on an instance of this attribute, the layout process finds or creates only one layout object of the specified layout object class, layout category or object type 'page'.

As an illustration, consider in Figure 14 the case in which the logical object is in layout stream 'B' and the attribute "new layout object" is specified with a layout object in layout stream 'B' or a layout object of object type 'page': the content of the layout stream 'B' will be laid out in the right hand frame of page 2 or page 4 respectively.

If the attribute has specified an object of the object class corresponding to the right hand frame or to the page, the content of the layout stream 'B' will also be laid out in the right hand frame of page 2 or page 4 respectively.

10.4.2 Floatability range

The attribute "floatability range" (see 9.7.5) is used to control the layout process such that a logical object, for example, representing a floating figure, may be placed only within a specified range.

Examples of use are that a figure shall occur between its reference and the end of the page that ends the subclause that contains the reference, or that a footnote shall start at the earliest suitable location on the page that contains the reference.

The layout process will ensure that all content associated with the logical object identified by “backward limit” is laid out before starting layout of the content associated with the logical object specifying the attribute “floatability range”. If the logical object specified by the parameter “backward limit” is earlier in sequential logical order then the logical object specifying the attribute “floatability range”, then no action may be needed to effect the layout process. If the logical object specified by “backward limit” is later in sequential logical order, then the layout of the logical object specifying the attribute “floatability range” may be deferred.

The layout process will ensure that all content associated with the logical object specifying the attribute “floatability range” is laid out before starting layout of the content associated with the logical object specified by “forward limit”. If the logical object specifying the attribute “floatability range” is earlier in sequential logical order than the logical object specified by the parameter “forward limit” then no action may be needed to effect the layout process.

If this should not be the case, the layout of a number of logical objects may be modified to satisfy the attribute. The reference layout process does not specify any particular algorithms by which the layout may be modified. However, for each layout stream, the sequential layout order shall not be changed, i.e., it shall correspond to the sequential logical order.

10.4.3 Same layout object

The attribute “same layout object” (see 9.7.13) is used to control the layout process such that the content associated with the logical object to which the attribute applies, and the content associated with another specified logical object, are laid out within a particular layout object. It specifies that the earliest basic layout object in the sequential layout order into which content is laid out from the logical object to which this layout directive applies shall be subordinate to the same layout object of a specified layout object class, stream root category, layout category or object type 'page' as the latest basic layout object in the sequential layout order into which content is laid out from the other specified logical object.

The phrase “beginning of content” refers to the first content portion of a basic logical object that is least advanced in the sequential layout order.

The phrase “end of content” refers to the last content portion of a basic logical object that is most advanced in the sequential layout order.

A typical use of this attribute is to control the layout process so that a footnote will start on the same page as its reference.

The other specified logical object shall precede, in the sequential logical order, the logical object for which the attribute applies. If this is not the case the attribute has no effect.

The layout process determines the layout position of the end of the content of the other specified logical object and the current layout position of the layout streams of the beginning of the content of the object for which the attribute applies. If these two layout positions are both within the same layout object of specified object class, stream root category, layout category or object type 'page', the effect of this attribute is fulfilled.

Otherwise, this attribute moves the current layout positions of all layout streams applicable to the logical object or its subordinates for which the attribute applies, forward in the sequential layout order to the layout object specified by the attribute. If the attribute cannot be fulfilled, this is the object where the end of content of the other specified logical object is laid out. These current layout positions are moved before any content belonging to the logical object or its subordinates is laid out. Any layout stream with a current layout position within or after the specified layout object is not affected.

The layout of the content of several logical objects may be modified so as to satisfy this attribute. The reference layout process does not identify any particular algorithms by which that layout may be modified so as to satisfy the constraint specified by this attribute. However, for each layout stream, the sequential layout order shall not be changed, i.e. it shall correspond to the sequential logical order.

10.4.4 Indivisibility

The attribute “indivisibility” is used to control the layout process such that all the content associated with the logical object to which the attribute applies is laid out within a particular layout object. It specifies that this content should be laid out in a single layout object of a specified object class, layout category or object type 'page'.

Typical uses of this attribute are to control the layout process so that a figure and its caption will be laid out on the same page; that a certain paragraph will not be split by a column break or page break; or that two paragraphs will be laid out completely on the same page.

Before any content of the logical object or its subordinates is laid out, the layout process determines if the current layout positions of all layout streams applicable to the logical object or its subordinates are within the same layout object of specified object class, layout category or object type 'page' and if all content that is associated with the logical object (if basic), or with all subordinates of the logical object (if composite), can be laid out completely within that layout object. If this is the case, the effect of this attribute is fulfilled.

If this is not the case, the layout of the content of several logical objects may be modified so as to satisfy this attribute. The reference layout process does not identify any particular algorithms by which the layout may be modified. However, for each layout stream, the sequential layout order shall not be changed, i.e. it shall correspond to the sequential logical order.

If the current layout positions of all content within this logical object are not within a layout object for which the attribute can be fulfilled, this attribute moves the current layout positions for all layout streams applicable to the object or its subordinates forward in the sequential layout order to a layout object specified by the attribute. The current layout positions for the layout streams are moved before any content associated with the logical object or its subordinates is laid out.

10.4.5 Synchronization

The attribute "synchronization" is used to control the layout process such that the logical object to which the attribute applies and another specified logical object have their content laid out in different blocks with trailing edges aligned along a line. It specifies that the earliest block in the sequential layout order containing content of the logical object for which the attribute applies shall be laid out with its trailing edge aligned with the trailing edge of the earliest block in the sequential layout order containing content of the other specified logical object.

A typical use of this attribute is for synchronization of multi-lingual multi-column text, where the start of each paragraph or sub-section is horizontally aligned. Another use is for synchronization of pictures (e.g. mirror pictures) on different pages or columns.

Two conditions are necessary for this attribute to be in effect.

- The other specified logical object must precede in the sequential logical order the logical object for which the attribute applies.
- The two different lowest level frames to which the two blocks involved are subordinate must have the same direction of layout path, i.e. they must have the same value of the attribute "layout path".

The layout process determines if these conditions are fulfilled. If this is not the case, the attribute will be ignored. If this is the case the reference layout process does not identify any particular algorithms by which the layout may be modified. However, for each layout stream the sequential layout order shall not be changed, i.e. it shall correspond to the sequential logical order.

If the content of each of the logical objects to be synchronized is not concatenated to content of previous logical objects, the layout process would normally move one of the two blocks forward in the direction of layout path such that the attribute can be fulfilled.

If the content of one or both of the logical objects is concatenated with content of a previous logical object, then the reference layout model does not specify an exact algorithm for the point at which synchronization occurs.

In the case that one or both of the logical objects have the attribute "fill order" with value 'reverse order', the reference layout process outlined does not specify an exact algorithm for the point at which synchronization occurs.

10.4.6 Balance

The attribute "balance" is used to control the layout process such that content is distributed among a specified set of layout objects such that the leading edge of each of these layout objects shall be approximately aligned.

A typical use of this attribute is for balancing multi-column text at the end of a chapter or section.

The attribute applies only if the layout component to which it applies has (or, for an object class, may have) only composite layout objects as its immediate subordinates. Thus the attribute has no effect if applied to a component description of a lowest level frame.

The layout process determines if all conditions for this attribute are fulfilled (see 9.4.2.1). If this is not the case, the attribute will have no effect.

If the conditions are fulfilled, the layout process does not identify any particular algorithms by which the layout may be modified. However, for each layout stream, the sequential layout order shall not be changed, i.e. it shall correspond to the sequential logical order.

Normally, the layout process will move content among the specified set of layout objects, typically from the layout objects earlier in the sequential layout order to those later in the sequential layout order, in order to "balance" the distribution of content so that their leading edges will be approximately aligned.

10.5 Layout process for frames

In the layout process, the placement of frames within superior layout objects is controlled by means of the layout attributes "position" and "dimensions".

The positioning of frames within their immediately superior layout object is determined in one of two ways, corresponding to the two methods of specifying the attribute "position":

a) a constant position;

b) a position derived from evaluation of a rule.

Frames immediately subordinate to a page shall be specified at constant positions.

The immediate subordinates of any frame are either all positioned at constant positions or all positioned using values derived from evaluation of a rule. The layout of blocks within a frame is described in 10.6. The layout of frames within a superior frame is described in this subclause.

If a frame specifies the attribute "border" then the position and dimensions of immediately subordinate frames are constrained to be such that no part of any of these frames falls within the border region of their immediate superior.

In the case that the attribute "border" applies, then for each edge the border allowance is the sum of the "border line width" and "border freespace width" for the frame. The default value defined in this Specification provides for each edge a border allowance of 0 SMUs. The border allowances for the four edges form the border allowance of the frame, which constrains the area available to the layout process for placement of all immediately subordinate frames.

10.5.1 Placement of frames with fixed position

In the case of frames placed at a fixed position, the value of the attribute "position" in the frame class description is a constant value for both of the sub-parameters "horizontal position" and "vertical position". If either of the sub-parameters "horizontal position" or "vertical position" is not specified then it assumes the default value.

In the case in which such a frame has variable dimensions, the area available to the layout process for the placement of immediately subordinate frames may be further constrained by the specified fixed position.

The dimensions of a fixed position frame are not constrained by other frames subordinate to the same immediately superior layout object.

For a frame for which the attribute "dimensions" specifies the sub-parameter "maximum size", the dimension is re-evaluated at the end of the layout process to fill completely the superior frame within the constraints of the specified fixed position.

10.5.2 Placement of frames with variable position

In the case of placement of frames at a position determined by a rule, the value of the attribute "position" in the frame class description specifies the four sub-parameters "offset", "separation", "alignment" and "fill order". These sub-parameters specify constraints on the placement of the frame which are used to determine uniquely the position of the frame.

In this case, the layout of frames is also controlled by means of the layout attribute "layout path" (see 9.4.2.2) of the immediately superior frame, which specifies a reference direction for the positioning of immediately subordinate frames.

The reference model for the layout process for variably positioned frames is described as a process which assumes that the placement of the variably positioned immediate subordinates of a frame occurs in their sequential layout order.

10.5.2.1 Determination of the area for placement of frames

Within a frame, the *area available for placement* for the placement of immediately subordinate frames is determined by the border allowance of the frame and the sub-parameters "fill order", "offset" and "separation" of the immediately subordinate frames. The area available for placement is a rectangular region within a frame and has edges named as for frames and blocks (see 7.3.3).

In all cases the area available for placement of a frame is constrained to be both within the border allowance and also within the region defined to satisfy the sub-parameter "offset" of the frame to be placed. For each edge, the constraint on the area available for placement within the immediately superior frame is to be no less than the border allowance; and, no less than the offset for the edge, as specified by the relevant one of the sub-sub-parameters: trailing offset, leading offset, left offset and right offset.

Two cases shall be taken into account, as described in 10.5.2.1.1 and 10.5.2.1.2.

10.5.2.1.1 Placement of frames in normal order

If there is no other frame currently within the immediately superior frame, then no additional constraints to those already described (i.e., border and offset) are specified.

If there already exist one or more frames laid out in normal order in the immediately superior frame, then the sub-parameter “separation” is used to constrain additionally the distance between the leading edge of the last laid out frame and the trailing edge of the area available for placement. That distance is constrained to be no less than the maximum of:

- the value of the sub-sub-parameter “leading edge” for the last laid out frame;
- the value of the sub-sub-parameter “trailing edge” for the frame to be laid out.

If there already exist one or more frames laid out in 'reverse order' in the immediately superior frame then the sub-sub-parameter “centre separation” of the sub-parameter “separation” is used to constrain additionally the distance between the leading edge of the area available for placement and the trailing edge of the first of the frames placed in reverse order. The distance is constrained to be no less than the maximum of:

- the value of the sub-sub-parameter “centre separation” for the first frame placed in reverse order;
- the value of the sub-sub-parameter “centre separation” for the frame to be placed;

In all cases, the frame is positioned as close to the trailing edge of the immediately superior frame as is possible under these constraints and those specified in 10.5.2.1.

10.5.2.1.2 Placement of frames in reverse order

For placement of frames in reverse order, the reference model for the layout process is described as a process which operates in three steps:

a) Additional constraints are determined depending on the values of the sub-parameter “fill order” for the immediate subordinates of the immediately superior frame. Determining these constraints may involve temporarily re-positioning some of these subordinate frames.

b) For the purpose of determining the dimensions of the area available for placement, all frames already laid out are temporarily positioned as close to the trailing edge of the immediately superior frame as is possible under these constraints and those defined in 10.5.2.1.

c) Finally, after the frame is laid out, all frames laid out in reverse order are re-positioned as far as possible in the direction of the layout path, without violating the border allowance of the immediately superior frame or the sub-parameters “offset” and “separation” specified for the various frames.

The details of steps a) and b) are:

a) If there is no other frame currently within the immediately superior frame, then no additional constraints to those described in 10.5.2.1 (i.e. border and offset) need to be considered.

b) If, within the immediately superior frame, there exist one or more frames laid out in reverse order but none in normal order then, for the purpose of calculating the size of the area available for placement, the frames already present are temporarily positioned as far as possible in the direction opposite to the layout path, without violating the border allowance or the sub-parameters “offset” and “separation” specified for the various frames. The sub-parameter “separation” is used to constrain the distance between the leading edge of the last laid out frame and the trailing edge of the area available for placement. That distance is constrained to be no less than the maximum of:

   - the value of the sub-sub-parameter “leading edge” for the last laid out frame;
   - the value of the sub-sub-parameter “trailing edge” for the frame to be laid out.

c) If, within the immediately superior frame, there exist one or more frames laid out in normal order and none laid out in reverse order, then the sub-parameter “separation” is used to constrain the distance between the leading edge of the last laid out frame and the area available for placement. The distance is constrained to be no less than the maximum of:

   - the value of the parameter “centre separation” for the last frame laid out in normal order;
   - the value of the parameter “centre separation” for the frame to be laid out.

d) If, within the immediately superior frame, there exist one or more frames laid out in normal order and one or more laid out in reverse order, then for the purpose of calculating the size of the area available for placement, those frames that were laid out in reverse order are temporarily positioned as far as possible in the direction opposite to the layout path, without violating the border allowance or the sub-parameters “offset” and “separation” specified for the various frames.

In particular, without violating the sub-sub-parameter “centre separation” specified for the first frame laid out in reverse direction and for the last frame laid out in normal order, that is, these frames are separated by a distance constrained to be no less than the maximum of these sub-sub-parameters. Thereafter the area available for placement is determined as described above.

10.5.2.2 Determination of the dimensions for variably positioned frames

The dimensions of a frame within the area available for placement is determined from the attribute “dimensions”.

The dimensions of a frame are said to be *tentatively determined* when they are determined subject to existing constraints but may be modified as a result of further constraints.

The dimensions of a variably dimensioned frame are tentatively determined whenever a constraint imposed on the position or dimension attributes of a different frame that is not subordinate to that frame have to be evaluated.

The dimensions of a variably dimensioned frame are tentatively determined whenever position or dimension attributes, or constraints on these of layout objects subordinate to this frame, have to be evaluated.

The dimensions of a variably dimensioned frame are permanently determined when a following frame receives content within the same immediately superior frame with the same value of the sub-parameter “fill order”. Subsequently, further content may be laid out into the frame but the frame dimensions cannot be changed.

Alternatively, the dimensions are permanently determined when there is no more content to be laid out in the frame.

For a frame for which the attribute “dimensions” specifies the sub-parameter “maximum size”, the dimensions and the positions of all other frames immediately subordinate to that frame are re-evaluated at the end of the layout process to fill the frame completely within the constraints of the parameters of the attribute “position”. If more than one variably positioned frame within a superior frame specifies “maximum size” and could be expanded in the direction of the layout path of the superior frame, the free space is evenly distributed between these frames, unless one or more of the frames cannot be expanded evenly for other reasons such as synchronization, in which case these restricted frames are expanded as much as possible and then the remaining frames are expanded evenly.

10.5.2.3 Alignment of variably positioned frames

When the dimensions both of a frame and of its immediately superior frame either have been permanently determined or are specified by the sub-parameter “fixed dimension” of the attribute “dimension”, the frame is aligned according to the sub-parameter “alignment” of the parameter “variable position” within the area available for placement, subject to the constraints specified by the sub-sub-parameters “right-hand offset” and “left-hand offset” of the sub-parameter “offset”.

10.6 Allocation of areas for blocks

The content is laid into blocks within frames by the layout process. The blocks are laid out within an available area within a lowest-level frame, which is determined by attributes including “layout path”, “fill order”, “offset”, “separation”, “border” and “concatenation” (see 9.4.2.2, 9.7.4, 9.7.12, 9.7.14, 9.4.1.3 and 9.7.3, respectively).

The content layout process described by each content architecture determines the exact dimensions of blocks within the available area. The structure within a block is also determined by the content architecture.

10.6.1 Determination of the available area

The available area within a lowest-level frame for the creation of a block to lay out the content is determined by the attributes “layout path”, “fill order”, “offset”, “separation”, “concatenation” and any previously laid out blocks within the frame.

In the case that the lowest level frame is specified to have variable dimensions with a size determined by the content (i.e. by the use of the sub-parameters “Rule A” “ or “Rule B”, see 9.4.1.2), then the constraints on the dimensions of the available area are to be transferred to the content layout process during the layout process. In this case, the size of the available area is maximised within the constraints given (which may be derived from the sub-sub-parameters “minimum dimensions”, “maximum dimensions” of the attribute “dimensions” of the frame and by the constraint on the frame dimensions specified by its immediately superior frame or page, see 9.4.1.2).

The attribute “border”, which applies to frames and blocks, can also affect the available area. If the attribute “border” is specified for the frame involved, or in an applicable default value list for layout objects of object type 'block', or in a presentation style associated to a basic logical or layout component, then either or both of the frame and block borders have to be taken into account and may reduce the available area.

A frame has a border inside it, and, for each of its edges specified by the attribute “border”, the values of the parameters “border line width” and “border freespace width” constrain the available area by deriving a border allowance.

A block has a border on the outside, and, for each of the edges specified by the attribute “border”, the values of the parameters “border line width” and “border freespace width” constrain the available area further by deriving a further border allowance.

The border allowance is, for each edge, the sum of the constraints derived from the attribute “border” applicable to the frame and derived from the attribute “border” applicable to the block. If, for one or more of the edges, the default value defined in this Specification applies for both the frame and for the block, then for these edges the border allowance is 0 SMUs.

In all cases, the available area is constrained to be within the border allowance and also to be within the region defined to satisfy the attribute “offset”.

Thus, for each edge, the greater of the border allowance and the offset for that edge, specified by the relevant one of the parameters “trailing offset”, “leading offset”, “left offset”, and “right offset”, constrain the available area within the lowest level frame.

Depending on the value of the attribute “fill order”, two cases need to be taken into account when determining the available area and laying out blocks within the available area. Its permissible values are 'normal order' and 'reverse order', these are related to the direction specified by the attribute “layout path” of the lowest level frame.

The two cases to be taken into account are as follows.

10.6.1.1 Layout of blocks in normal order

If there is no other block in the immediately superior page or frame, then only the attribute “offset” and the border allowance need to be considered.

If there exist one or more blocks laid out in normal order but none in reverse order, then the attribute “separation” is used to constrain further the distance between the leading edge of the last laid out block and the available area. That distance is constrained to be no less than the maximum of:

- the value of the parameter “leading edge” for the first logical object with content in the last laid out block;
- the value of the parameter “trailing edge” for the logical object to be laid out;
- the sum of the border allowances for the leading edge of the last laid out block and for the trailing edge of the block to be laid out.

In all the other directions the attribute “offset” and the border allowance are used as described in 10.6.1.

The block is positioned as close to the trailing edge of the lowest level frame as is possible under those constraints.

If there exist one or more blocks laid out in reverse order, then the parameter “centre separation” of the attribute “separation” is used to constrain further the distance between the available area and the trailing edge of the first of the blocks laid out in reverse order (which is the closest block laid out in reverse order). That distance is constrained to be greater than the maximum of:

- the value of the parameter “centre separation” for the first logical object with content in the first of the blocks laid out in reverse order;
- the value of the parameter “centre separation” for the logical object to be laid out;
- the sum of the border allowances for the trailing edge of the first of the blocks laid out in reverse order and for the leading edge of the block to be laid out.

The preceding description specifies how the available area is determined when a new block is to be created. This is the case, when concatenation is not in effect. When concatenation is in effect, then the content is continued in an already created block, with or without fixed dimensions, and all constraints are derived from those attributes applicable to the concatenated sequence of components with content in the block.

10.6.1.2 Layout of blocks in reverse order

If there is no other block in the immediately superior page or frame, then only the attribute “offset” and the border allowance need to be considered.

If there exist one or more blocks laid out in reverse order but none in normal order, then, for the purpose of calculating the available area, the blocks already present are temporarily positioned as far as possible in the direction opposite to the layout path, without violating the attribute “offset” or the border allowance specified for the first logical object with content in the first block laid out in reverse order. The attribute “separation” is used to constrain further the distance between the leading edge of the last laid out block and the available area. That distance is constrained to be no less than the maximum of:

- the value of the parameter "leading edge" for the first logical object with content in the last laid out block;
- the value of the parameter "trailing edge" for the logical object to be laid out;
- the sum of the border allowances for the leading edge of the last laid out block and for the trailing edge of the block to be laid out.

In all the other directions the attribute "offset" and the border allowance are used as described in 10.6.1.

The block is positioned as close to the leading edge of the lowest level frame as is possible under those constraints.

If there exist one or more blocks laid out in normal order and none laid out in reverse order, then the attribute "separation" and the border allowance are used to constrain further the distance between the leading edge of the last laid out block and the available area. That distance is constrained to be greater than the maximum of:

- the value of the parameter "centre separation" for the first logical object with content in the last of the blocks laid out in normal order;
- the value of the parameter "centre separation" for the logical object to be laid out;
- the sum of the border allowances for the leading edge of the last of the blocks laid out in normal order and for the trailing edge of the block to be laid out.

In all the other directions the attribute "offset" and the border allowance are used as described in 10.6.1.

The block is positioned as close to the leading edge of the lowest level frame as is possible under those constraints.

If there exist one or more blocks laid out in normal order and one or more blocks laid out in reverse order, then, for the purpose of calculating the size of the available area, the blocks laid out in reverse order are temporarily positioned as far as possible in the direction opposite to the layout path. This is done without violating the border allowance, or the parameter "centre separation" of the attribute "separation" specified for the logical object of the first block laid out in reverse order, and for the first logical object with content in the last block laid out in normal order. Thus the blocks are separated by a distance equal to the maximum of those parameters or the combined border allowances. Thereafter the available area is further constrained as described in 10.6.1.

Finally, after the block dimensions have been determined, all blocks laid out in reverse order are positioned as far as possible in the direction of the layout path, without violating the border allowances and the attributes "offset" and "separation" applicable to the various blocks.

The preceding description specifies how the available area is determined when a new block is to be created. This is the case, when concatenation is not in effect. When concatenation is in effect, then the content is continued in an already created block, with or without fixed dimensions, and all constraints are derived from those attributes applicable to the concatenated sequence of components with content in the block.

10.6.1.3 Block alignment orthogonal to the layout path

In the direction orthogonal to the layout path, the block position in the available area is defined by the value of the attribute "block alignment" of the first logical object with content in the block. The value may be 'right aligned', 'left aligned', or 'centred' within the available area in the direction orthogonal to the layout path. If the attribute "block alignment" specifies a value of 'null' then the reference layout process does not define an alignment in the direction orthogonal to the layout path.

10.7 Alternative representation

If an alternative representation is specified, the circumstances under which it will be used are determined by particular implementations and are not defined in this Specification.

If the alternative representation is used, then all the layout directive attributes specified for the basic object continue to apply. The character string in the attribute "alternative representation" is treated by the layout process as if the string had been specified in the attribute "content information"; (see ITU-T Rec. T.416 | ISO/IEC 8613-6). The character set is specified by the document profile attribute "alternative representation character sets" (see ITU-T Rec. T.414 | ISO/IEC 8613-4).

11 Reference model of the document imaging process

This clause provides a description of the document imaging process as applicable to documents of the formatted document architecture class or the formatted processable document architecture class (see 6.3.13).

Such documents include constituents representing a specific layout structure and may optionally include constituents representing a generic layout structure and/or presentation styles. In the case of the formatted processable document architecture class, other constituents are present but these do not affect the imaging process.

The purpose of the reference model specified in this clause is to aid the understanding of the semantics of the attributes affecting the presentation of the document structure. It provides additional normative information on the semantics of the attributes defined in clause 9, but the reference model is not intended to specify any process that might be carried out in a particular implementation.

The content imaging process, which controls the imaging of content portions within basic layout objects, is not described here, it is included in the specification of individual content architectures.

11.1 Imaging order

The *imaging order* determines the precedence of layout objects for imaging in the layout object to which they are immediately subordinate. This order constrains how the image of the document is resolved for displaying on the presentation surface (see 11.2).

If necessary, the layout structure is derived from the set of descriptions including alternative descriptions present in the document by a process called initialization; this process is performed analogously to the initialization of the layout process (see 10.1.2).

The imaging order of layout objects is determined by the imaging order applicable to their lowest common superior layout object. In particular, the imaging order of layout objects that are immediately subordinate to the same layout object is determined by the imaging order of that common superior object.

The imaging order is determined by the attribute “imaging order”. If this attribute is not specified, the imaging order is determined by the sequential layout order. Thus the imaging order for all layout objects can be uniquely determined.

11.2 Intersection principles

Pages are independent units for presentation, and do not intersect. Within a page, frames and blocks may be positioned in such a way that they intersect partially or fully, so as to share common areas. In all cases, subordinates are fully contained within their superiors (see 7.3.1).

A page or frame can be considered as an area which carries within its surface other areas representing its immediately subordinate objects, which may be frames or blocks. Similarly, a block can be considered as an area on which content is placed.

These areas have a texture which is described by two attributes: “colour” and “transparency”:

The ITU-T Rec. T.410-Series | ISO/IEC 8613 specifies that:

- “colour” is one of 'colour of media' or 'coloured';
- “transparency” is one of 'transparent' or 'opaque';
- the texture of pages, frames and blocks is one of:
  - coloured, paque,
  - colour of media, transparent,
  - colour of media, opaque.

The texture 'colour of media', 'opaque' is intended for hard copy; it allows the colour of page, frame and block areas to be that of the media.

For pages, 'colour of media', 'transparent' and 'colour of media', 'opaque' are equivalent (since pages cannot overlay any other layout objects).

When frames or blocks intersect, their intersection is governed by the following rules:

- Layout objects later in the imaging order overlay layout objects earlier in the imaging order.
- If an opaque layout object overlays other layout object(s), any content or texture of the underlying object(s), and their subordinates, is not imaged in the area of intersection.
- If a transparent layout object overlays other layout object(s), then the image of this layout object and the layout object(s) which it overlays are imaged superimposed in the area(s) of intersection. Content in the area(s) of intersection is combined.
- The border line and border free space are considered to be an extension of the block area when present. In particular, the border freespace has the same texture as the block.

11.3 General rules for positioning pages on presentation surfaces

11.3.1 Nominal page and assured reproduction areas

The page is intended to be positioned and imaged on a unit of the presentation surface. The ideal size of the presentation surface, as assumed by the originator of the document, is a rectangular area called the *nominal page*.

Thus, the layout object "page" is positioned on a single nominal page. The dimensions of the nominal page are specified by the attribute "medium type".

The nominal page is equal to the ideal paper size (see, for example, ISO 216). Hard-copy devices have to allow for the possibility of edge losses caused, for example, by gripping losses for paper feeding, paper size tolerances, skew, etc. In order to cater for these edge losses, an assured reproduction area is defined which is the rectangular area that remains on the nominal page after deducting an agreed allowance for edge losses.

11.3.2 Positioning of the page

The position of the layout object page relative to the nominal page is specified by means of an orthogonal coordinate system. The origin of this coordinate system is at the top left corner of the nominal page. The horizontal axis corresponds to the top edge and the vertical axis corresponds to the left edge of the nominal page as shown in Figure 15. Horizontal positions are measured positively from the vertical axis to the right and vertical positions are measured positively from the horizontal axis downwards.

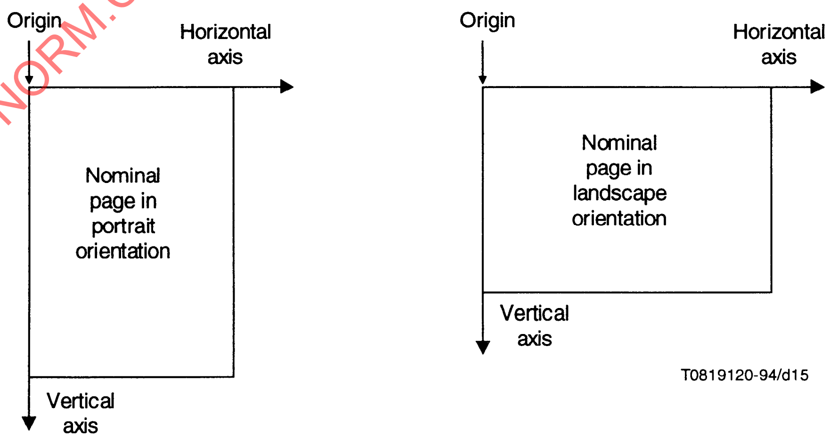


Figure 15 – Nominal page co-ordinate system and orientation

If the horizontal dimension is greater than the vertical dimension then the nominal page is in landscape orientation, otherwise it is in portrait orientation (see Figure 15). The landscape or portrait orientation places no constraint on the orientation of the content on the page.

For example, although the nominal page may be specified to be in portrait orientation, the graphic elements within the page may be rotated so that the nominal page is intended to be viewed in landscape orientation.

The reference point for the positioning of a layout object page is the top left corner of that page. The position of the page reference point relative to the top left corner of the nominal page is specified by the attribute "page position".

The size of the nominal page and positions on the nominal page are specified as integral multiples of the scaled measurement unit.

If the attribute "page position" is specified, then the position of the layout object page relative to the nominal page is completely determined.

In the case when nominal page and assured reproduction areas are defined by this Specification, then, if the attribute "page position" is not specified, the following rules apply to the positioning of the page:

- If each dimension of the page is equal to or less than those of the assured reproduction area, then the page shall be positioned such that its reference point is coincident with the top left corner of the assured reproduction area.

- If either or both of the page dimensions are larger than those of the assured reproduction area but smaller than those of the nominal page area, then the page shall be positioned on the nominal page such that the possibility of information loss is minimized.

- If the page dimensions are equal to those of the nominal page, then the page shall be positioned such that its reference point is coincident with the top left corner of the nominal page.

- If either or both of the page dimensions are larger than those of the nominal page, then the page shall be positioned relative to the nominal page such that the possibility of information loss is minimized.

In all four cases, it is intended that the page be positioned such that its edges are parallel to the edges of the assured reproduction area.

In the case that this Specification does not define nominal page and assured reproduction areas, then the rules that apply to the positioning of the page if the attribute "page position" is not specified are implementation dependant.

Although the provisions of this Specification would, in principle, permit text to be positioned anywhere on the nominal page, the originator shall rely on text reproduction by the recipient only within the assured reproduction area.

11.3.3 Definition of assured reproduction area

The assured reproduction areas for ISO A4, North American letter, ISO A3, Japanese legal and Japanese letter paper sizes are defined as shown in Figures 16, 17, 18, 19 and 20, respectively.

For landscape orientation, the assured reproduction area for each paper size is derived from Figures 16 to 20 by rotating the figures such that the bottom left corners of these figures are positioned as the top left corners in landscape orientation, that is the figures are rotated 90° clockwise.

The dimensions in these figures are expressed in basic measurement units (BMUs), (see 7.3.4.1).

For other page sizes, the dimensions of the assured reproduction area and its position on the nominal page shall be defined in a document application profile, if one is referenced by the document. If no document application profile is referenced, then an assured reproduction area is not defined and the defaulting of page position is implementation dependant.

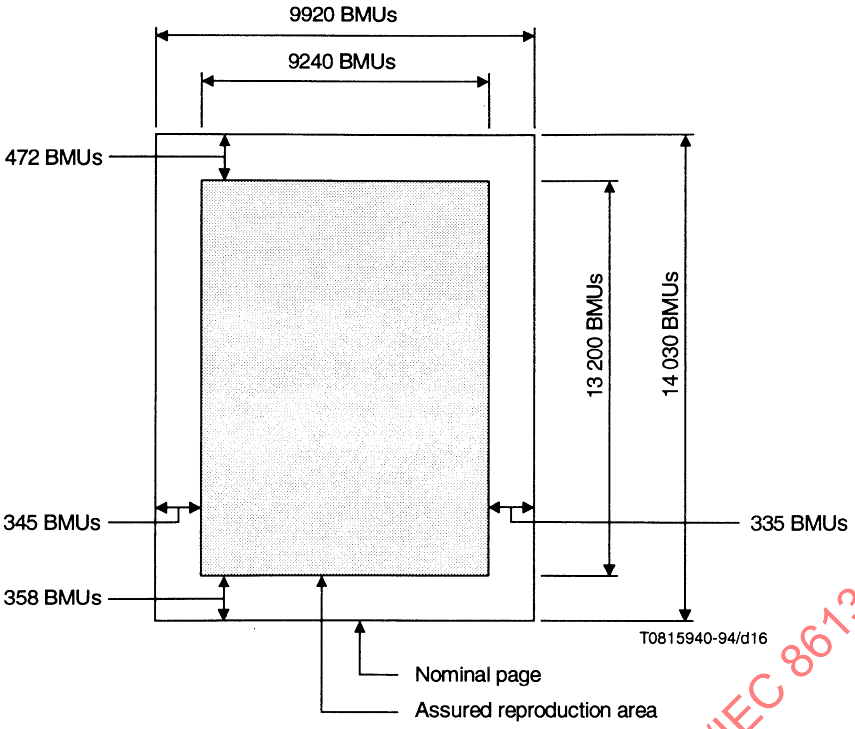


Figure 16 – Dimensions and assured reproduction areas for ISO A4 page size

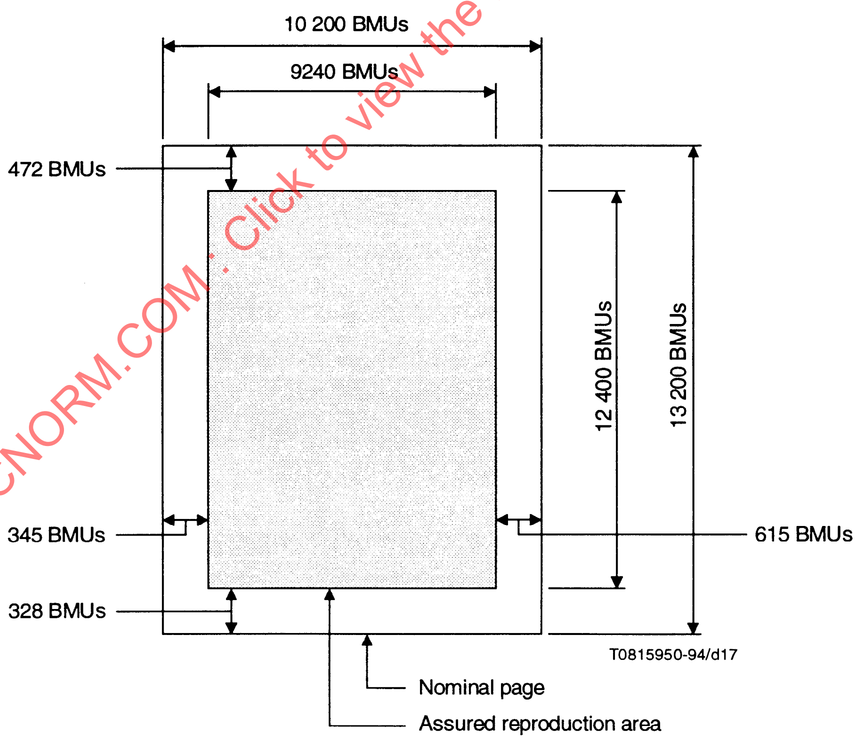


Figure 17 – Dimensions and assured reproduction areas for North American letter page size

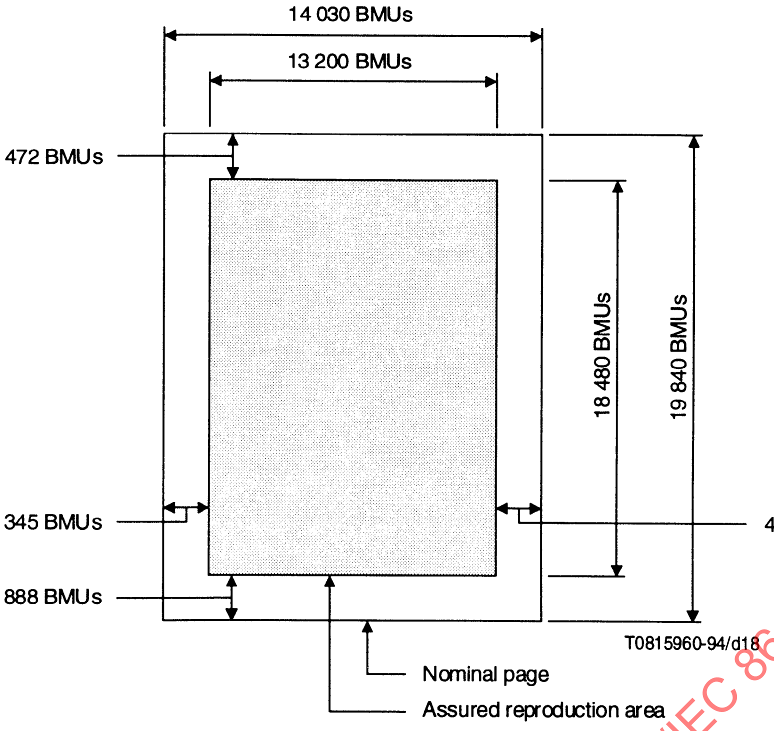


Figure 18 – Dimensions and assured reproduction areas for ISO A3 page size

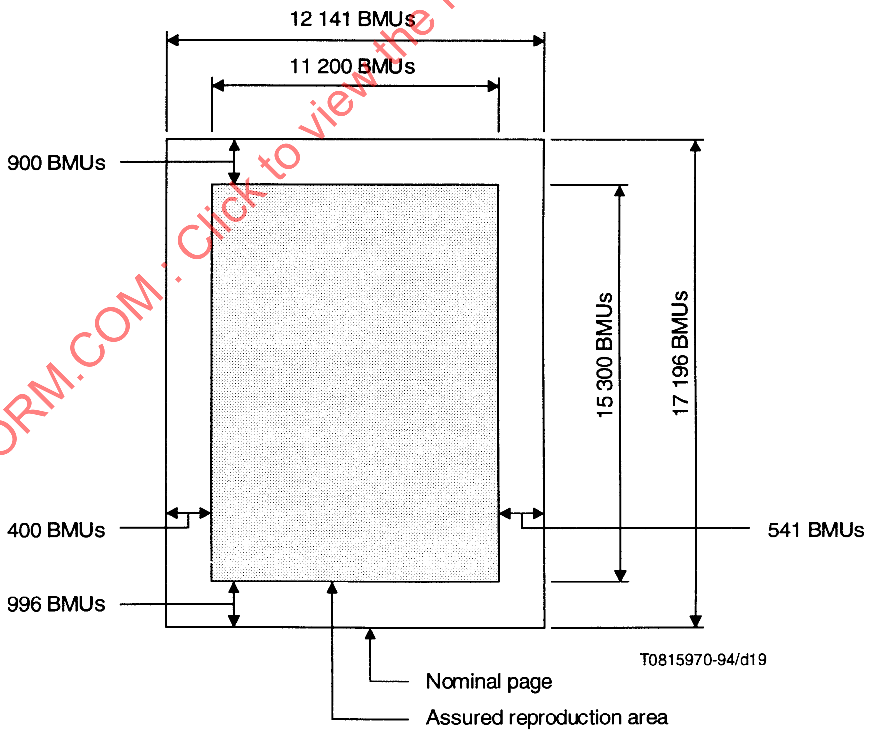


Figure 19 – Dimensions and assured reproduction areas for Japanese legal page size

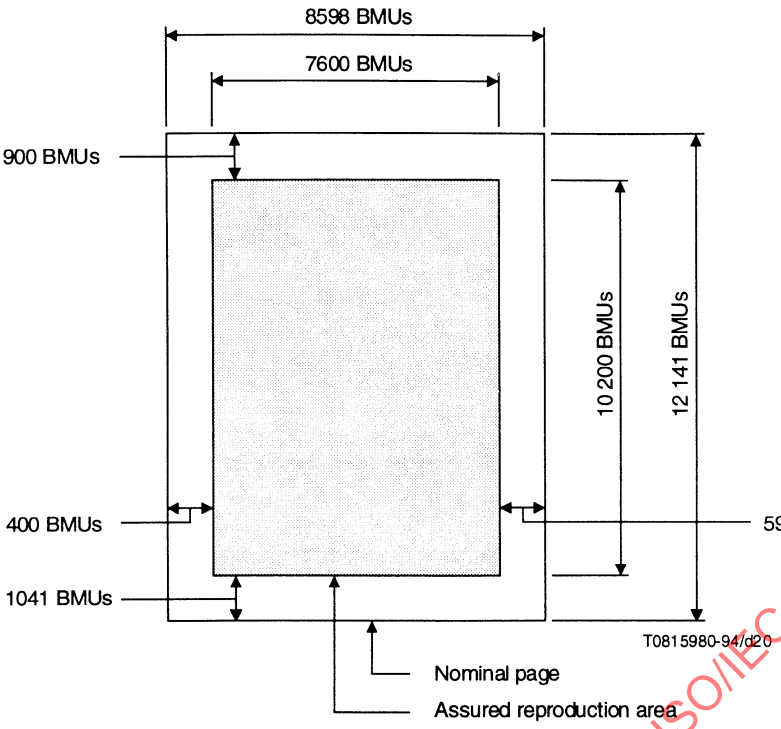


Figure 20 – Dimensions and assured reproduction areas for Japanese letter page size

11.3.4 Recto/verso pages

The originator can specify that a page is to be imaged on a particular side of a sheet of a hard copy medium.

The nominal page is specified to be imaged on either the 'recto' side, the 'verso' side or on an 'unspecified' side of this medium.

When a document consisting of 'recto' and 'verso' pages is opened, the 'verso' side of a sheet and the 'recto' side of the following sheet are simultaneously visible.

11.3.5 Positioning of pages on soft copy media

For positioning of pages on soft copy media, the specific characteristics of the soft copy device have to be taken into account. Such devices do not require the concept of nominal page or of assured reproduction area.

The physical screen may be organised either to image the complete page or a part of it; such a part is called a window. Mapping complete pages or windows onto the screen is a local operation which can be achieved in different ways; therefore, this mapping is not defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613.

12 Reference model for protecting parts of a document

This clause provides a description of an abstract reference model for protecting parts of a document.

The purpose of this clause is to aid the understanding of the semantics of the attributes related to the different aspects of security in the ITU-T Rec. T.410-Series | ISO/IEC 8613. It does not imply an actual implementation or definition of a standardized process.

The document security processing consists of two phases. One phase enciphers or seals parts of the document and creates security information that is added to the document. The other phase makes use of security information in the document for deciphering a part of the document or checking a seal of a part of the document. These phases may occur during several stages of document processing, eg. during the editing process, before the layout process or after the layout process.

The description of the document security model is made in two steps: firstly, an overall model covering the interchange of a document between an originator and a recipient (see 12.1); secondly, covering the local systems of the originator and the recipient (see 12.2).

The local system is here defined as those parts of a system for interchanging documents on which the originator or the recipient has a direct influence, i.e. while preparing the document resulting in a valid data stream on the originator's side and after receiving an appropriate data stream on the recipient's side.

The rest of the system consists of parts that are responsible for the actual transfer of the document and those security facilities that implement the security policy of the security domains to which the originator and the recipient belong.

A more detailed description is found in Annex G.

12.1 The overall model

Throughout the following the distinction is drawn between the handling of the complete document by the system and its security facilities and mechanisms, and the handling of specified parts of the document in the possession of the user, an originator or a recipient.

The processes used for the preparation of the data stream belong to the local system of the originator.

The processes used for handling the received data stream belong to the local system of the recipient.

The two local systems are assumed to be able to provide and utilize the security information described here concerning the parts of the document.

The local system may generate information concerning the handling of the complete document by a security facility outside the local system, but this is advisory. This information, an ODA security label, will be interpreted by the security facility in the context of the security policy in force in the security domain to which the originator belongs.

12.2 The local system

The model of the local system describes the security processes involved and their relationships to the three processes (editing process, layout process and imaging process) described in the document processing model (see 6.4).

The local system of the originator prepares the document, including the interchange of security information intended either for the recipient or for the security facility of the security domain of the originator (see Figure 2).

Those aspects of the security information intended for the recipient are dealt with by the local system of the recipient and deal only with the security of protected parts of the document.

Those aspects of the security information intended for the security facility of the security domain of the originator not belonging to the local system are specified in the ODA security label of the document profile, and this facility will handle the document according to the security policy in force. It can only handle the document only as a single unit, i.e. the whole of the document.

The originator can:

- encipher certain parts of the document in order to provide confidentiality, i.e. encipherment;
- provide a seal which allows a privileged recipient to perform checks for:
  - content integrity;
  - origin authenticity;
  - non-repudiation of origin.

NOTE – Sealing has no influence on the content itself, whereas encipherment will change the content.

A privileged recipient can:

- decipher enciphered parts of the document;
- perform a check on content integrity;
- perform a check on origin authenticity;
- perform a check on non-repudiation of origin.

Security protection can be applied to a document in either processable, formatted processable or formatted form. In other words, the security protection can be performed either before, after or both before and after the layout process. Depending on which form the security protection is applied to, the protection will be different.

Sealing of a document before or after the layout process is called pre-sealing and post-sealing, respectively.

Sealing has no impact on the layout process or the imaging process.

A seal that has been made on a document can be checked only when the document is in the same form as when the seal was made.

Encipherment of a document before or after the layout process has quite different effects. Pre-encipherment is the term used for encipherment of parts of a document before the layout process and post-encipherment when it is performed after the layout process.

Pre-encipherment of a document in processable form will result in a pre-enciphered processable document interchange format.

A layout process will ignore all pre-enciphered parts in a pre-enciphered processable form document. The created layout structure will thus have no knowledge or indication of the existence of any pre-enciphered parts.

A document in processable form or pre-enciphered processable form can serve as input to a layout process. This process will result in one of the forms: formatted processable form, formatted form, pre-enciphered formatted processable form, or pre-enciphered formatted form.

These four forms can be post-enciphered, resulting in the four forms: post-enciphered formatted processable form, post-enciphered formatted form, pre- and post-enciphered formatted processable form, or pre- and post-enciphered formatted form.

The imaging process will ignore all pre- and post-enciphered parts of the document. But since the size and position of a post-enciphered layout object are specified explicitly in the specific layout structure, the post-enciphered parts within the laid out areas will not be imaged.

The imaging process receiving any of these documents will present them such that:

- all clear text parts will be imaged;
- the pre-enciphered parts will be completely lost in the imaging process;
- the post-enciphered parts will have claimed areas of correct dimensions, but their content is not imaged.

All combinations of protection can be applied to a document but not all combinations are possible for an individual part of the document.

13 Document architecture classes

13.1 Definition of document architecture classes

This clause defines the document architecture classes that may be used in particular applications using the ITU-T Rec. T410-Series | ISO/IEC 8613. It also defines additional normative information on the semantics of the attributes defined in clause 9.

Three document architecture classes are distinguished:

- formatted document architecture class;
- processable document architecture class;
- formatted processable document architecture class.

Each document architecture class defines

- the structures that must, or may optionally, be used in documents that pertain to that document architecture class;
- the types of style permitted in documents that pertain to that document architecture class;
- for each structure, the attributes and attribute values that are applicable to the objects in that structure.

For each document architecture class, there is no restriction on the object types within those structures that can be used in documents that pertain to that class. However, a document application profile may impose restrictions concerning which object types can be used.

NOTE – Document application profiles are described further in the ITU-T Rec. T.411 | ISO/IEC 8613-1.

For each object type, there is a minimum set of attributes that shall be supported by all document application profiles allowing the use of components of that object type. These minimum sets are defined in 13.3.

The document application profiles specify which other attributes may be used.

13.2 The constituents of document architecture classes

The structures that shall and that may optionally be present in documents that pertain to the three document classes are defined in Table 2.

Table 2 – Document architecture classes

Document architecture class	Generic logical structure	Specific logical structure	Generic layout structure	Specific layout structure	Layout styles	Presentation styles
FDA	not permitted	not permitted	optional (partial, factor only)	shall be present	not permitted	optional
PDA	optional	shall be present	optional (complete only)	not permitted	optional	optional
FPDA	optional	shall be present	shall be present (complete only)	shall be present	optional	optional

FDA Formatted document architecture class
PDA Processable document architecture class
FPDA Formatted processable document architecture class

For each class, some structures shall be present, some are optional and some are not permitted. A structure that shall be present, shall be present in any document of that class. An optional structure may but need not occur in a document of that class. In the case of optional structures, it is the responsibility of each document application profile to define whether that structure shall be used, may be used or shall not be used in documents of the class conforming to that document application profile. A structure that is not permitted shall not be present in any document of that class.

The structures which are present in any particular document, and whether generic structures are complete generator sets, partial generator sets or factor sets, is specified by attributes of the document profile.

Processable and formatted processable document architecture classes permit the use of layout styles. Presentation styles are permitted in any of the three document architecture classes. Within these constraints it is the responsibility of each document application profile to define the particular styles that may be used in documents conforming to that document application profile.

13.3 Minimum attribute sets

13.3.1 Minimum attribute sets applicable to logical component descriptions

The minimum set of attributes applicable to logical objects is:

a) For all object types:

- object type,
- object identifier,
- object class.

b) In addition, in the case of objects of object type document logical root or composite logical object:

- subordinates.

c) In addition, in the case of objects of object type basic logical object:

- content portions;
- content architecture class.

The minimum set of attributes applicable to logical object classes is:

a) For all object types:

- object type;
- object class identifier.

b) In addition, in the case of object classes of object type basic logical object:

- content portions;
- content architecture class.

13.3.2 Minimum attribute sets applicable to layout component descriptions

The minimum set of attributes applicable to layout objects is:

a) For all object types:

- object type;
- object identifier;
- object class.

b) In addition, in the case of object types frame or block:

- position;
- dimensions.

c) In addition, in the case of object types basic page or block:

- content portions;
- content architecture class.

The minimum set of attributes applicable to layout object classes is:

a) For all object types:

- object type;
- object class identifier.

b) In addition, in the case of object classes of object type frame or block:

- position;
- dimensions.

c) In addition, in the case of object classes of object type basic page or block:

- content portions;
- content architecture class.

13.4 Constraints on attributes

The following attributes shall not to be used in documents conforming to formatted document architecture class:

- layout path;
- logical source;
- layout stream categories;
- layout stream sub-categories;
- permitted categories.

The following attributes are permitted to be specified for constituents of the specific layout structure; however no semantics of their application in this case is defined in this Specification:

- layout path;
- bindings;
- balance;
- layout stream categories;
- layout stream sub-categories;
- permitted categories.

Annex A

Notation used to represent document structures

(This annex forms an integral part of this Recommendation | International Standard)

A.1 Notation for structure diagrams

The notation described in this annex is intended to be an aid to illustrating document structures. In this method, structures are illustrated in the form of structure diagrams, in which each component is represented by a rectangular box. The document root is placed at the top of the diagram and subsequent hierarchical levels in the structure are added by progressing from top to bottom. Lines joining the components represent the division of components into their immediately subordinate components.

In the case of diagrams representing generic structures, one of three mnemonic symbols may be placed below each box to indicate how the immediately subordinate objects are to be generated. This symbol indicates the type of construction expression associated with the object class represented by the box and contained in the attribute "generator for subordinates". These mnemonic symbols are:

- SEQ – Indicates a sequence construction: the immediately subordinate objects are to be generated in order from left to right as written.
- AGG – Indicates an aggregate construction: the immediately subordinate objects may be generated in any order.
- CHO – Indicates a choice construction: only one object may be chosen to form the immediately subordinate object.

In addition, one or two mnemonic symbols may be placed against a branch to indicate how many times the object, or group of objects, at that branch may occur.

These mnemonic symbols are:

- OPT – Indicates that an object, or group of objects, is optional; when this symbol is used on its own, it indicates that an object, or group of objects, can occur zero or one time only.
- REP – Indicates that an object, or group of objects,may be repeated; when this symbol is used on its own, it indicates that an object, or group of objects, is to occur one or more times.
- OPT REP – The use of the symbols together indicates that an object, or group of objects, can occur zero, one or more times.

The absence of any of these symbols indicates that the object shall occur once, and once only.

In the case of specific structures, these mnemonic symbols are not used. The diagrams indicate specifically the occurrence of each object in the structure, in the order in which they are specified by the attribute "subordinates".

A dashed rectangle called a *connector* may be used to indicate where subtrees are to be added to the structure. For example, subtrees may be illustrated elsewhere in order to simplify the main structure.

In the example below, the subtree shown in Figure A.2 is intended to be added in the main structure in Figure A.1 at the point indicated by CONNECTOR Z.

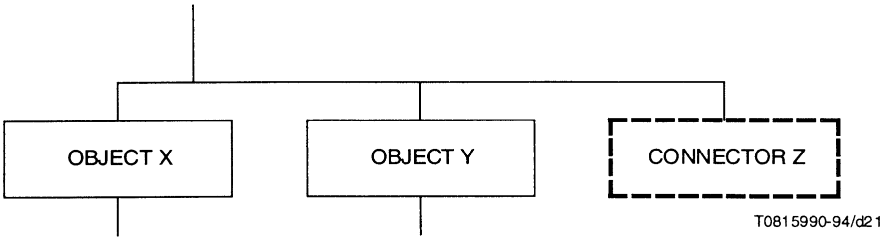


Figure A.1 – Example of document structure notation – Main structure

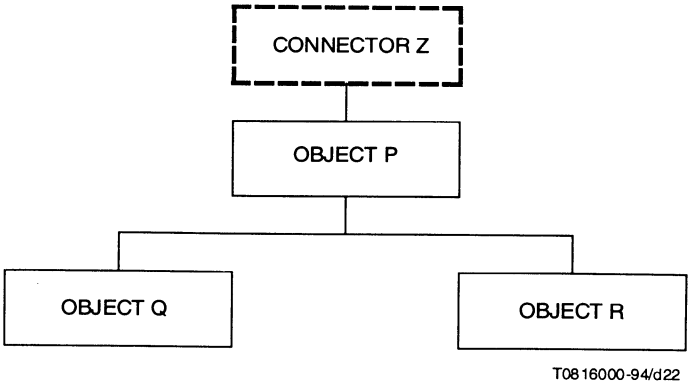
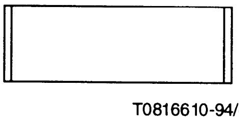


Figure A.2 – Example of document structure notation – Subtree

The symbol:



is used in a specific structure to indicate a content portion and in a generic structure to indicate a generic content portion.

Each box contains a name to identify the component. These names could correspond to the names of components contained within the attribute "user-visible name" and might be used in a user's application to process a document.

In the case of layout components, the names DOCUMENT LAYOUT ROOT, PAGE-SET, PAGE, FRAME and BLOCK are used to specify the layout object types. The distinction between composite and basic logical objects need not be indicated on the diagrams since this should be obvious in the majority of cases.

A.2 Notation for expressions

This subclause describes a notation for specifying various types of expression in human readable form.

This notation can be used, for example, in describing example documents or in the specification of a document application profile.

A.2.1 Conventions for production rules

This subclause includes notations for:

- construction expressions;
- string expressions;
- numeric expressions;
- object identifier expressions;
- bindings;
- references to binding values.

There are two aspects to the definition of this notation. Firstly, it is necessary to define the symbols used to denote any particular instance of an expression. Secondly, it is necessary to define a series of production rules for defining all valid instances of the strings of symbols used to denote the expressions.

The production rules are defined using a Backus-Naur-Form (BNF) which makes use of the following symbols:

a) ::= The definition operator, specifies that the string of symbols on the right-hand side is to be substituted for the non-terminal symbol on the left-hand side;

b) | the alternative operator, used to separate alternatives;

c) < > used to delimit a non-terminal symbol;

d) - - used to indicate a comment string;

e) { } used to delimit a syntactical unit;

f) [ ] used to delimit an optional syntactical unit, i.e. the syntactical unit may be present or absent;

g) ... symbol that may follow a syntactical unit, delimited as in c), e) or f) above, to indicate that the syntactical unit may appear one or more times.

The definition of the notation for each type of expression is given in the following subclauses. Each definition specifies the allowable terminal symbols that may be used.

A.2.2 Notation for construction expressions

The value of the attribute "generator for subordinates" is a construction expression (see 9.3.2.1).

This subclause defines a human readable notation for construction expressions.

The terminal symbols used are as follows:

SEQUENCE SEQ, AGGREGATE AGG, CHOICE CHO OPT REP ( )

The allowable formats of construction expressions are defined by the following production rules:

```
<construction expression>                 ::= <construction term>
                                               | <sequence construction>
                                                 | <aggregate construction>
                                                   | <choice construction>

<sequence construction>                   ::= SEQUENCE(<term sequence>)
                                               | SEQ(<term sequence>)

<aggregate construction>                  ::= AGGREGATE(<term sequence>)
                                               | AGG(<term sequence>)

<choice construction>                     ::= CHOICE(<term sequence>)
                                               | CHO(<term sequence>)

<term sequence>                           ::= { <construction term> }

<construction term>                       ::= <required construction factor>
                                               | <optional construction factor>
                                                 | <repetitive construction factor>
                                                   | <optional repetitive construction factor>

<required construction factor>            ::= <construction factor>

<optional construction factor>            ::= OPT <construction factor>

<repetitive construction factor>          ::= REP <construction factor>

<optional repetitive construction factor> ::= OPT REP <construction factor>

<construction factor>                     ::= <object class identifier>
                                               | <construction expression>

<object class identifier>                 ::= - - any character string from the set of characters: hyphen –:
                                                  the capital letters A, B ..., Z; the small letters a, b ..., z; the
                                                  digits 0 ..., 9.
```

A character string used to represent an object class identifier is a symbol used in this notation only; it is a symbolic representation of an actual object class identifier value (which is according to the format defined in 9.3.1.3).

A.2.3 Notation for string expressions

This subclause defines a human readable notation for string expressions, as defined in 9.1.3.1.

The terminal symbols used in this notation are as follows:

MAKE-STRING MK-STR UPPER-ALPHA U-ALPHA LOWER-ALPHA L-ALPHA

UPPER-ROMAN U-ROM LOWER-ROMAN L-ROM 'H " + ( )

The allowable formats of string expressions are defined by the following production rules:

```
<string expression>                       ::= <atomic string expression>
                                               | <atomic string expression> + <string expression>
```

The symbol + indicates concatenation of terms.

```
<atomic string expression>          ::= <string literal>
                                        | <binding reference
                                          | <string function application>

<string literal>                    ::= "<character string>"
                                        | '<hexadecimal string>'H

<string function application>       ::= <make string application>
                                        | <upper alpha application>
                                          | <lower alpha application>
                                            | <upper Roman application>
                                              | <lower Roman application>

<make string application>           ::= MAKE-STRING(<numeric expression>)
                                        | MK-STR(<numeric expression)

<upper alpha application>           ::= UPPER-ALPHA(<numeric expression>)
                                        | U-ALPHA(<numeric expression>)

<lower alpha application>           ::= LOWER-ALPHA(<numeric expression>)
                                        | L-ALPHA(<numeric expression>)

<upper Roman application>           ::= UPPER-ROMAN(<numeric expression>)
                                        | U-ROM(<numeric expression>)

<lower Roman application>           ::= LOWER-ROMAN(<numeric expression>)
                                        | L-ROM(<numeric expression>)

<binding reference>                 ::= - - see A.2.7.

<numeric expression>                ::= - - see A.2.4.
```

When a string literal consists of a character string, the character repertoire that is being used shall be indicated. Whenever a character string contains the character “(quotation mark), the convention is to denote this by the characters “” (two consecutive quotation marks).

When a hexadecimal string is used, the allowable characters are:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

A.2.4 Notation for numeric expressions

This subclause defines a human readable notation for numeric expressions (see 9.1.3.2).

The terminal symbols used are as follows:

INCREMENT INC DECREMENT DEC ORDINAL ORD ( ) –

The allowable formats of numeric expressions are defined by the following production rules:

```
<numeric expression>                ::= <numeric literal>
                                        | <binding reference>
                                          | <numeric function application>

<numeric literal>                   ::= - - any negative, zero or positive integer: negative integer
                                            values are preceded by hyphen -; integer values are
                                            represented by a string of digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

<binding reference>                 ::= - - see A.2.7.

<numeric function application>      ::= <increment function>
                                        | <decrement function>
                                          | <ordinal function>

<increment function>                ::= INCREMENT (<numeric expression>)
                                        | INC (<numeric expression>)

<decrement function>                ::= DECREMENT (<numeric expression>)
                                        | DEC (<numeric expression>)
```

```
<ordinal function>                    ::== ORDINAL (<object reference>
                                           | ORD (<object reference>)

<object reference>                    ::= <object identifier>
                                           | <object identifier expression>

<object identifier>                   ::= - -  any character string from the set of characters: hyphen -;
                                               the capital letters A, B ..., Z; the small letters a, b ..., z; the
                                               digits 0 ..., 9.

<object identifier expression>        ::= - - see A.2.5.
```

A character string used to represent an object identifier is a symbol used in this notation only; it is a symbolic representation of an actual object identifier value (which is according to the format defined in 9.3.1.2).

A.2.5 Notation for object identifier expressions

This subclause defines a human readable notation for object identifier expressions.

The terminal symbols used in this notation are as follows:

CURRENT-OBJECT CURR-OBJ CURRENT-INSTANCE CURR-INST SUPERIOR-OBJECT SUP-OBJ PRECEDING-OBJECT PREC-OBJ DOCUMENT_LAYOUT_ROOT DLAR PAGE_SET PAGE FRAME BLOCK DOCUMENT_LOGICAL_ROOT DLOR COMPOSITE_LOGICAL_OBJECT CLO BASIC_LOGICAL_OBJECT BLO ( )

The allowable formats of object identifier expressions are defined by the following production rules:

```
<object identifier expression>        ::=<current-object function>
                                           | <current-instance function>
                                             | <superior-object function>
                                               | <preceding-object function>

<current-object function>             ::= CURRENT-OBJECT
                                           | CURR-OBJ

<current-instance function>           ::= CURRENT-INSTANCE(<class-or-type>,
                                                          <object reference>)
                                           | CURR-INST(<class-or-type>,
                                                     <object reference>)

<superior-object function>            ::= SUPERIOR-OBJECT(<object identifier expression>)
                                           | SUP-OBJ(<object identifier expression>)

<preceding-object function>           ::= PRECEDING-OBJECT(<object identifier expression>)
                                           | PREC-OBJ(<object identifier expression>)

<class-or-type>                       ::= <object class identifier>
                                           | <object type>

<object class identifier>             ::= - - see A.2.2.

<object type>                         ::= DOCUMENT_LAYOUT_ROOT| DLAR
                                      | PAGE_SET
                                           | PAGE
                                             | FRAME
                                               | BLOCK
                                                 | DOCUMENT_LOGICAL_ROOT| DLOR
                                                   | COMPOSITE_LOGICAL_OBJECT| CLO
                                                      | BASIC_LOGICAL_OBJECT| BLO

<object reference>                    ::=<object identifier>
                                           | (<object identifier expression>)

<object identifier>                   ::= - - see A.2.4.
```

A character string used to represent an object identifier or an object class identifier is a symbol used in this notation only; it is a symbolic representation of an actual object identifier value or an object class identifier value (which are according to the formats defined in 9.3.1.2 and 9.3.1.3, respectively).

A.2.6 Bindings

The attribute "bindings" consists of a set of pairs of parameters, each pair consisting of a binding name and a binding value (see 9.3.5.4).

This subclause defines a human readable notation for this pair of parameters.

<binding pair> ::= <binding name>, <binding value>

<binding name> ::= - - *any character string from the minimum subrepertoire of ISO6937-2, being the value of the binding name parameter.*

<binding value> ::= <string expression>
| <numeric expression>
| <object reference>

<string expression> ::= - - *see A.2.3.*

<numeric expression> ::= - - *see A.2.4.*

<object reference> ::= <object identifier>
| <object identifier expression>

<object identifier> ::= - - *see A.2.4.*

<object identifier expression> ::= - - *see A.2.5.*

A character string used to represent an object identifier is a symbol used in this notation only; it is a symbolic representation of an actual object identifier value (which is according to the format defined in 9.3.1.2).

A.2.7 Notation for references to binding values

This subclause defines a human readable notation for references to binding values.

The terminal symbols used in this notation are as follows:

BINDING_REFERENCE B_REF CURRENT-OBJECT CURR-OBJ SUPERIOR SUP PRECEDING PREC

The allowable formats of binding references are defined by the following production rules:

<binding reference> ::= BINDING_REFERENCE (<binding reference expression>) (<binding name>)
| B_REF(<binding reference expression>) (<binding name>)

<binding reference expression> ::= <object identifier>
| <binding selection function>

<binding name> ::= - - *see A.2.6.*

<object identifier> ::= - - *see A.2.4.*

<binding selection function> ::= <current-object function>
| <current-instance function>
| <superior function>
| <preceding function>

<current-object function> ::= CURRENT-OBJECT
| CURR-OBJ

<current-interface function> ::= - - *see A.2.5.*

<superior function> ::= SUPERIOR(<object identifier expression>)
| SUP(<object identifier expression>)

<preceding function> ::= PRECEDING(<object identifier expression>)
| PREC(<object identifier expression>)

<object identifier expression> ::= - - *see A.2.5.*

A character string used to represent an object identifier is a symbol used in this notation only; it is a symbolic representation of an actual object identifier value (which is according to the format defined in 9.3.1.2).

Annex B

Examples of document structures

(This annex does not form an integral part of this Recommendation | International Standard)

B.1 Introduction

This annex presents examples of the application of the document architecture specified in this Specification to a single specimen document, namely a typical business letter. Although the prime purpose of these examples is to illustrate the document structures, they also refer to the document layout process described in clause 10 and to content layout processes which are described in other Specifications of the ITU-T Rec. T.410-Series | ISO/IEC 8613 which specify individual content architectures. Some knowledge of these processes, although not essential, would be helpful in reading these examples.

The first two examples (see B.4.1, B.4.2) describe how the specimen document may be represented in terms of a specific layout structure and a specific logical structure, respectively. These examples show that the originator may take two distinct views of the same document when it is created, according to the originator's application. That is, they show how the same document may be constructed in formatted form or in processable form.

The third example (see B.5) again shows the specimen document in processable form but, in this case, the document includes a generic logical structure and a generic layout structure. These generic structures may be used to provide two examples of document classes. In the first of these examples, the generic logical structure alone may be regarded as the document class from which the specific logical structure, previously described in the second example, may have been generated.

In the second example of document classes, the two generic layout structures may be used together to form the document class. In this case, the generic layout structure may be used to control the layout of the document during the layout process.

The next example (see B.6) describes the specific layout structure that would be generated by that layout process. This specific layout structure is not identical to that described in the first example (see B.4.1), for reasons that will be explained, but the appearance of the document on a presentation device will be exactly the same as if the document had been laid out according to the specific layout structure described in the first example.

Also note that if the specific layout structure generated by the generic layout structure is interchanged together with the specific logical structure, generic logical structure and generic layout structure described in the third example, then the document is said to be in formatted processable form.

The structures of the document are illustrated by structure diagrams which make use of the notation defined in Annex A.

B.2 Notation used to specify constituents of a document

A notation described in this clause is used to specify the sets of attribute values which characterize the constituents of the document.

B.2.1 General

In this notation, the specification of each constituent is separated by a horizontal line. In the specification of each constituent the left hand column specifies the attribute names and the right hand column the attribute values.

For example:

Object Type	COMPOSITE LOGICAL

In this example, “object type” is the name of an attribute for which one of the possible values is 'composite logical'.

In this notation, object class descriptions are identified by a name in parentheses as well as a numeric string, rather than a numeric string only, as specified in 9.3.1.3. This makes them easier to relate to the structure diagrams.

Object class descriptions may occur in any order in the tables given in these examples since they are not hierarchically structured. However, they are presented where possible in an order similar to that of the structure diagrams.

Object descriptions are identified by sequences of numbers separated by spaces. That is, their identifiers in these examples have the same form as the value of the attribute "object identifier" as specified in 9.3.1.2.

Object descriptions are written in the tables in these examples in the sequential order defined by the specific structure to which they belong. Subordinate objects are identified by the last element of their identifier.

B.2.2 Generator for subordinates

The notation for construction expressions is as defined in Annex A.

B.2.3 Content portions

Generic content portion descriptions are identified by a name in parentheses as well as a numeric string, in a manner analogous to that used to identify object class descriptions.

Content portion descriptions within the representation of specific structures are identified by sequences of numbers separated by spaces.

The value of the content information of a content portion is represented in one of two ways:

- as a quoted string, for example, "is a string";
- as a comment string, for example, *This is another string*.

Quoted strings are used when the content information can be "reasonably" represented by this syntax.

Comment strings are used when the content information can not be reasonably represented by the quoted string syntax or when to do so would not significantly improve understanding of the example.

Within quoted strings multiple spaces and new lines have no significance i.e. the presentation of a string is to be interpreted as having no significance. Where control characters are to be considered as significant they are written as:

\x

where x is a single letter or a number followed by a letter.

The letters have the following meanings:

- n: new line;
- s: space;
- t: tabulate.

When one of the above letters is preceded by a number it means that the number of control functions as specified by the letter are to be considered as present.

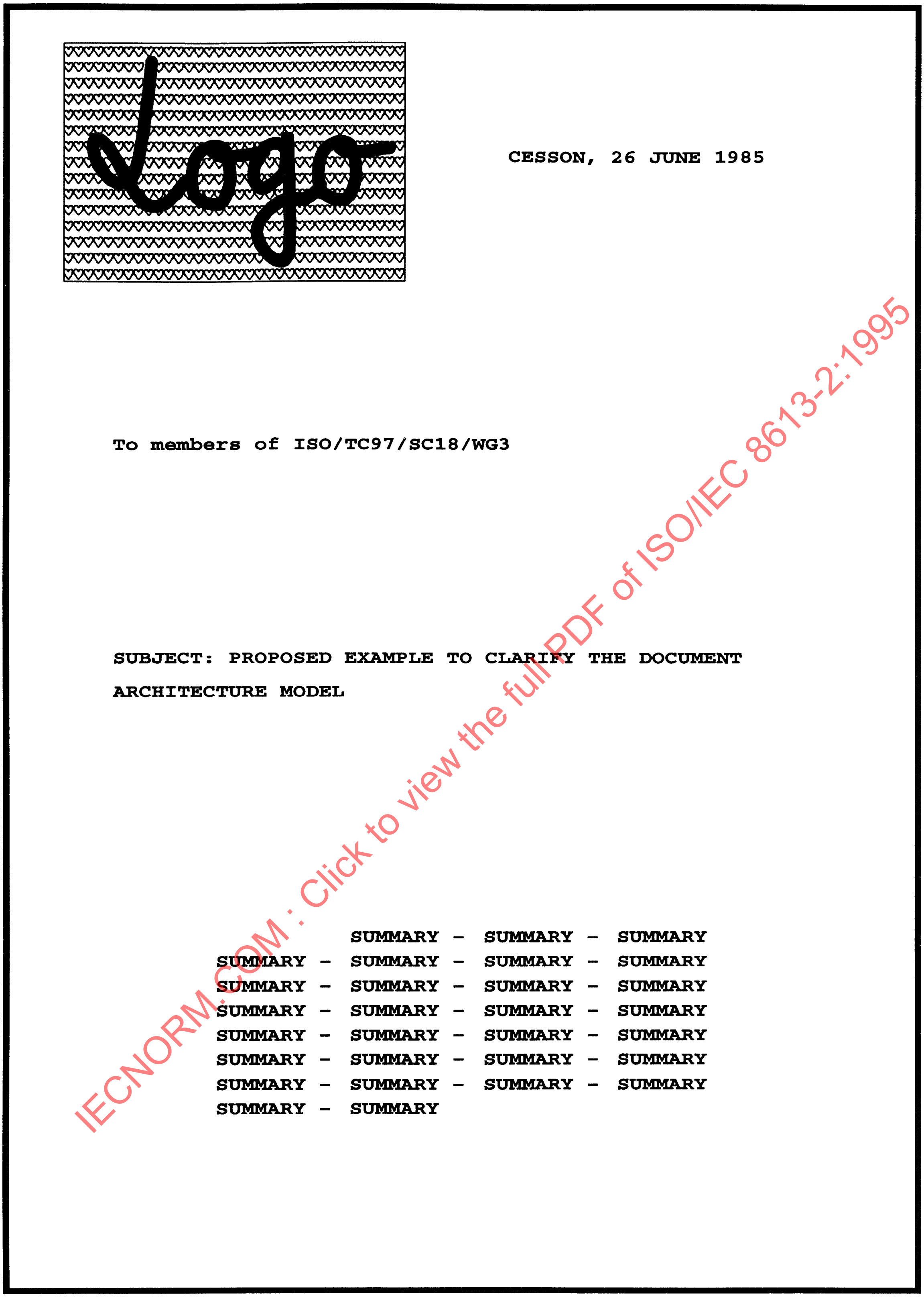
When content portion descriptions are associated with both the specific logical structure and the specific layout structure they are represented as separate constituents with appropriate attribute values for their identifiers.

B.3 Introduction to specimen document

There follows an illustration of an application of the document architecture to a class of documents called "letter". Figures B.1 to B.3 illustrate the specimen document.

The specimen document consists of three pages. The first page contains a logo, a date, the name of the addressee, a statement of the subject and a summary. The second page contains two paragraphs, a figure and the first part of a third paragraph. The last page contains the remaining part of the third paragraph, a fourth paragraph, a formal ending and the signature and name of the letter's originator. The contents of the various paragraphs is shown in a symbolic form.

Figures B.4 to B.6 illustrate the layout structure of this document by outlining various blocks within each page.



CESSON, 26 JUNE 1985

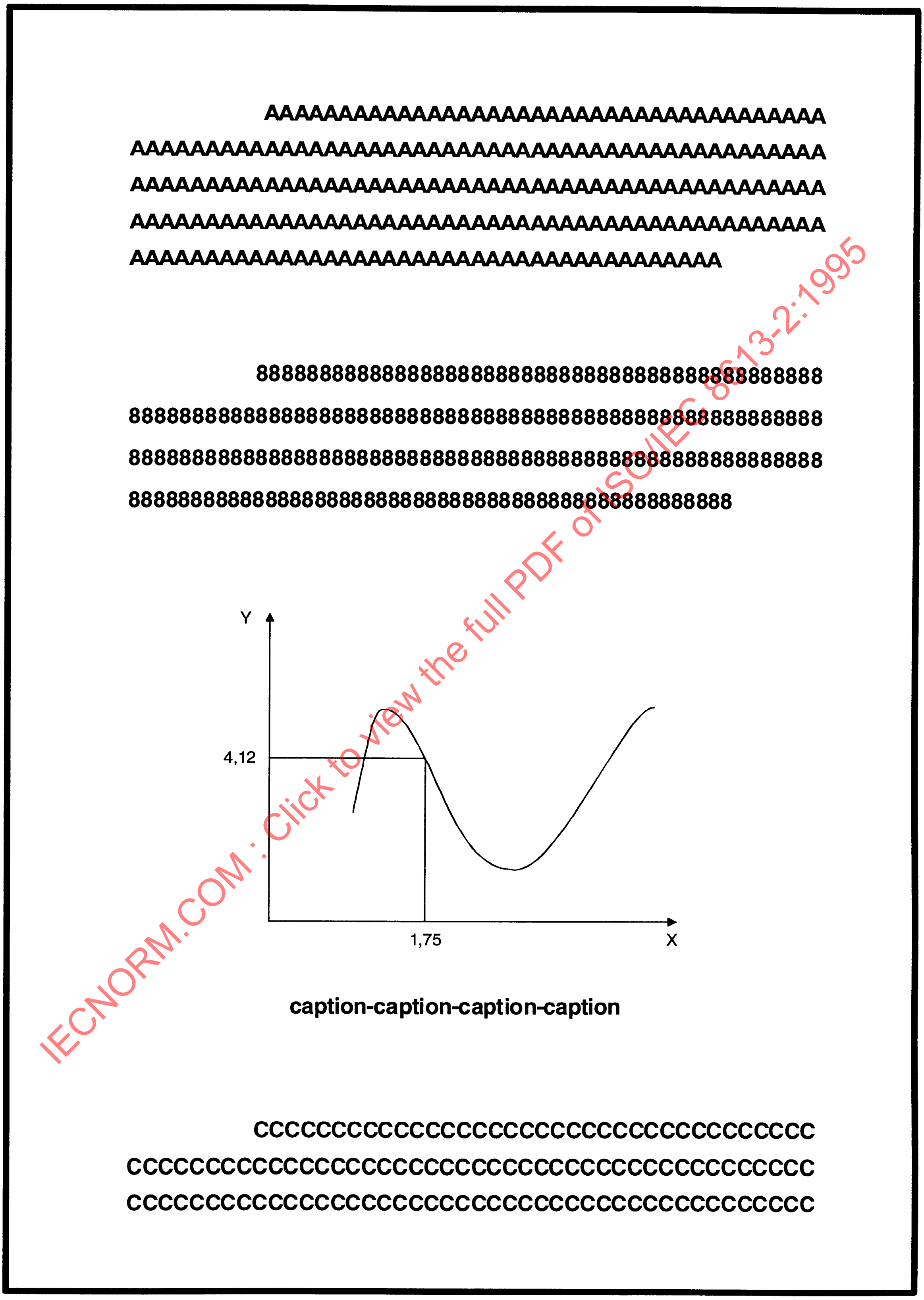
To members of ISO/TC97/SC18/WG3

SUBJECT: PROPOSED EXAMPLE TO CLARIFY THE DOCUMENT
ARCHITECTURE MODEL

SUMMARY – SUMMARY – SUMMARY
SUMMARY – SUMMARY – SUMMARY – SUMMARY
SUMMARY – SUMMARY – SUMMARY – SUMMARY
SUMMARY – SUMMARY – SUMMARY – SUMMARY
SUMMARY – SUMMARY – SUMMARY – SUMMARY
SUMMARY – SUMMARY – SUMMARY – SUMMARY
SUMMARY – SUMMARY – SUMMARY – SUMMARY
SUMMARY – SUMMARY

T0816010-94/d24

Figure B.1 – Specimen document "letter" (1)



AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

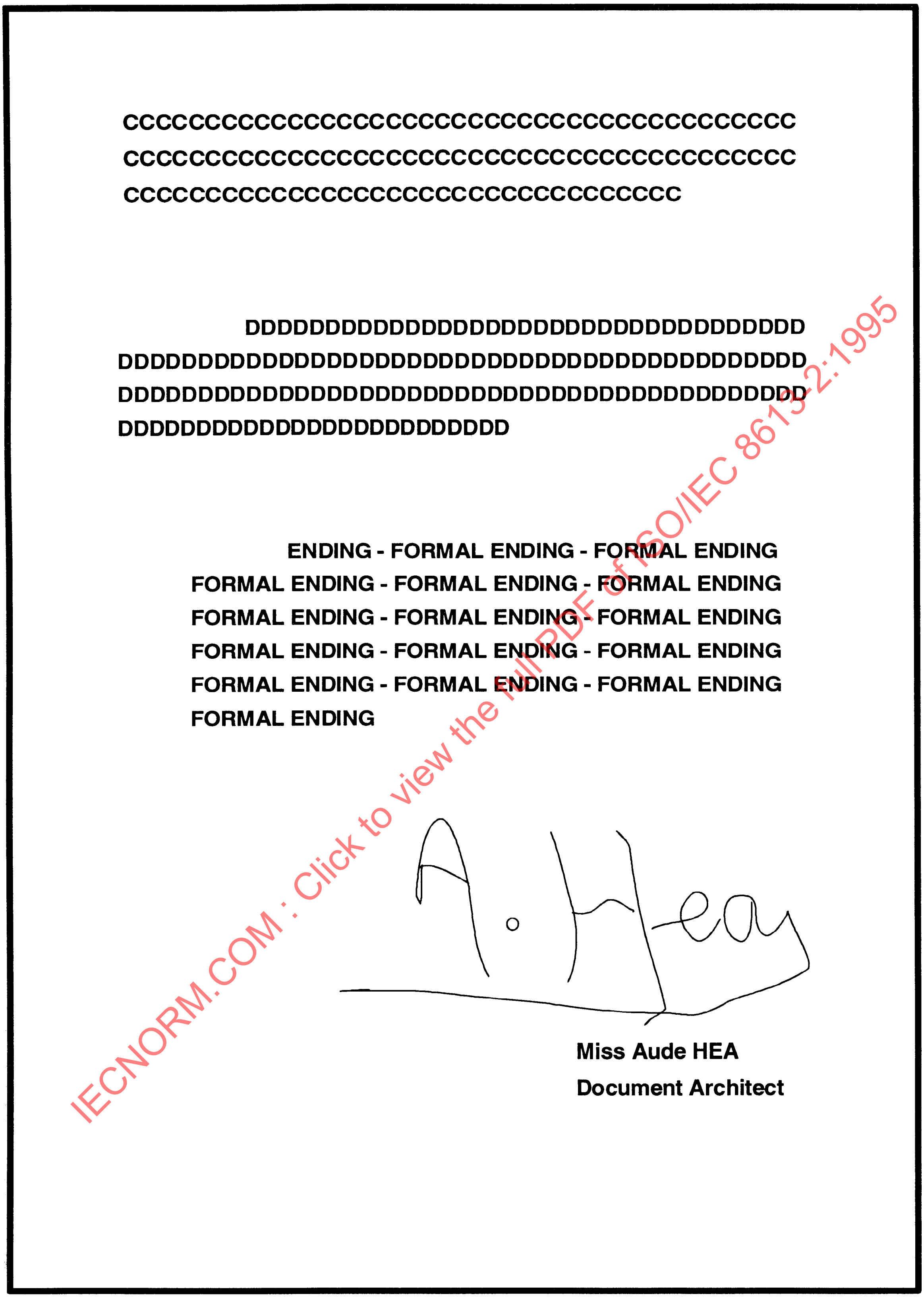
8888888888888888888888888888888888888888888888
88888888888888888888888888888888888888888888888888888888
88888888888888888888888888888888888888888888888888888888
8888888888888888888888888888888888888888888888888

caption-caption-caption-caption

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

T0816020-94/d25

Figure B.2 – Specimen document “letter” (2)



CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDD

ENDING - FORMAL ENDING - FORMAL ENDING
FORMAL ENDING - FORMAL ENDING - FORMAL ENDING
FORMAL ENDING - FORMAL ENDING - FORMAL ENDING
FORMAL ENDING - FORMAL ENDING - FORMAL ENDING
FORMAL ENDING - FORMAL ENDING - FORMAL ENDING
FORMAL ENDING

A. Hea

Miss Aude HEA
Document Architect

T0816030-94/d26

Figure B.3 – Specimen document "letter" (3)

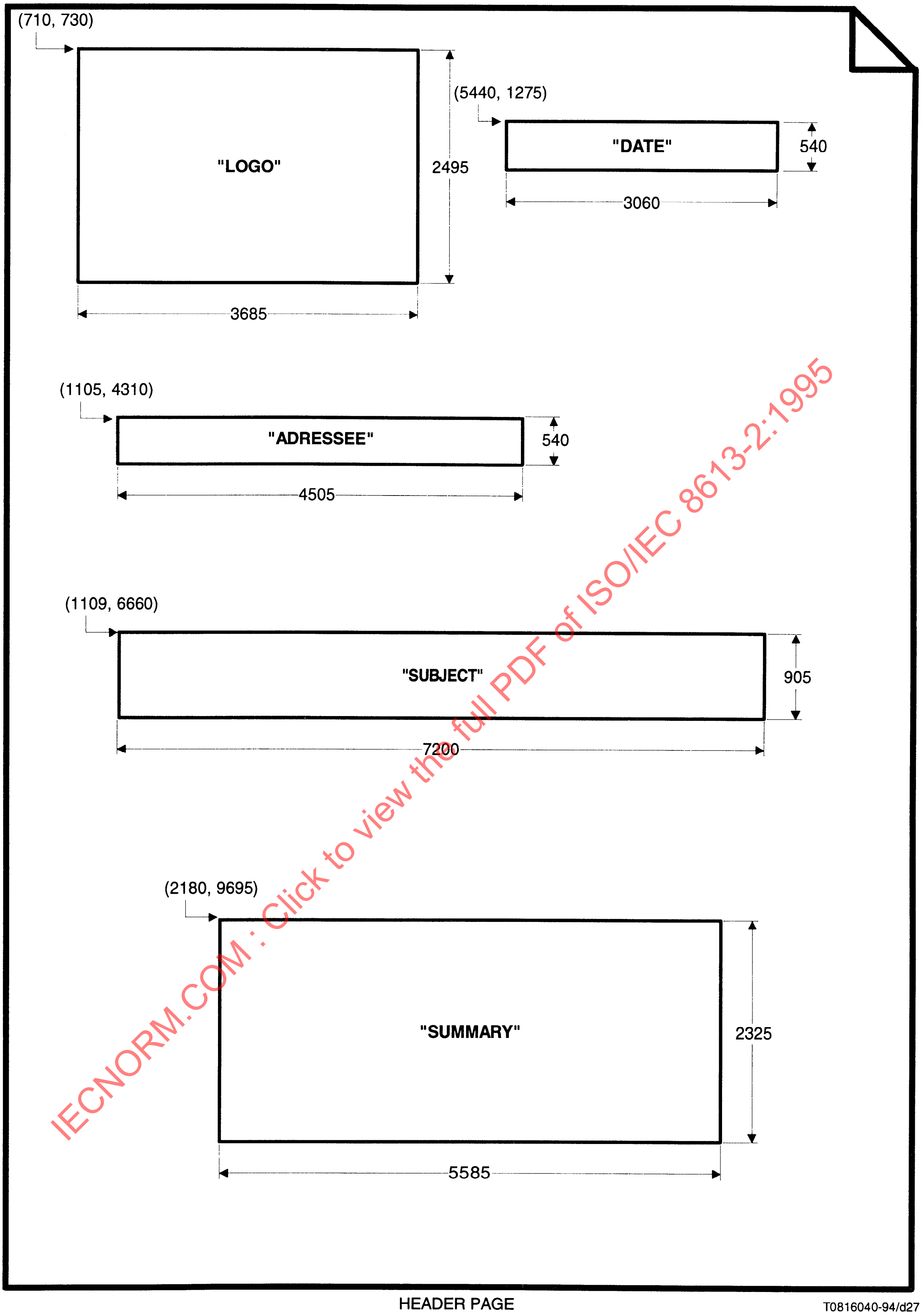
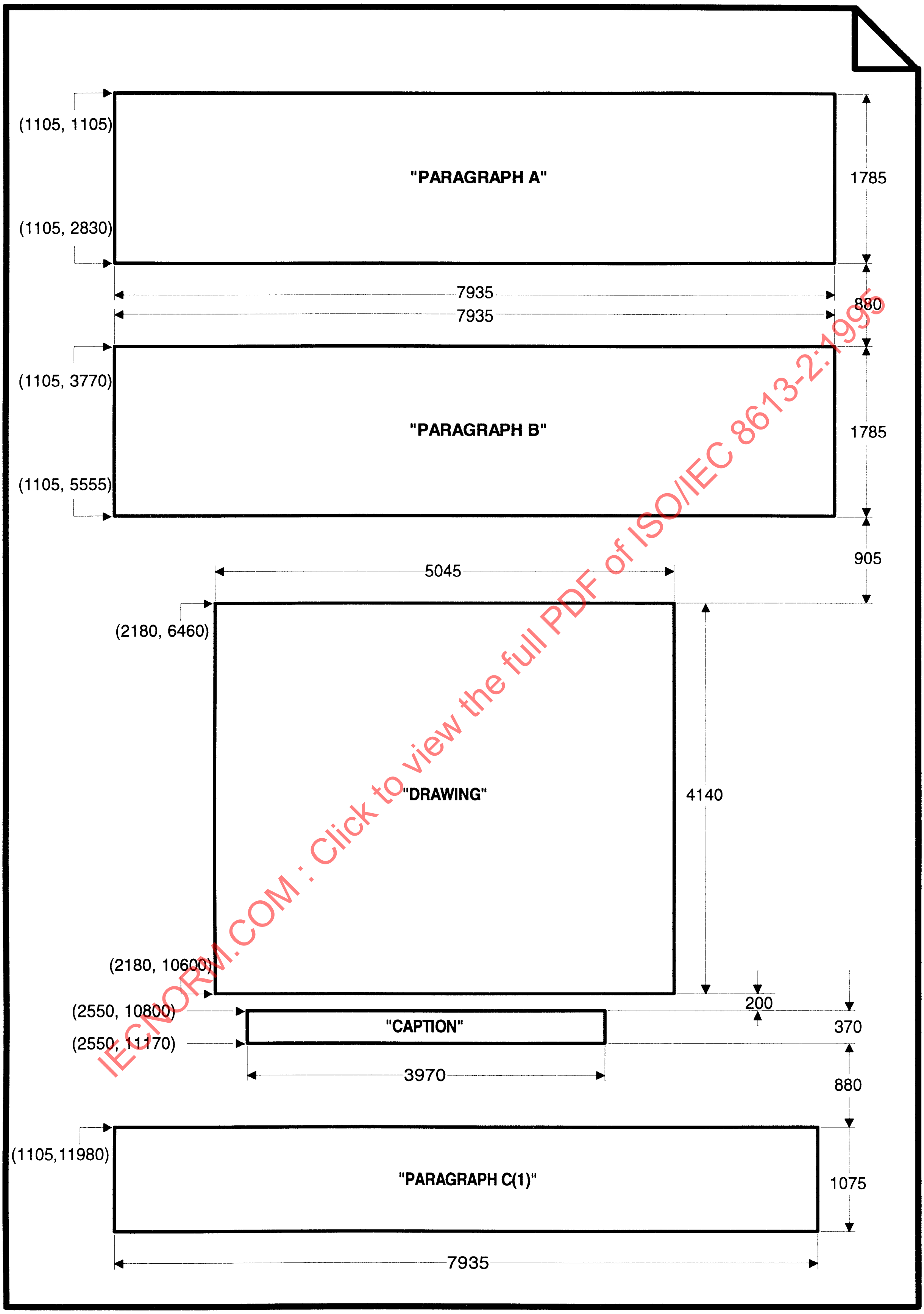


Figure B.4 – Layout of "letter" showing pages and blocks (1)



T0816050-94/d28

Figure B.5 – Layout of "letter" showing pages and blocks (2)

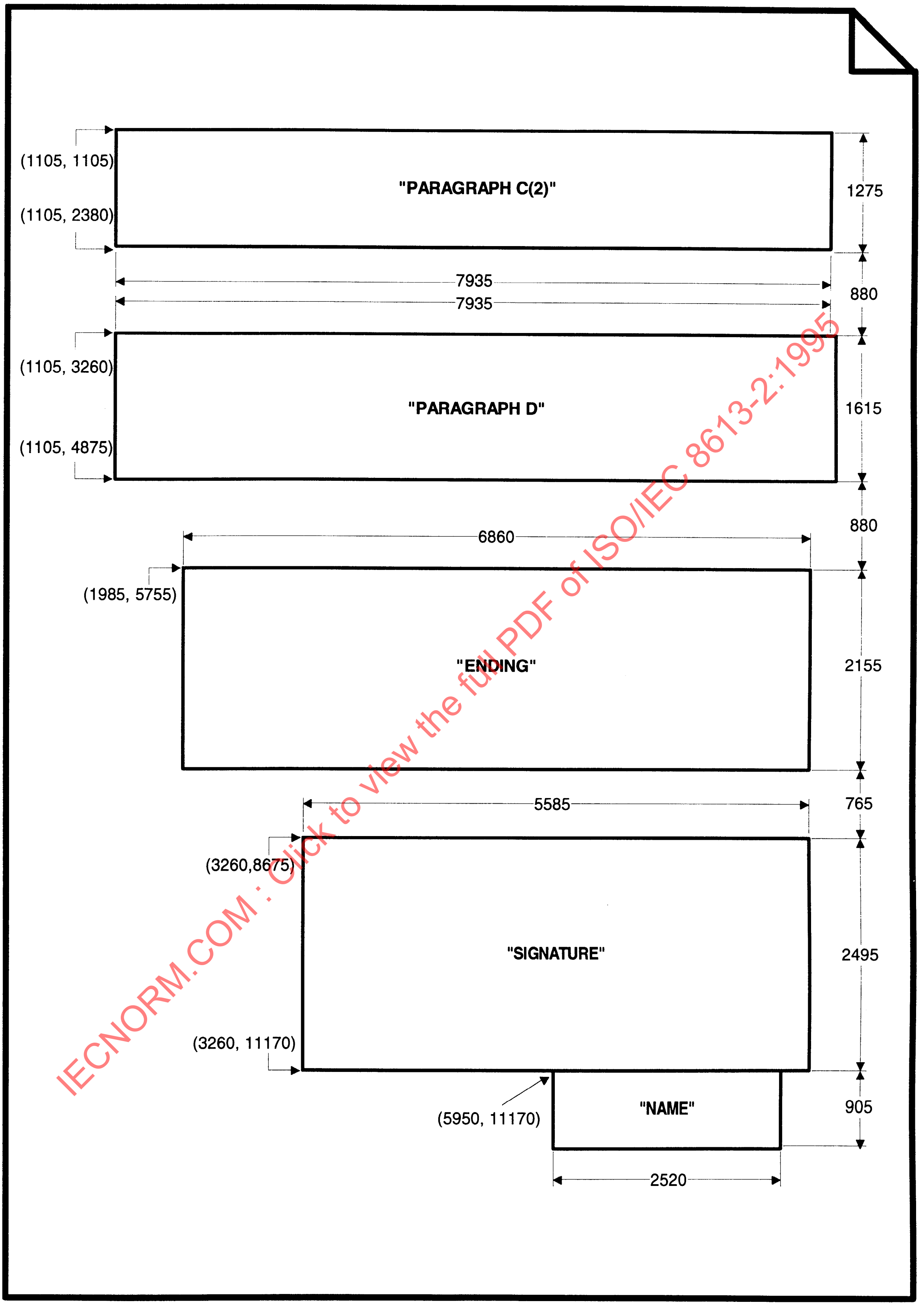


Figure B.6 – Layout of "letter" showing pages and blocks (3)

B.4 Specific structures

A document can be viewed in two different ways:

- as a layout structure in which the appearance of the document content is of prime concern;
- as a logical structure in which the meaning of the document content is considered, such as its division into chapters, paragraphs.

These two structures are described further below.

B.4.1 Formatted form documents with specific layout structure only

The specimen document has a specific layout structure that may be illustrated by a diagram as in figure B.7.

The diagram contains the same pages and blocks as Figures B.4 to B.6 but represents these in a hierarchical form. The content is divided into a number of content portions and each of these is allocated to a block. Hence there is a logo block, a date block, a subject block, etc. The content portions that belong to each block are found in the bottom row of Figure B.7 (in double lined boxes).

The document contains one paragraph – paragraph C – which is contained in two blocks, each block on a different page.

Each layout object, i.e. page or block, is characterised by a number of attributes. Table B.1 contains a list of all the attributes that have to be specified in this example.

Some attributes that are "defaultable" are not shown in the table. Their values may be derived from the standard default values as defined in this Specification (see 9.1.2.4).

B.4.2 Processable form document with specific logical structure only

The logical objects of the specimen document are organised in a hierarchical structure as shown in Figure B.8.

It shows that the document consists of two directly subordinate composite logical objects representing the 'header' and 'body'. The 'header' consists of basic logical objects representing the 'date', 'addressee', 'subject', etc. Content portions are assigned to the basic logical objects. There is no basic logical object for the logo because this is assumed to be part of the layout structure only (i.e. either part of the generic layout structure or preprinted on the presentation medium).

The same logical object class may be used in several places (for example, paragraph) but with different content.

Each logical object is characterised by its attributes in the same way as for the layout objects. Tables B.2 and B.3 list all the constituents and the attributes that have to be specified in this example. Defaultable attributes are not listed unless they have been assigned non-default values.

The presentation attributes applicable to the content associated with the basic logical objects are contained in presentation styles which are listed in Table B.3. When required, each basic logical object contains an attribute which references an appropriate presentation style.

Presentation attributes specify how the document content is to be presented and imaged on the presentation media. For example, in the case of character content, these attributes specify the line and character spacing intervals and whether or not the content is to be justified.

However, this information is insufficient to lay out the content of the document. Additional information is required concerning the document layout structure, that is information is required relating to the characteristics and internal structure (if any) of the pages on which the document content is to be laid out. In this example, it is assumed that the recipient will provide the required information. The recipient may use the ITU-T Rec. T.410-Series | ISO/IEC 8613 to define such a layout process or the recipient may specify his own rules for deriving the document layout structure.

Hence the recipient is not likely to produce exactly the same layout for the document as illustrated earlier in this annex. The next example (see B.5) describes how a document can be interchanged in processable form with sufficient information, in the form of generic layout structure, to indicate the layout required.

Also, this example does not make use of layout styles, which contain attributes that, for example, specify the amount of space to be inserted between successive logical objects (using the attribute "separation") and whether or not certain logical objects shall be placed at the start of a new page (using attribute "new layout object"). This does not mean that layout styles cannot be included in documents which are interchanged with logical structure only. However, there is no obligation for the originator to include such information and again it might be necessary for the recipient to supply additional information to obtain an acceptable document layout.

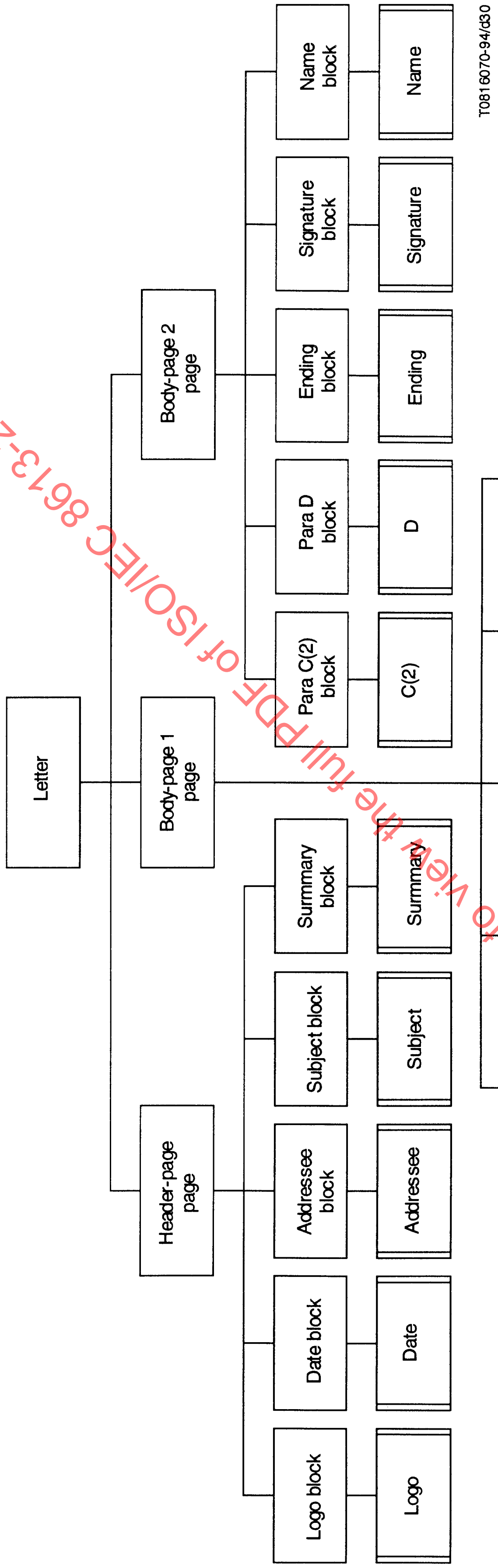


Figure B.7 – Specific layout structure (showing pages and blocks)

Table B.1 – Constituents for Figure B.7 – Specific layout structure

Object Type Object Identifier User-Visible Name Subordinates	DOCUMENT LAYOUT ROOT 1 "Letter" 0, 1, 2
Object Type Object Identifier User-Visible Name Dimensions Subordinates	PAGE 1 0 "Header Page" HD = 9920, VD = 14030 0, 1, 2, 3, 4
Object Type Object Identifier User-Visible Name Position Dimensions Content Architecture Class Content Portions	BLOCK 1 0 0 "Logo" HP = 710, VP = 730 HD = 3685, VD = 2495 FORMATTED FORM RASTER GRAPHICS 0
Content Identifier – Layout Content Information	1 0 0 0 /*Array of raster-graphics content elements for the logo*/
Object Type Object Identifier User-Visible Name Position Dimensions Content Portions	BLOCK 1 0 1 "Date" HP = 5440, VP = 1275 HD = 3060, VD = 540 0
Content Identifier – Layout Content Information	1 0 1 0 "CESSON, 26 JUNE 1985"
Object Type Object Identifier User-Visible Name Position Dimensions Content Portions	BLOCK 1 0 2 "Addressee" HP = 1105, VP = 4310 HD = 4505, VD = 540
Content Identifer-Layout Content Information	1 0 2 0 "To members of ISO/TC97/SC18/WG3"
Object Type Object Identifier User-Visible Name Position Dimensions Line Spacing Content Portions	BLOCK 1 0 3 "Subject" HP = 1105, VP = 6660 HD = 7200, VD = 905 300 0
Content Identifier-Layout Content Information	1 0 3 0 "SUBJECT: PROPOSED EXAMPLE TO CLARIFY THE DOCUMENT\n ARCHITECTURE MODEL"
Object Type Object Identifier User-Visible Name Position Dimensions Alignment Content Portions	BLOCK 1 0 4 "Summary" HP = 2180, VP = 9695 HD = 5585, VD = 2325 JUSTIFIED 0
Content Identifier-Layout Content Information	1 0 4 0 /*Formatted string of SUMMARY-*/

Table B.1 (continued)

Object Type Object Identifier User-Visible Name Dimensions Subordinates	PAGE 1 1 "Body Page 1" HD = 9920, VD = 14030 0, 1, 2, 3, 4
Object Type Object Identifier User-Visible Name Position Dimensions Line Spacing Alignment Content Portions	BLOCK 1 1 0 "Para A" HP = 1105, VP = 1105 HD = 7935, VD = 1785 300 JUSTIFIED 0
Content Identifier-Layout Content Information	1 1 0 0 /*Formatted string of A's*/
Object Type Object Identifier User-Visible Name Position Dimensions Line Spacing Alignment Content Portions	BLOCK 1 1 1 "Para B" HP = 1105, VP = 3770 HD = 7935, VD = 1785 400 JUSTIFIED 0
Content Identifier-Layout Content Information	1 1 1 0 /*Formatted string of B's*/
Object Type Object Identifier User-Visible Name Position Dimensions Content Architecture Class Content Portions	BLOCK 1 1 2 "Drawing" HP = 2180, VP = 6460 HD = 5045, VD = 4140 FORMATTED GEOMETRIC GRAPHICS 0
Content Identifier-Layout Content Information	1 1 2 0 /*Ordered set of geometric-graphics content elements for the diagram*/
Object Type Object Identifier User-Visible Name Position Dimensions Content Portions	BLOCK 1 1 3 "Caption" HP = 2550, VP = 10800 HD = 3970, VD = 370 0
Content Identifier-Layout Content Information	1 1 3 0 /*Formatted string for the caption*/
Object Type Object Identifier User-Visible Name Position Dimensions Line Spacing Alignment Content Portions	BLOCK 1 1 4 "Para C(1)" HP = 1105, VP = 11980 HD = 7935, VD = 1075 300 JUSTIFIED 0
Content Identifier-Layout Content Information	1 1 4 0 /*Formatted string of C's*/

Table B.1 (end)

Object Type Object Identifier User-Visible Name Dimensions Subordinates	PAGE 1 2 "Body Page 2" HD = 9920, VD = 14030 0, 1, 2, 3, 4
Object Type Object Identifier User-Visible Name Position Dimensions Line Spacing Alignment Content Portions	BLOCK 1 2 0 "Para C(2)" HP = 1105, VP = 1105 HD = 7935, VD = 1275 300 JUSTIFIED 0
Content Identifier-Layout Content Information	1 2 0 0 /*Formatted string of C's*/
Object Type Object Identifier User-Visible Name Position Dimensions Line Spacing Alignment Content Portions	BLOCK 1 2 1 "Para D" HP = 1105, VP = 3260 HD = 7935, VD = 1615 300 JUSTIFIED 0
Content Identifier-Layout Content Information	1 2 1 0 /*Formatted string of D's*/
Object Type Object Identifier User-Visible Name Position Dimensions Line Spacing Alignment Content Portions	BLOCK 1 2 2 "Ending" HP = 1985, VP = 5755 HD = 6860,VD = 2155 300 JUSTIFIED 0
Content Identifier-Layout Content Information	1 2 2 0 /*Formatted string of FORMAL ENDING*/
Object Type Object Identifier User-Visible Name Position Dimensions Content Architecture Class Content Portions	BLOCK 1 2 3 "Signature" HP = 3260, VP = 8675 HD = 5585, VD = 2495 FORMATTED RASTER GRAPHICS 0
Content Identifier-Layout Content Information	1 2 3 0 /*Array of raster-graphics content elements for the signature*/
Object Type Object Identifier User-Visible Name Position Dimensions Line Spacing Content Portions	BLOCK 1 2 4 "Name" HP = 5950, VP = 11170 HD = 2520, VD = 905 300 0
Content Identifier-Layout Content Information	1 2 4 0 "Miss Aude HEA\nDocument Architect"

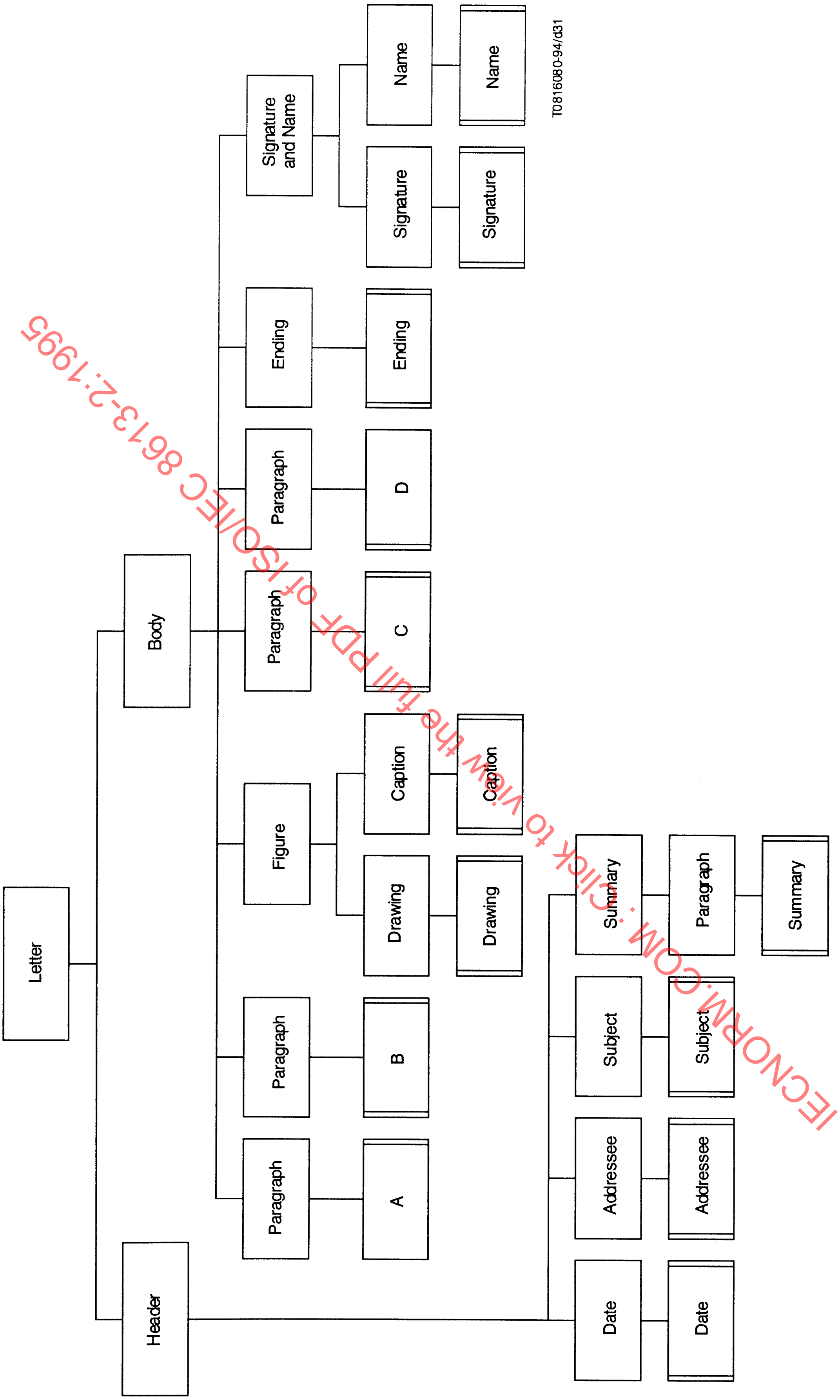


Figure B.8 – Specific logical structure

Table B.2 – Constituents for Figure B.8 – Specific logical structure

Object Type	DOCUMENT LOGICAL ROOT
Object Identifier	3
User-Visible Name	“Letter”
Subordinates	0, 1
Default Value Lists	list of basic logical objects
	attribute: content architecture class
	value: 'processable character'
Object Type	COMPOSITE LOGICAL
Object Identifier	3 0
User-Visible Name	“Header”
Subordinates	0, 1, 2, 3
Object Type	BASIC LOGICAL
Object Identifier	3 0 0
User-Visible Name	“Date”
Content Portions	0
Content Identifier-Logical	3 0 0 0
Content Information	“CESSON 26 JUNE 1985”
Object Type	BASIC LOGICAL
Object Identifier	3 0 1
User-Visible Name	“Addressee”
Content Portions	0
Content Identifier-Logical	3 0 1 0
Content Information	“To members of ISO/TC97/SC18/WG3”
Object Type	BASIC LOGICAL
Object Identifier	3 0 2
User-Visible Name	“Subject”
Presentation Style	5 0
Content Portions	0
Content Identifier-Logical	3 0 2 0
Content Information	“SUBJECT: PROPOSED EXAMPLE TO CLARIFY THE DOCUMENT ARCHITECTURE MODEL”
Object Type	COMPOSITE LOGICAL
Object Identifier	3 0 3
User-Visible Name	“Summary”
Subordinates	0
Object Type	BASIC LOGICAL
Object Identifier	3 0 3 0
User-Visible Name	“Summary paragraph”
Presentation Style	5 1
Content Portions	0
Content Identifier-Logical	3 0 3 0 0
Content Information	/*Unformatted string of SUMMARY-*/
Object Type	COMPOSITE LOGICAL
Object Identifier	3 1
User-Visible Name	“Body”
Subordinates	0, 1, 2, 3, 4, 5, 6
Object Type	BASIC LOGICAL
Object Identifier	3 1 0
User-Visible Name	“Paragraph A”
Presentation Style	5 2
Content Portions	0

Table B.2 (continued)

Content Identifier-Logical Content Information	3 1 0 0 /*Unformatted string of A's*/
Object Type Object Identifier User-Visible Name Presentation Style Content Portions	BASIC LOGICAL 3 1 1 “Paragraph B” 5 2 0
Content Identifier-Logical Content Information	3 1 1 0 /*Unformatted string of B's*/
Object Type Object Identifier User-Visible Name Subordinates	COMPOSITE LOGICAL 3 1 2 “Figure” 0, 1
Object Type Object Identifier User-Visible Name Content Architecture Class Content Portions	BASIC LOGICAL 3 1 2 0 “Drawing” FORMATTED PROCESSABLE GEOMETRIC GRAPHICS 0
Content Identifier-Logical Content Information	3 1 2 0 0 /*Ordered set of geometric-graphics content elements for the diagram*/
Object Type Object Identifier User-Visible Name Content Portions	BASIC LOGICAL 3 1 2 1 “Caption” 0
Content Identifier-Logical Content Information	3 1 2 1 0 /*Unformatted string for the caption*/
Object Type Object Identifier User-Visible Name Presentation Style Content Portions	BASIC LOGICAL 3 1 3 “Paragraph C” 5 2 0
Content Identifier-Logical Content Information	3 1 3 0 /*Unformatted string of C's*/
Object Type Object Identifier User-Visible Name Presentation Style Content Portions	BASIC LOGICAL 3 1 4 “Paragraph D” 5 2 0
Content Identifier-Logical Content Information	3 1 4 0 /*Unformatted string of D's*/
Object Type Object Identifier User-Visible Name Presentation Style Content Portions	BASIC LOGICAL 3 1 5 “Ending” 5 3 0
Content Identifier-Logical Content Information	3 1 5 0 /*Unformatted string for Ending*/

Table B.2 (end)

Object Type Object Identifier User-Visible Name Subordinates	COMPOSITE LOGICAL 3 1 6 “Signature and Name” 0, 1
Object Type Object Identifier User-Visible Name Content Architecture Class Content Portions	BASIC LOGICAL 3 1 6 0 “Signature” PROCESSABLE FORM RASTER GRAPHICS 0
Content Identifier-Logical Content Information	3 1 6 0 0 /*Array of raster-graphics content elements for the signature*/
Object Type Object Identifier User-Visible Name Content Portions Presentation Style	BASIC LOGICAL 3 1 6 1 “Name” 0 5 0
Content Identifier-Logical Content Information	3 1 6 1 0 “Miss Aude HEA Document Architect”

Table B.3 – Presentation styles for Figure B.8

Presentation Style Identifier Line Spacing	5 0 300

Presentation Style Identifier
First Line Offset
Line Spacing
	5 1
	1417
	300

Presentation Style Identifier First Line Offset Alignment Line Spacing	5 2 1417 JUSTIFIED 300
Presentation Style Identifier First Line Offset Alignment Line Spacing	5 3 1020 JUSTIFIED 300
Presentation Style Identifier First Line Offset Alignment Line Spacing	5 4 1417 JUSTIFIED 400

B.5 Processable form document with generic logical structure and generic layout structure

B.5.1 Introduction

This clause illustrates how the specimen document may be interchanged in processable form with a generic logical structure and a generic layout structure to accompany the specific logical structure.

The generic logical structure facilitates subsequent editing of the document by a recipient. It may be used during the revision process to ensure that the document remains in conformance with a pre-defined structure, i.e. a document class.

The generic logical structure illustrated in this example (see Figure B.9) is one which may have been used to generate the specific logical structure shown in Figure B.8.

The generic layout structure is used to control the layout of a logically structured document and the imaging of a laid out document when it is applied to the document layout and imaging processes. The generic layout structure specifies what types of layout object may be created during the layout process and in what order they may be created. During the imaging process the generic layout structure provides for attributes that direct the imaging process and provides generic content to be imaged.

The generic layout structure illustrated in this example (see Figure B.10) will create an image of the document that is identical to that which would be produced by the specific layout structure described in the first example (see Figure B.7).

In order to lay out a logically structured document, each logical object description in that document must be related to a layout object description created by the document layout process using the generic layout structure. This is achieved by means of layout styles, each of which consists of a set of attributes called layout directive attributes. Each logical object description contains an attribute which relates a particular layout style to that object description. The attributes in the layout style then relate the logical object description to the appropriate layout object class description and guide the precise layout of the logical object during the document layout process.

Similarly, the document contains presentation styles, each of which contains a set of attributes called presentation attributes. These guide the layout and imaging of the content associated with the basic logical object descriptions within the document. As in the case of layout styles, a basic logical object description may contain a reference to a certain presentation style and this has the effect of associating a particular set of presentation attributes with that object description.

In this example, the references to layout and presentation styles are not contained within the attributes of the specific logical object descriptions but are contained within the attributes of the object class description corresponding to the specific logical object description. This arrangement may be used, for example, to reduce the number of coded bits required to interchange a document or to facilitate subsequent editing.

B.5.2 Generic logical structure

The generic logical structure, which is shown in Figure B.9, specifies the logical object descriptions that may occur in a corresponding specific logical structure and their permitted sequential orders. The sequential order of object descriptions in the specific logical structure is significant since it indicates the order in which the objects are to be processed by the document layout and imaging processes (see clauses 10 and 11).

Referring to Figure B.9, the following are the implications for any corresponding specific logical structure. The document logical root description, which has been given the name “letter”, consists of the subordinate composite logical object descriptions given the names “header” and “body”. Both of these object descriptions shall occur once only in any corresponding specific logical structure. The mnemonic symbol SEQ placed below the document level indicates that the sequential order of these composite object descriptions in any corresponding specific logical structure shall be “header” followed by “body” (i.e., in the order left to right as indicated in the diagram).

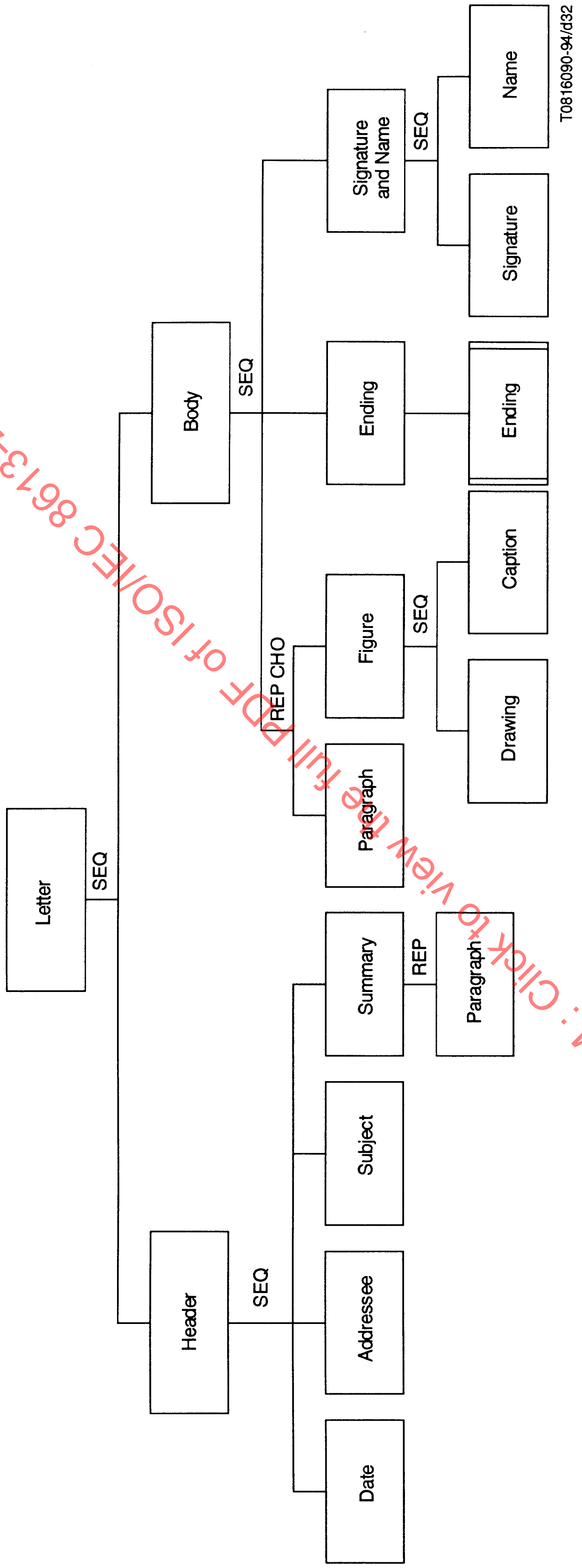


Figure B.9 – Generic logical structure

The object descriptions subordinate to "header" consist of "date", "addressee", "subject", and "summary". All of these shall occur once in any corresponding specific logical structure and shall occur in the order shown in the diagram. The "summary" is a composite logical object description consisting of one or more basic logical object descriptions "paragraph". Similarly, the composite object description "body" consists of any number and combination of the logical objects "paragraph" and "figure" (as indicated by the mnemonics REP, CHO) followed by object descriptions "ending" and "signature and name" which shall occur once only. There is no logical object "logo" because this is assumed to be part of the generic layout structure.

In this example, the two "paragraph" object class descriptions are distinct because different attribute values are associated with them. However, it would be possible, in a different example, for the object descriptions of "summary" and "body" both to refer to the same object class description "paragraph", which would result in a non-hierarchic generic layout structure.

The object class description "ending" contains a generic content portion description. Any specific logical structure generated from this generic logical structure would contain a logical object description corresponding to "ending" but no content portion description would be associated with this object description. However, since this logical object description would contain a reference to the object class description "ending" in the generic logical structure, the generic content portion description associated with this object class description would be considered to represent the content of the logical object description.

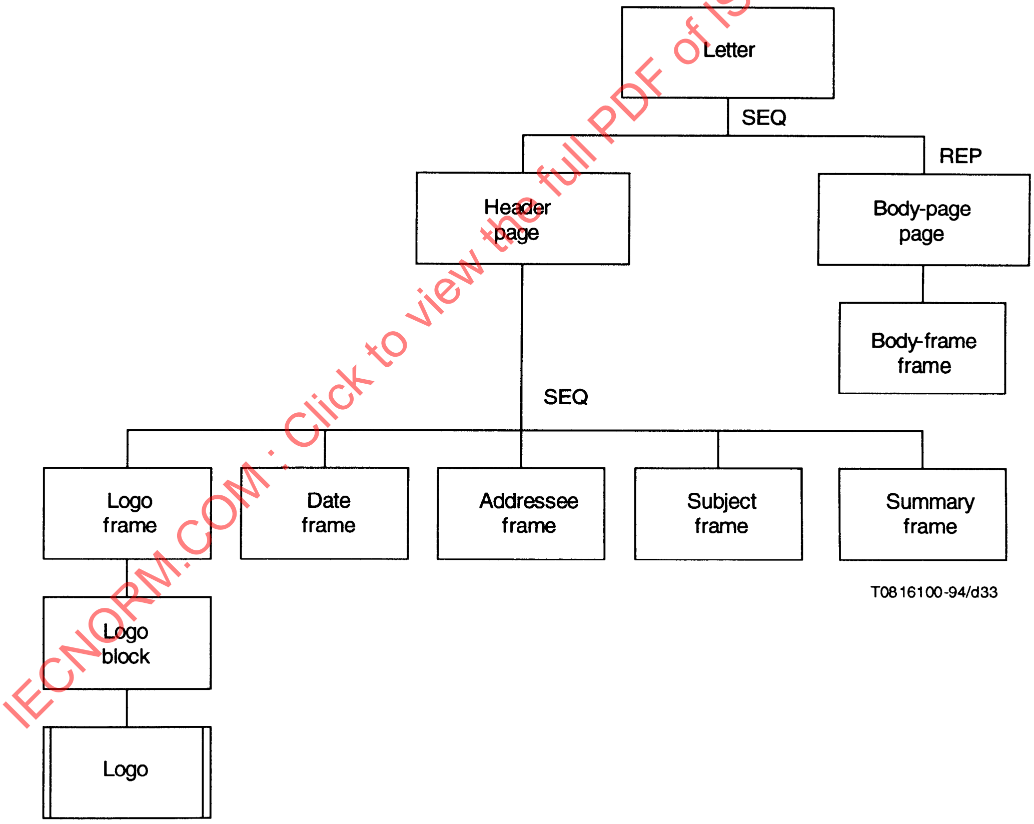


Figure B.10 – Generic layout structure

B.5.3 Specific logical structure

The specific logical structure included in this processable form document is identical to that shown in Figure B.8 with the single exception that the basic logical object description "ending" will not have a content portion description associated with it. As described in B.5.2, this content portion description will be obtained from the generic logical structure.

B.5.4 Generic layout structure

In the example shown in Figure B.10, the generic layout structure called "letter" indicates that a specific layout structure created by the document layout process shall consist of a "header" page followed by at least one "body-page" page.

The "header" page contains one generic content portion description for a "logo" and four layout object class descriptions for frames, "date", "addressee", "subject" and "summary".

Each "body-page" page contains one subordinate frame. The diagram indicates that the order of creation of the pages and their subordinate layout objects is significant.

The user-visible names of the layout object class descriptions used in this example have been chosen to indicate the correspondence between object classes in the generic and the specific layout structures.

B.5.5 Object descriptions and object class descriptions

The object class descriptions pertaining to the generic logical structure are given in Table B.4. Table B.5 describes the presentation styles and the layout styles associated with the document.

Table B.6 lists the object descriptions for the objects in the specific logical structure. The object descriptions in this table have to be interpreted together with the object class descriptions given in Table B.4. All content portion descriptions are grouped together since it is assumed that interchange format class A (see ITU-T Rec. T.415 | ISO/IEC 8613-5) will be used to interchange this particular document.

Table B.7 lists the object class descriptions pertaining to the generic layout structure.

B.6 Specific layout structure

Figure B.11 illustrates the specific layout structure generated by the document layout process (see clause 10) from the specific logical structure, generic logical structure and generic layout structure described in B.5.

If a document is in formatted processable form, then this structure would be generated by the originator and interchanged together with the other document structures.

This specific layout structure differs from that in B.4.1 in the following respects:

- the specific layout structure of B.4.1 contains no frames;
- the specific layout structure of B.4.1 contains a content portion description for the object description named "logo", whereas in Figure B.11 this content portion description is omitted because it is specified as part of the generic layout structure.

The object descriptions corresponding to Figure B.11 are listed in Table B.8. These are to be interpreted in conjunction with the layout object class descriptions listed in Table B.7. Also, note that in Table B.8, the presentation attributes associated with the content associated with various blocks are specified by means of presentation styles, which are listed in Table B.5. In a formatted-processable form document, the content portions in Table B.8 replace the ones described in Table B.6.

Figures B.12, B.13 and B.14 illustrate the layout structure of the specimen document corresponding to that described in Table B.8, by outlining the frames and blocks within each page.

Table B.4 – Constituents for Figure B.9 – Generic logical structure

Object Type Object Class Identifier User-Visible Name Generator for Subordinates	DOCUMENT LOGICAL ROOT 2(Letter) "Letter" SEQ (Header, Body)
Object Type Object Class Identifier User-Visible Name Generator for Subordinates	COMPOSITE LOGICAL 2 0 (Header) "Header" SEQ (Date, Addressee, Subject, Summary)
Object Type Object Class Identifier User-Visible Name Layout Style Content Architecture Class	BASIC LOGICAL 2 0 0 (Date) "Date" 4 0 PROCESSABLE CHARACTERS
Object Type Object Class Identifier User-Visible Name Layout Style Content Architecture Class	BASIC LOGICAL 2 0 1 (Addressee) "Addressee" 4 1 PROCESSABLE CHARACTERS
Object Type Object Class Identifier User-Visible Name Layout Style Presentation Style Content Architecture Class	BASIC LOGICAL 2 0 2 (Subject) "Subject" 4 2 5 0 PROCESSABLE CHARACTERS
Object Type Object Class Identifier User-Visible Name Layout Style Generator for Subordinates	COMPOSITE LOGICAL 2 0 3 (Summary) "Summary" 4 3 REP Summary-paragraph
Object Type Object Class Identifier User-Visible Name Layout Style Presentation Style Content Architecture Class	BASIC LOGICAL 2 0 3 1 (Summary-paragraph) "Summary-paragraph" 4 4 5 1 PROCESSABLE CHARACTERS
Object Type Object Class Identifier User-Visible Name Layout Style Generator for Subordinates	COMPOSITE LOGICAL 2 1 (Body) "Body" 4 5 SEQ (REP CHO (Body-paragraph, Figure), Ending, Signature-and-Name)
Object Type Object Class Identifier User-Visible Name Layout Style Generator for Subordinates	COMPOSITE LOGICAL 2 1 0 (Figure) "Figure" 4 6 SEQ (Drawing, Caption)
Object Type Object Class Identifier User-Visible Name Content Architecture Class Layout Style	BASIC LOGICAL 2 1 0 0 (Drawing) "Drawing" FORMATTED PROCESSABLE GEOMETRIC GRAPHICS 4 7
Object Type Object Class Identifier User-Visible Name Layout Style Content Architecture Class	BASIC LOGICAL 2 1 0 1 (Caption) "Caption" 4 8 PROCESSABLE CHARACTERS

Table B.4 (continued)

Object Type Object Class Identifier User-Visible Name Layout Style Presentation Style Content Architecture Class	BASIC LOGICAL 2 1 1 (Body-paragraph) "Body-paragraph" 4 9 5 3 PROCESSABLE CHARACTERS
Object Type Object Class Identifier User-Visible Name Layout Style Presentation Style Content Portions Content Architecture Class	BASIC LOGICAL 2 1 2 (Ending) "Ending" 4 10 5 4 0 (Ending-content) PROCESSABLE CHARACTERS
Content Identifier-Logical Content Information	2 1 2 0 (Ending-content) /*Unformatted string for ending*/
Object Type Object Class Identifier User-Visible Name Generator for Subordinates	COMPOSITE LOGICAL 2 1 3 (Signature-and-Name) "Signature-and-Name" SEQ (Signature, Name)
Object Type Object Class Identifier User-Visible Name Content Architecture Class Layout Style	BASIC LOGICAL 2 1 3 0 (Signature) "Signature" PROCESSABLE FORM RASTER GRAPHICS 4 11
Object Type Object Class Identifier User-Visible Name Layout Style Presentation Style Content Architecture Class	BASIC LOGICAL 2 1 3 1 (Name) "Name" 4 12 5 0 PROCESSABLE CHARACTERS

Table B.5 – Styles for Figure B.9 – Generic logical structure

Presentation Style Identifier Line Spacing	5 0 300
Presentation Style Identifier First Line Offset Alignment	5 1 1417 JUSTIFIED
Presentation Style Identifier First Line Offset Alignment Line Spacing	5 3 1417 JUSTIFIED 300
Presentation Style Identifier First Line Offset Alignment Line Spacing	5 4 1020 JUSTIFIED 300
Layout Style Identifier Layout Object Class Offset	4 0 0 0 1 (Date) TRAILING = 710, RIGHT-HAND = 395

Table B.5 (end)

Layout Style Identifier Layout Object Class	4 1 0 0 2 (Addressee)
Layout Style Identifier Layout Object Class	4 2 0 0 3 (Subject)
Layout Style Identifier Layout Object Class	4 3 0 0 4 (Summary)
Layout Style Identifier Offset	4 4 LEFT-HAND = 705
Layout Style Identifier New Layout Object	4 5 (Body-Page)
Layout Style Identifier Indivisibility	4.6 0 1 0 (Body-Frame)

Layout Style Identifier
Offset
Separation
	4 7
	RIGHT-HAND = 1615, LEFT-HAND = 2155
	TRAILING = 905

Layout Style Identifier
Offset
Separation
	4 8
	RIGHT-HAND = 1985, LEFT-HAND = 2860
	TRAILING = 200

Layout Style Identifier
Offset
Separation
	4 9
	TRAILING = 540, LEADING = 280, RIGHT-HAND = 540, LEFT-HAND = 340
	TRAILING = 880

Layout Style Identifier
Offset
Separation
	4 10
	RIGHT-HAND = 1420, LEFT-HAND = 535
	TRAILING = 880

Layout Style Identifier
Offset
Separation
	4 11
	RIGHT-HAND = 2695, LEFT-HAND = 535
	TRAILING = 765

Layout Style Identifier Offset	4 12 RIGHT-HAND = 5385, LEFT-HAND = 910

Table B.6 – Constituents for specific logical structure

Object Type Object Identifier Object Class User-Visible Name Subordinates	DOCUMENT LOGICAL ROOT 3 2 (Letter) “Letter” 0, 1
Object Type Object Identifier Object Class User-Visible Name Subordinates	COMPOSITE LOGICAL 3 0 2 0 (Header) “Header” 0, 1, 2, 3
Object Type Object Identifier Object Class User-Visible Name Content Portions	BASIC LOGICAL 3 0 0 2 0 0 (Date) “Date” 0

Table B.6 (continued)

Object Type	BASIC LOGICAL
Object Identifier	3 0 1
Object Class	2 0 1 (Addressee)
User-Visible Name	"Addressee"
Content Portions	0
Object Type	BASIC LOGICAL
Object Identifier	3 0 2
Object Class	2 0 2 (Subject)
User-Visible Name	"Subject"
Content Portions	0
Object Type	COMPOSITE LOGICAL
Object Identifier	3 0 3
Object Class	2 0 3 (Summary)
User-Visible Name	"Summary"
Subordinates	0
Object Type	BASIC LOGICAL
Object Identifier	3 0 3 0
Object Class	2 0 3 1 (Summary-paragraph)
User-Visible Name	"Summary-paragraph"
Content Portions	0
Object Type	COMPOSITE LOGICAL
Object Identifier	3 1
Object Class	2 1 (Body)
User-Visible Name	"Body"
Subordinates	0, 1, 2, 3, 4, 5, 6
Object Type	BASIC LOGICAL
Object Identifier	3 1 0
Object Class	2 1 1 (Body-paragraph)
User-Visible Name	Paragraph A
Content Portions	0
Object Type	BASIC LOGICAL
Object Identifier	3 1 1
Object Class	2 1 1 (Body-paragraph)
User-Visible Name	Paragraph B
Line Spacing	400
Content Portions	0
Object Type	COMPOSITE LOGICAL
Object Identifier	3 1 2
Object Class	2 1 0 (Figure)
User-Visible Name	"Figure"
Subordinates	0, 1
Object Type	BASIC LOGICAL
Object Identifier	3 1 2 0
Object Class	2 1 0 0 (Drawing)
User-Visible Name	"Drawing"
Content Portions	0
Object Type	BASIC LOGICAL
Object Identifier	3 1 2 1
Object Class	2 1 0 1 (Caption)
User-Visible Name	"Caption"
Content Portions	0
Object Type	BASIC LOGICAL
Object Identifier	3 1 3
Object Class	2 1 1 (Body-paragraph)
User-Visible Name	Paragraph C
Content Portions	0

Table B.6 (end)

Object Type Object Identifier Object Class User-Visible Name Content Portions	BASIC LOGICAL 3 1 4 2 1 1 (Body-paragraph) Paragraph D 0
Object Type Object Identifier Object Class User-Visible Name	BASIC LOGICAL 3 1 5 2 1 2 (Ending) "Ending"
Object Type Object Identifier Object Class User-Visible Name Subordinates	COMPOSITE LOGICAL 3 1 6 2 1 3 (Signature and Name) "Signature and Name" 0, 1
Object Type Object Identifier Object Class User-Visible Name Content Portions	BASIC LOGICAL 3 1 6 0 2 1 3 0 (Signature) "Signature" 0
Object Type Object Identifier Object Class User-Visible Name Content Portions	BASIC LOGICAL 3 1 6 1 2 1 3 1 (Name) "Name" 0
Content Identifier – Logical Content Information	3 0 0 0 "CESSON, 26 JUNE 1985"
Content Identifier – Logical Content Information	3 0 1 0 "To members of ISO/TC97/SC18/WG3"
Content Identifier – Logical Content Information	3 0 2 0 "SUBJECT: PROPOSED EXAMPLE TO CLARIFY THE DOCUMENT ARCHITECTURE MODEL"
Content Identifier – Logical Content Information	3 0 3 0 0 /*Unformatted string of SUMMARY-*/
Content Identifier – Logical Content Information	3 1 0 0 /*Unformatted string of A's*/
Content Identifier – Logical Content Information	3 1 1 0 /*Unformatted string of B's*/
Content Identifier – Logical Content Information	3 1 2 0 0 /*Ordered set of geometric-graphics content elements for the diagram*/
Content Identifier – Logical Content Information	3 1 2 1 0 /*Unformatted string for the caption*/
Content Identifier – Logical Content Information	3 1 3 0 /*Unformatted string of C's*/
Content Identifier – Logical Content Information	3 1 4 0 /*Unformatted string of D's*/
Content Identifier – Logical Content Information	3 1 6 0 0 /*Array of raster-graphics content elements for the signature*/
Content Identifier – Logical Content Information	3 1 6 1 0 "Miss Aude HEA Document Architect"

Table B.7 – Constituents for Figure B.10 – Generic layout structure

Object Type Object Class Identifier User-Visible Name Generator for Subordinates	DOCUMENT LAYOUT ROOT 0 (Letter) "Letter" SEQ (Header, REP Body-Page)
Object Type Object Class Identifer User-Visible Name Dimensions Generator for Subordinates	PAGE 0 0 (Header) "Header" HD = 9920, VD = 14030 SEQ (Logo Frame, Date, Addressee, Subject, Summary)
Object Type Object Class Identifier Position Dimensions Generator for Subordinates	FRAME 0 0 0 (Logo Frame) HP = 710, VP = 730 HD = 3685, VD = 2495 Logo
Object Type Object Class Identifier User-Visible Name Content Architecture Class Content Portions	BLOCK 0 0 0 0 (Logo) "Logo" FORMATTED FORM RASTER GRAPHICS 0 (Logo)
Object Type Object Class Identifier User-Visible Name Position Dimensions	FRAME 0 0 1 (Date) "Date" HP = 5045, VP = 565 HD = 3970, VD = 1615
Object Type Object Class Identifier User-Visible Name Position Dimensions	FRAME 0 0 2 (Addressee) "Addressee" HP = 1105, VP = 4310 HD = 5395, VD = 1415
Object Type Object Class Identifier User-Visible Name Position Dimensions	FRAME 0 0 3 (Subject) "Subject" HP = 1105, VP = 6660 HD = 7200, VD = 1785
Object Type Object Class Identifier User-Visible Name Position Dimensions	FRAME 0 0 4 (Summary) "Summary" HP = 2180, VP = 9695 HD = 6290, VD = 3570
Object Type Object Class Identifier User-Visible Name Dimensions Generator for Subordinates	PAGE 0 1 (Body-Page) "Body" HD = 9920, VD = 14030 Body-Frame
Object Type Object Class Identifier User-Visible Name Position Dimensions	FRAME 0 1 0 (Body-Frame) "Body" HP = 565, VP = 565 HD = 8815, VD = 12870
Content Identifier – Layout Content Information	0 0 0 0 (Logo) /*Array of raster-graphics content elements for the logo*/

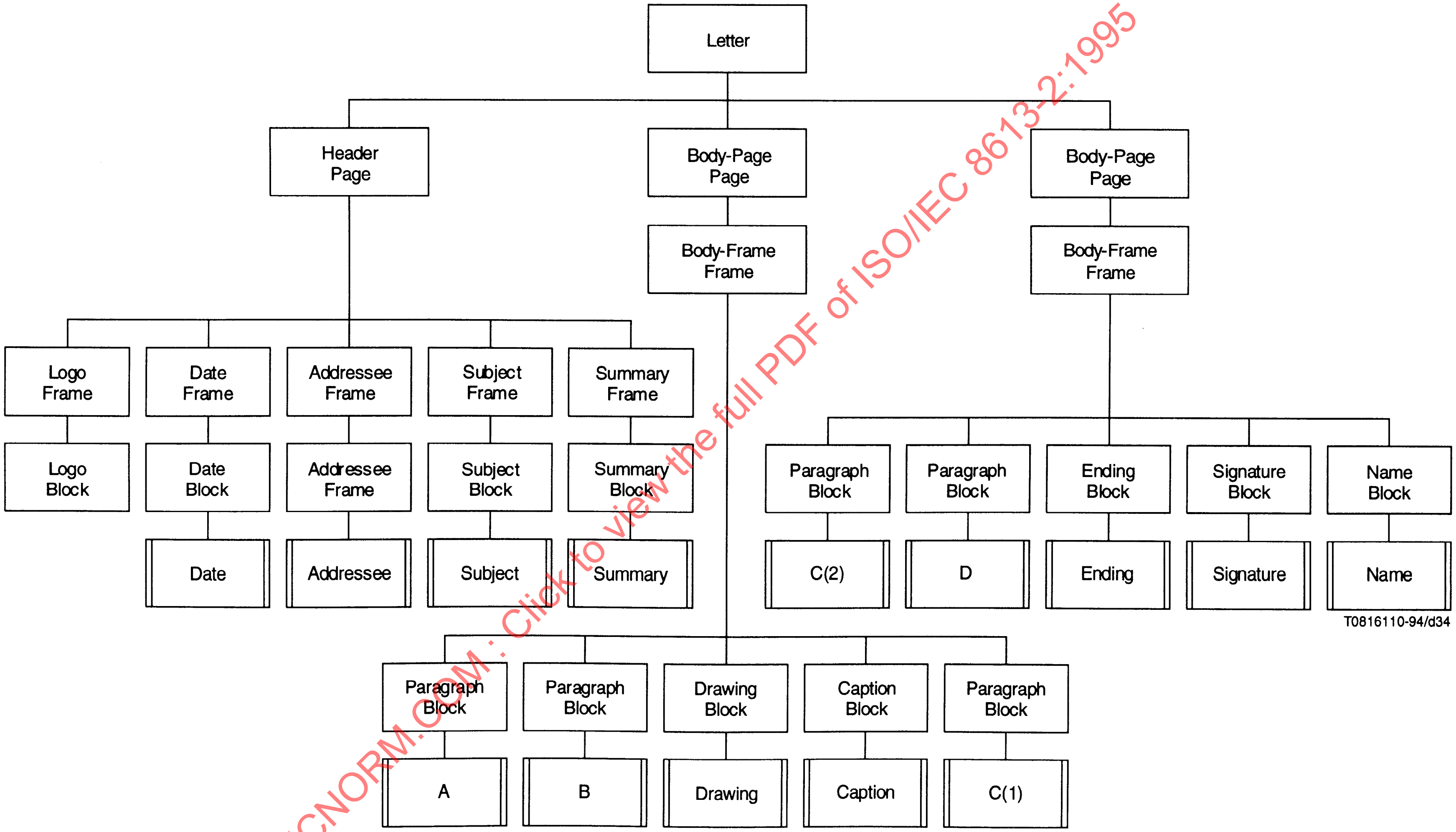


Figure B.11 – Specific layout structure

Table B.8 – Constituents for Figure B.11 – Specific layout structure

Object Type Object Identifier Object Class User-Visible Name Subordinates	DOCUMENT LAYOUT ROOT 1 0 (Letter) “Letter” 0, 1, 2
Object Type Object Identifier Object Class User-Visible Name Subordinates	PAGE 1 0 0 0 (Header) “Header” 0, 1, 2, 3, 4
Object Type Object Identifier Object Class Subordinates	FRAME 1 0 0 0 0 0 (Logo Frame) 0
Object Type Object Identifier Object Class User-Visible Name Content Architecture Class	BLOCK 1 0 0 0 0 0 0 0 (Logo) “Logo” FORMATTED FORM RASTER GRAPHICS
Object Type Object Identifier Object Class User-Visible Name Subordinates	FRAME 1 0 1 0 0 1 (Date) “Date” 0
Object Type Object Identifier Position Dimensions Content Portions Content Architecture Class	BLOCK 1 0 1 0 HP = 395, VP = 710 HD = 3060, VD = 540 0 FORMATTED PROCESSABLE CHARACTERS
Object Type Object Identifier Object Class User-Visible Name Subordinates	FRAME 1 0 2 0 0 2 (Addressee) “Addressee” 0
Object Type Object Identifier Dimensions Content Portions Content Architecture Class	BLOCK 1 0 2 0 HD = 4505, VD = 540 0 FORMATTED PROCESSABLE CHARACTERS
Object Type Object Identifier Object Class User-Visible Name Subordinates	FRAME 1 0 3 0 0 3 (Subject) “Subject” 0
Object Type Object Identifier Dimensions Presentation Style Content Portions Content Architecture Class	BLOCK 1 0 3 0 HD = 7200, VD = 905 5 0 0 FORMATTED PROCESSABLE CHARACTERS

Table B.8 (continued)

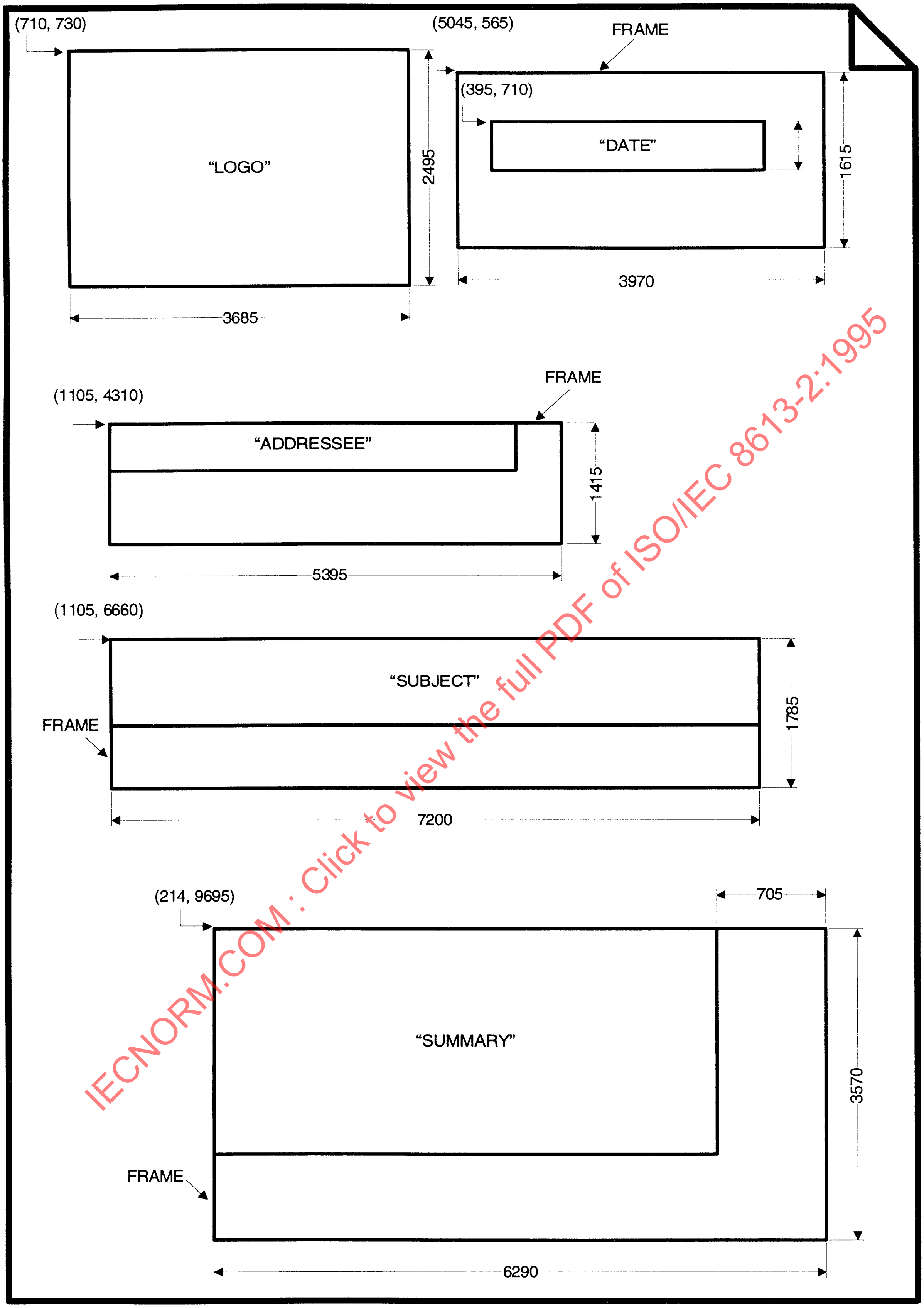
Object Type Object Identifier Object Class User-Visible Name Subordinates	FRAME 1 0 4 0 0 4 (Summary) “Summary” 0
Object Type Object Identifier Dimensions Presentation Style Content Portions Content Architecture Class	BLOCK 1 0 4 0 HD = 5585, VD = 2325 5 1 0 FORMATTED PROCESSABLE CHARACTERS
Object Type Object Identifier Object Class User-Visible Name Subordinates	PAGE 1 1 0 1 (Body-Page) “Body” 0
Object Type Object Identifier Object Class User-Visible Name Subordinates	FRAME 1 1 0 0 1 0 (Body-Frame) “Body” 0, 1, 2, 3, 4
Object Type Object Identifier Position Dimensions Presentation Style Content Portions Content Architecture Class	BLOCK 1 1 0 0 HP = 540, VP = 540 HD = 7935, VD = 1785 5 3 0 FORMATTED PROCESSABLE CHARACTERS
Object Type Object Identifier Position Dimensions Presentation Style Line Spacing Content Portions Content Architecture Class	BLOCK 1 1 0 1 HP = 540, VP = 3205 HD = 7935, VD = 1785 5 3 400 0 FORMATTED PROCESSABLE CHARACTERS
Object Type Object Identifier Position Dimensions Content Architecture Class Content Portions	BLOCK 1 1 0 2 HP = 1615, VP = 5895 HD = 5045, VD = 4140 FORMATTED PROCESSABLE GEOMETRIC GRAPHICS 0
Object Type Object Identifier Position Dimensions Content Portions Content Architecture Class	BLOCK 1 1 0 3 HP = 1985, VP = 10235 HD = 3970, VD = 370 0 FORMATTED PROCESSABLE CHARACTERS
Object Type Object Identifier Position Dimensions Presentation Style Content Portions Content Architecture Class	BLOCK 1 1 0 4 HP = 540, VP = 11485 HD = 7935, VD = 1075 5 3 0 FORMATTED PROCESSABLE CHARACTERS

Table B.8 (continued)

Object Type Object Identifier Object Class User-Visible Name Subordinates	PAGE 1 2 0 1 (Body-Page) "Body" 0
Object Type Object Identifier Object Class User-Visible Name Subordinates	FRAME 1 2 0 0 1 0 (Body-Frame) "Body" 0, 1, 2, 3, 4
Object Type Object Identifier Position Dimensions Presentation Style Content Portions Content Architecture Class	BLOCK 1 2 0 0 HP = 540, VP = 540 HD = 7935, VD = 1275 5 3 0 FORMATTED PROCESSABLE CHARACTERS
Object Type Object Identifier Position Dimensions Presentation Style Content Portions Content Architecture Class	BLOCK 1 2 0 1 HP = 540, VP = 2695 HD = 7935, VD = 1615 5 3 0 FORMATTED PROCESSABLE CHARACTERS
Object Type Object Identifier Position Dimensions Presentation Style Content Portions Content Architecture Class	BLOCK 1 2 0 2 HP = 1820, VP = 5190 HD = 6860, VD = 2155 5 4 0 FORMATTED PROCESSABLE CHARACTERS
Object Type Object Identifier Position Dimensions Content Type Content Portions	BLOCK 1 2 0 3 HP = 2695, VP = 8110 HD = 5585, VD = 2495 FORMATTED PROCESSABLE RASTER GRAPHICS 0
Object Type Object Identifier Position Dimensions Presentation Style Content Portions Content Architecture Class	BLOCK 1 2 0 4 HP = 5385, VP = 10605 HD = 2520, VD = 905 5 0 0 FORMATTED PROCESSABLE CHARACTERS
Content Identifier – Layout Content Identifier – Logical Content Information	1 0 1 0 0 3 0 0 0 "CESSON, 26 JUNE 1985"
Content Identifier – Layout Content Identifier – Logical Content Information	1 0 2 0 0 3 0 1 0 "To members of ISO/TC97/SC18/WG3"
Content Identifier – Layout Content Identifier – Logical Content Information	1 0 3 0 0 3 0 2 0 "SUBJECT: PROPOSED EXAMPLE TO CLARIFY THE DOCUMENT\n ARCHITECTURE MODEL"

Table B.8 (end)

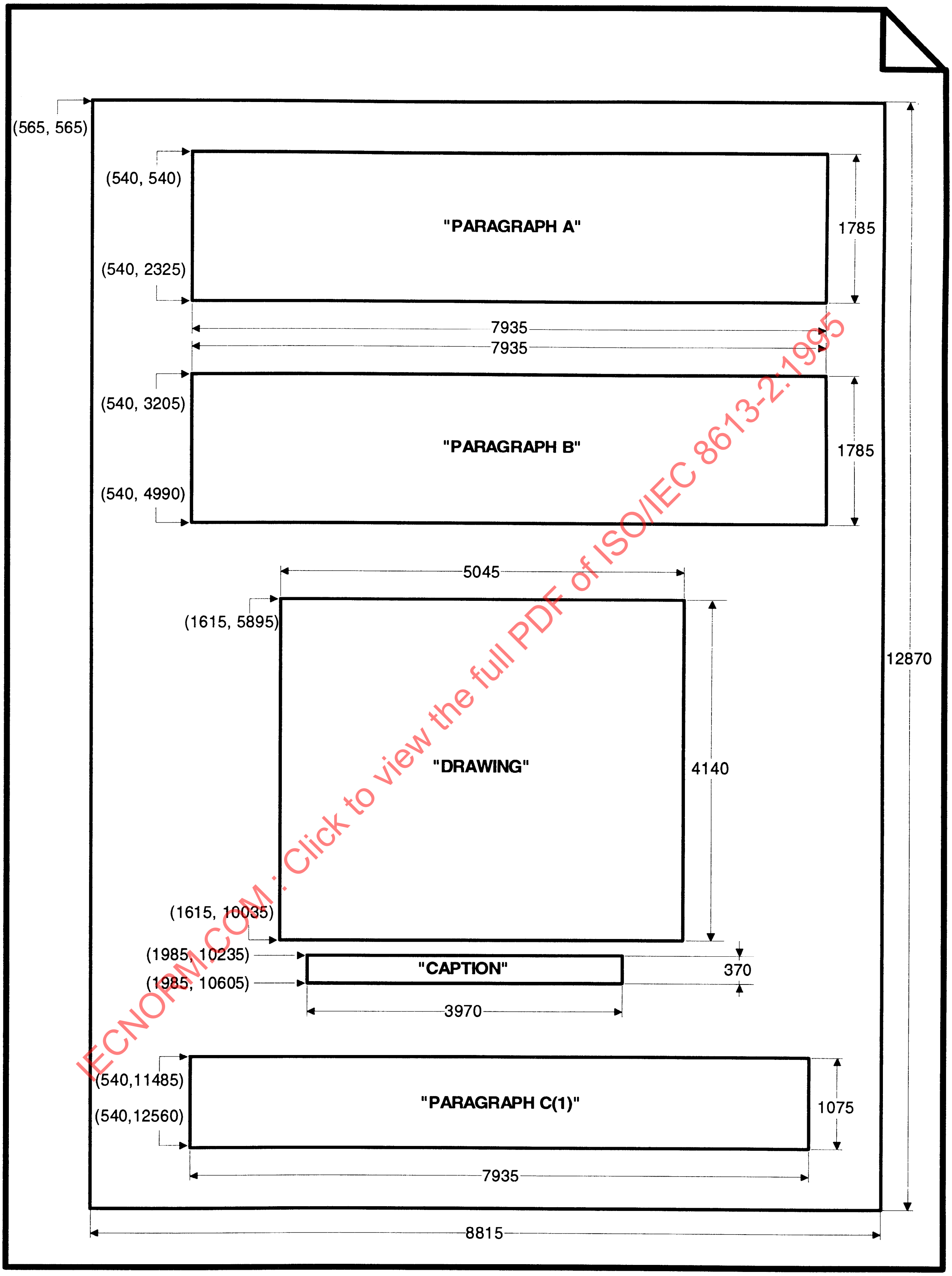
Content Identifier – Layout Content Identifier – Logical Content Information	1 0 4 0 0 3 0 3 0 0 /*Formatted string of SUMMARY-*/
Content Identifier – Layout Content Identifier – Logical Content Information	1 1 0 0 0 3 1 0 0 /*Formatted string of A's*/
Content Identifier – Layout Content Identifier – Logical Content Information	1 1 0 1 0 3 1 1 0 /*Formatted string of B's*/
Content Identifier – Layout Content Identifier – Logical Content Information	1 1 0 2 0 3 1 2 0 0 /*Ordered set of geometric-graphics content elements for the diagram*/
Content Identifier – Layout Content Identifier – Logical Content Information	1 1 0 3 0 3 1 2 1 0 /*Formatted string for the caption*/
Content Identifier – Layout Content Identifier – Logical Content Information	1 1 0 4 0 3 1 3 0 /*Formatted string of C's*/
Content Identifier – Layout Content Identifier – Logical Content Information	1 2 0 0 0 3 1 3 1 /*Formatted string of C's*/
Content Identifier – Layout Content Identifier – Logical Content Information	1 2 0 1 0 3 1 4 0 /*Formatted string of D's*/
Content Identifier – Layout Content Information	1 2 0 2 0 /*Formatted string of ENDING*/
Content Identifier – Layout Content Identifier – Logical Content Information	1 2 0 3 0 3 1 6 0 0 /*Array of raster-graphics content elements for the signature*/
Content Identifier – Layout Content Identifier – Logical Content Information	1 2 0 4 0 3 1 6 1 0 “Miss Aude HEA\nDocument Architect”



HEADER PAGE

T0816120-94/d35

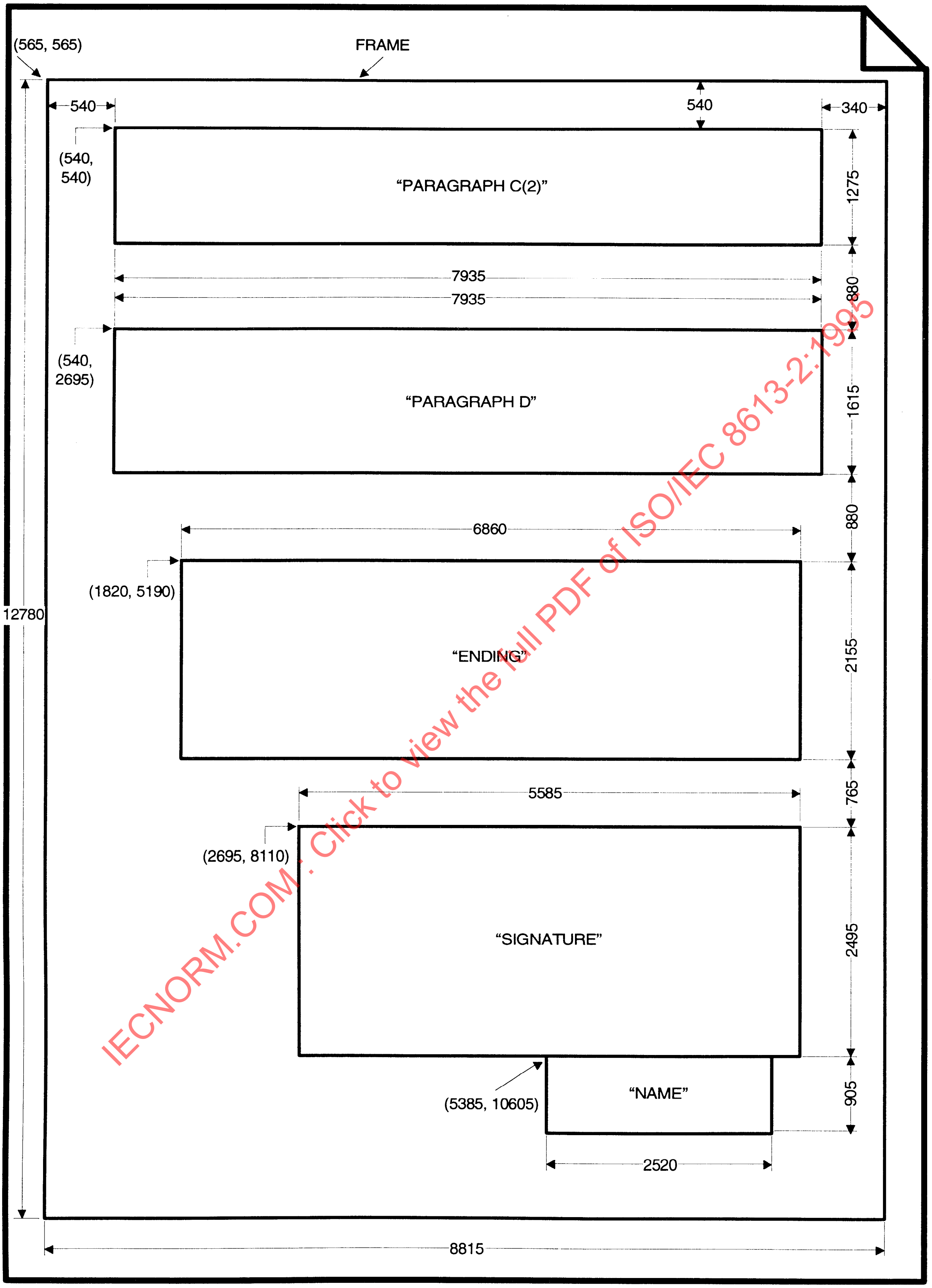
Figure B.12 – Layout structure of "letter" showing "pages", "frames" and "blocks" (1)



FIRST BODY PAGE

T0820000-94/d36

Figure B.13 – Layout structure of "letter" showing "pages", "frames" and "blocks" (2)



T0816140-94/d37

Figure B.14 – Layout structure of "letter" showing "pages", "frames" and "blocks" (3)

Annex C

Examples of particular document architecture features

(This annex does not form an integral part of this Recommendation | International Standard)

C.1 Layout of frames

This clause presents examples of the use of the attributes "position", "dimensions" and "layout path" as applicable to frames.

C.1.1 Conventions

Page, frame and block names are specified in their top left corner indicated by P for pages, F for frames and B for blocks, respectively, indexed by a sequence of digits. Frames or blocks subordinate to pages or other frames, in general, have the same initial sequence of digits as their superior object, for example, F11 is immediately subordinate to P1.

Layout path is illustrated by an arrow beside the frame name.

A variable dimension is indicated by double lines.

The values of the attributes are indicated inside the outline of the layout object by short forms:

- HP – Horizontal position, the value is either a constant, for the sub-parameter "horizontal position", indicated by notations of the form "0", "B", "C", "2C", or is derived from the parameter "variable position", indicated by notations of the form "alignment = centre" or "left-hand offset = C".
- VP – Vertical position, the value is either a constant, for the sub-parameter "vertical position", indicated by notations of the form "0", "A", "B", or is derived from the parameter "variable position", indicated by notations of the form "fill order = reverse order" or "leading edge separation = A".
- HD – Horizontal dimension, the value is either the default value defined in this Specification indicated by the notation "default", or a constant value for the parameter "horizontal dimension", indicated by notations of the form "D", or is derived from a rule, indicated by notations of the form "Rule B", where the rules are as defined in 9.4.1.2.
- VD – Vertical dimension, the value is either the default value defined in this Specification indicated by the notation "default", or a constant value for the parameter "vertical dimension", indicated by notations of the form "D", or is derived from a rule, indicated by notations of the form "Rule A" or "Rule B", where the rules are as defined in 9.4.1.2.

Content of blocks is indicated by annotation of the form "<...>"

The notation adopted for the specification of construction expressions is that specified in A.2.2.

C.1.2 Position attribute for a frame in a fixed position

See Figure C.1.

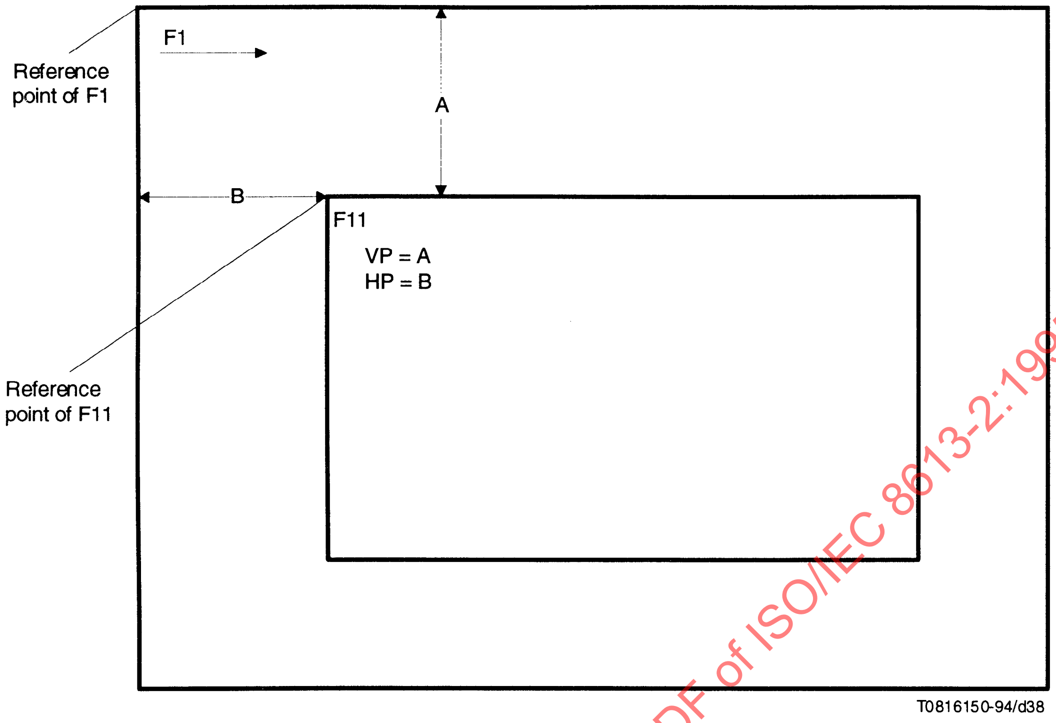
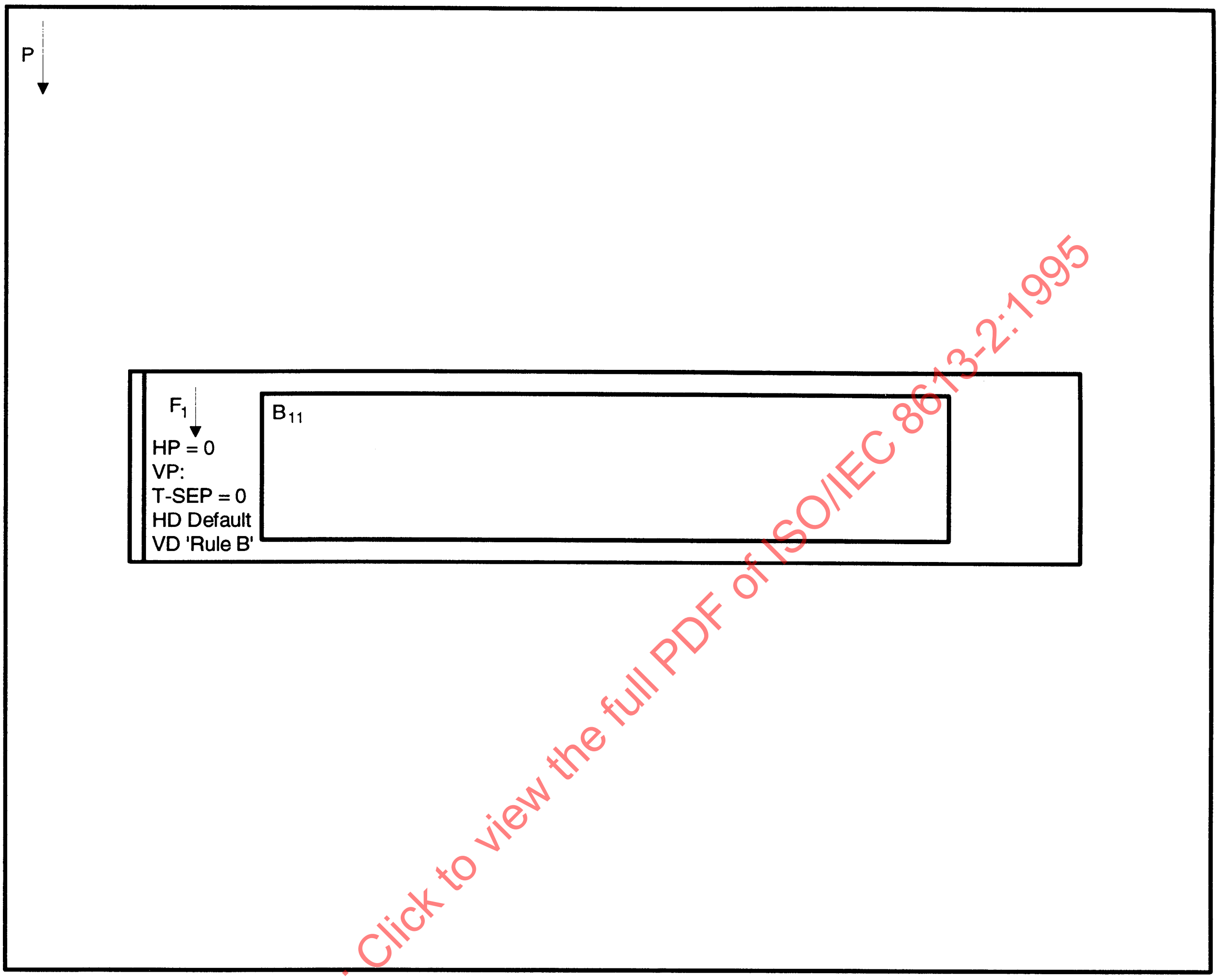


Figure C.1 – Fixed position frame

C.1.3 Single column text of variable height

Informal description of feature illustrated:

Single column text on a page, positioned after any other material already laid out on the page. The frame is to be generated just sufficiently large as is required to hold the content, with the intention possibly to include other forms of layout on the same page. This may be used to hold a paragraph of text or a figure. For an illustration, Figure C.2.



T08161 60-94/d39

Figure C.2 – Single column text of variable height

Generator for subordinates:

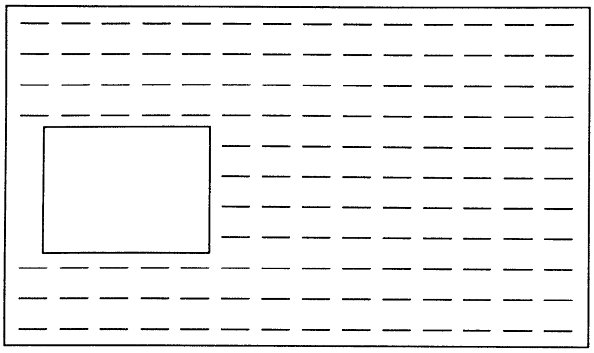
P SEQ (CHO (F0 ...))

F0 None

C.1.4 "Wrapping" of text around a left aligned picture

Informal description of feature illustrated:

Single column text on a page, the column being able to include a picture and to allow the "wrapping" of text around a picture. The picture to be placed at the left of the column.



T0816170-94/d40

Assumptions:

Picture in geometric graphics content architecture, the dimensions of the picture are specified by the presentation attribute "picture dimensions", specified in that content architecture.

The automatic case is not used, since the dimensions of the picture could then be set so as to use the whole of F1.

The picture specifies layout object class = F11 in order that each picture that occurs causes one instance of F1.

The picture comes earlier in sequential logical order than any of the text to be placed beside it.

Generator for subordinates:

P	SEQ (CHO (F0 F1))
F0	As in C.1.3
F1	SEQ (F11 F12)
F11, F12	None

For an illustration, see Figure C.3.

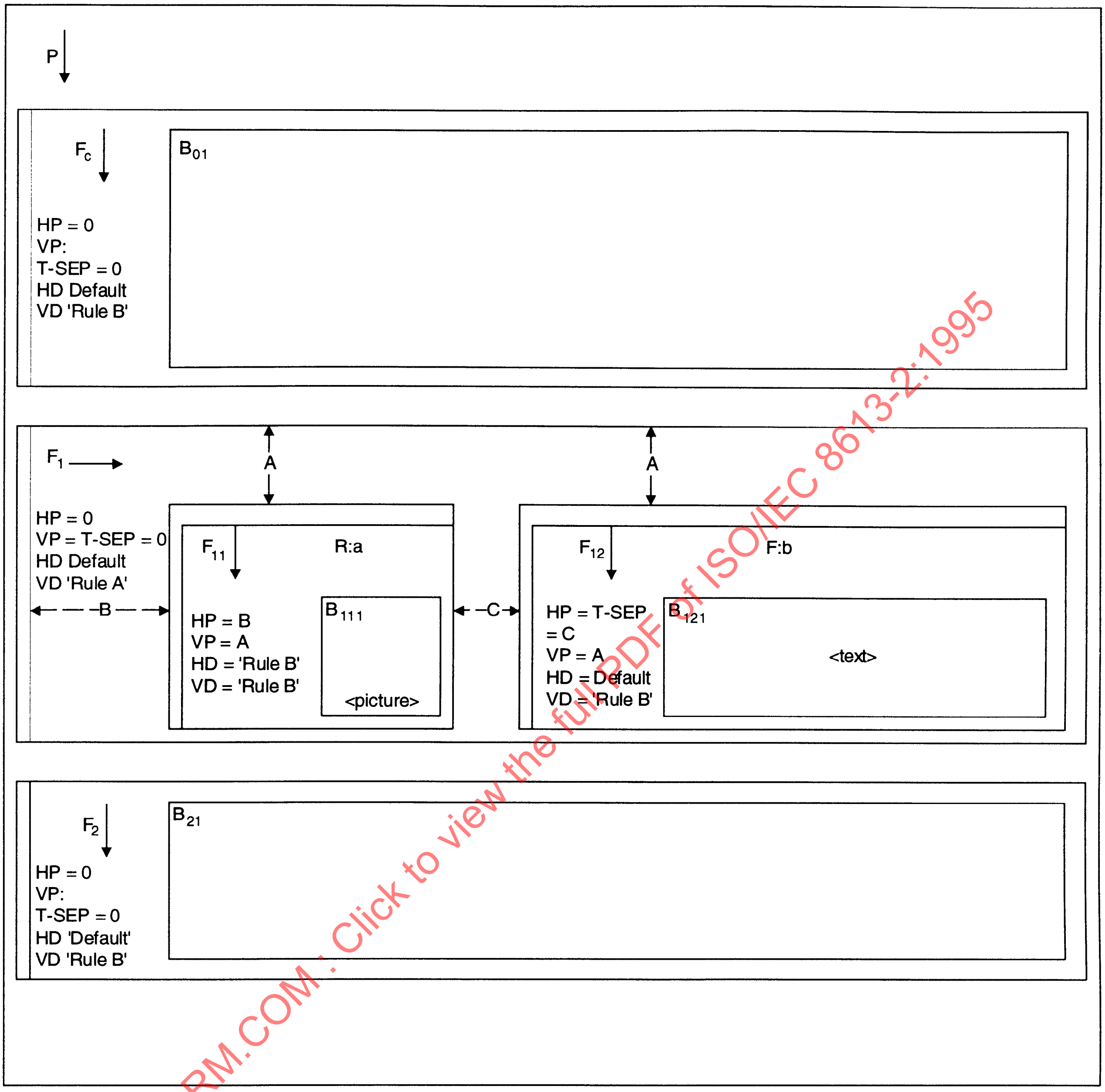
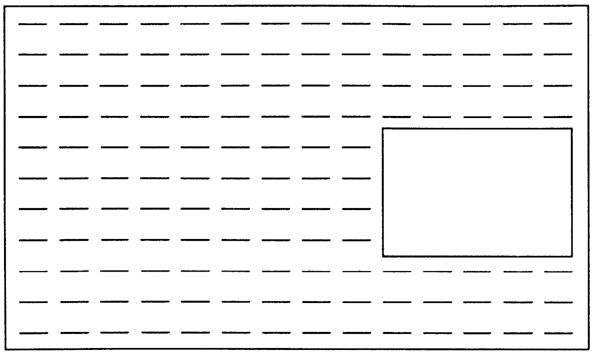


Figure C.3 – "Wrapping" of text around a left aligned picture

Variations:

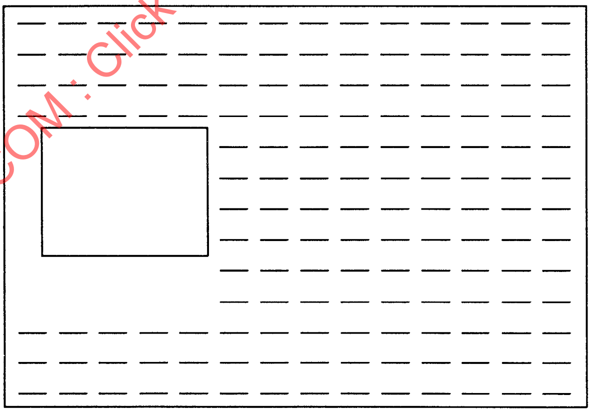
1) If the picture is required to be right aligned:



T0816190-94/d42

then the layout path for frame F1 would be reversed (180°). F11 is still the frame with the picture, the positions of F11 and F12 being interchanged, other parameters would be unchanged. The picture is to come earlier in sequential logical order than any of the text to be placed beside it.

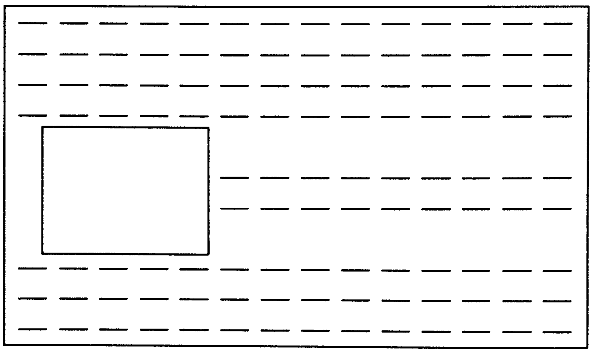
2) If the required layout were that the material to be placed beside the picture is to be associated with it and that the description of the figure is to be completed in the indented column, before resuming full width lines:



T0816200-94/d43

then the rule determining the vertical dimension of F1 would be changed to 'Rule B', other parameters would be unchanged.

3) If the text beside the picture is to be centred vertically in the area beside the picture:



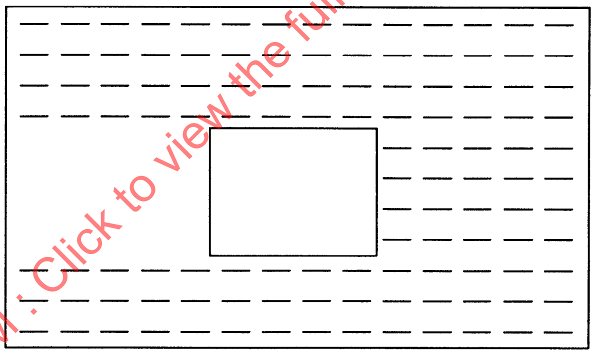
T0816210-94/d44

then the vertical position of F12 is specified by the sub-parameter "alignment" with the value 'centred', other parameters would be unchanged.

C.1.5 Picture with text on both sides

Informal description of feature illustrated:

Single column text on a page, with a picture with associated text on both sides of it.



T0816220-94/d45

Assumptions:

Picture in geometric graphics content architecture, the dimensions of the picture are specified by the presentation attribute "picture dimensions", specified in that content architecture.

The picture precedes the associated text in the sequential logical order.

The logical object containing the picture has the layout directive attribute "layout object class" F12 applied to it.

The text with the figure title is to be at the bottom of the figure on the left hand side. This is controlled by specifying for the logical object containing that text the layout directive attributes:

- layout object class = F11;
- fill order = reverse order.

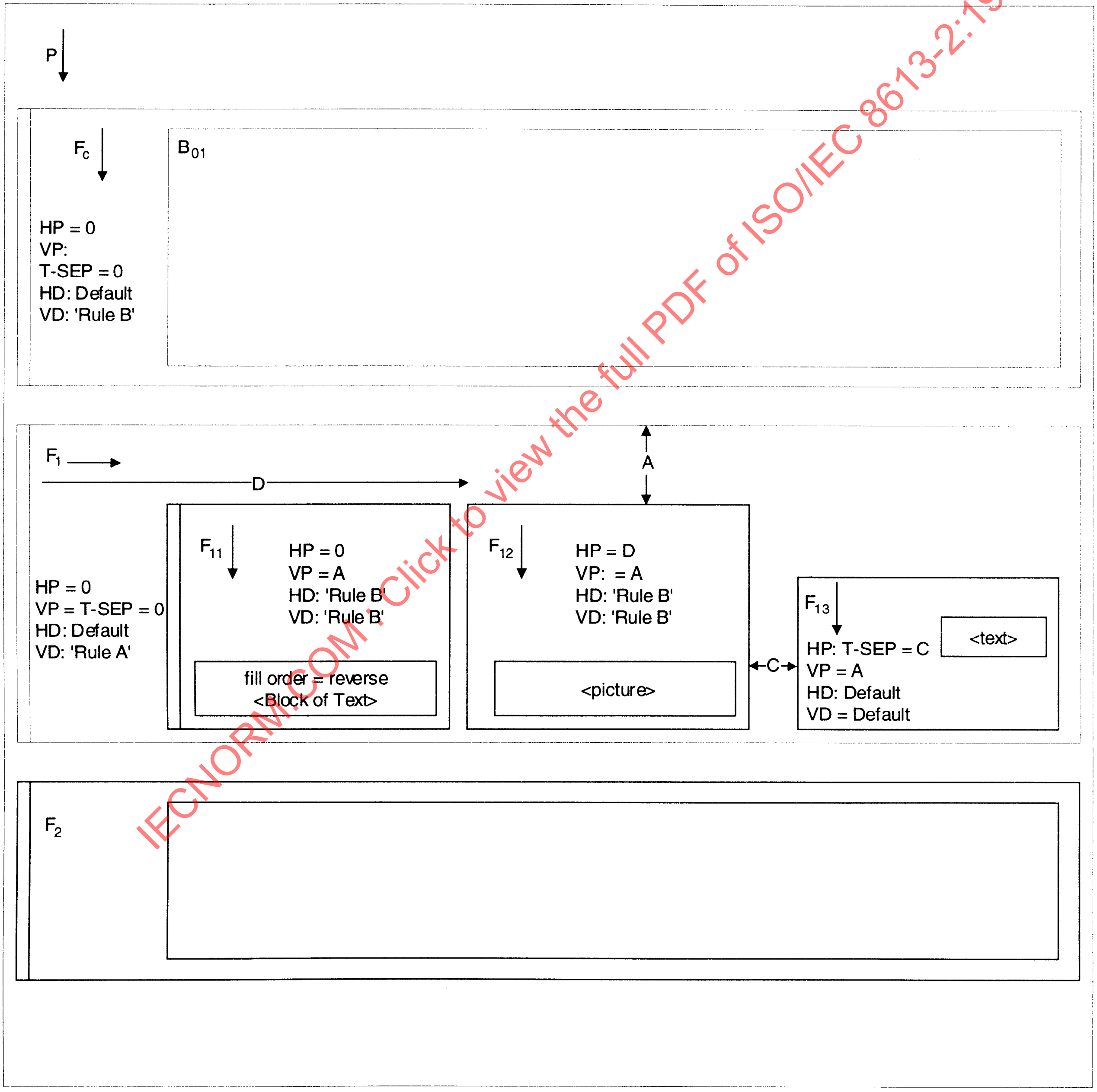
The text with other explanatory text associated with the figure is to be centred vertically on the right hand side of the figure. This is controlled by specifying for the logical object containing that text the layout directive attributes:

–	layout object class	= F13;
–	block alignment	= centred.

Generator for subordinates:

P	SEQ (CHO (F0 F1))
F0	As in C.1.3
F1	SEQ (F11 F12 F13)
F11, F12, F13	None.

For an illustration, see Figure C.4.

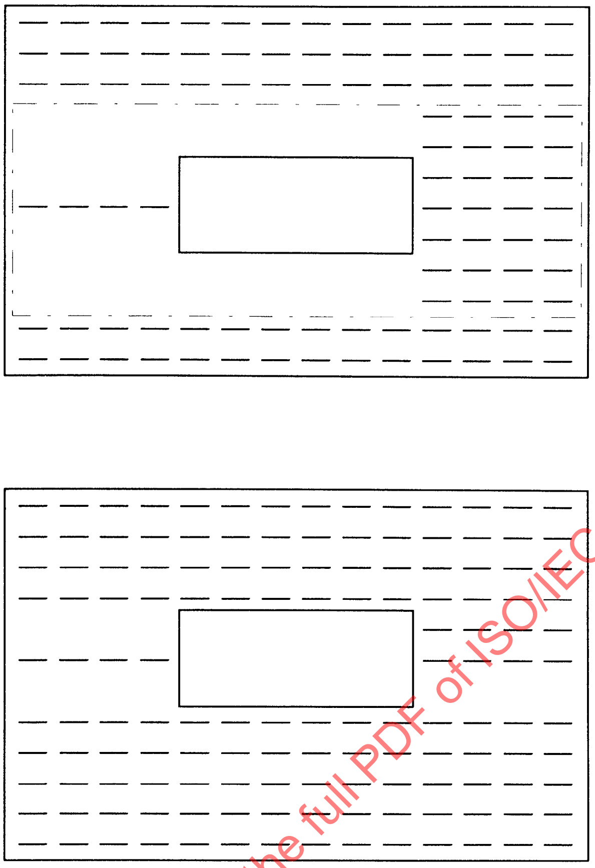


T0816230-94/d46

Figure C.4 – Horizontally centred picture with text on both sides

Variations:

1) The descriptive text on the right associated with the text might be longer or shorter than the figure.



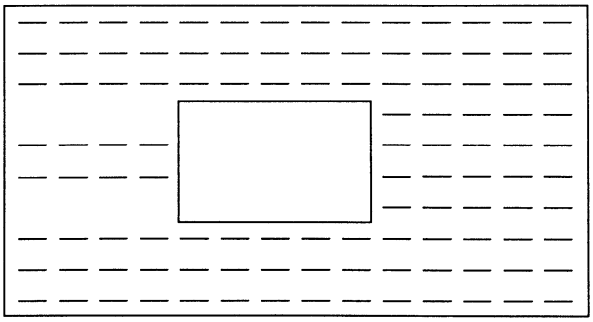
T0816240-94/d47

The following modifications to the specification will achieve this effect:

- F1 – 'Rule B' rather than 'Rule A';
- F11, F12 and F13 – Each specify vertical position by sub-parameter “alignment” with value 'centred';

The upper of the two pictures illustrates the case when the frame F1 specifies the attribute “border” for all edges; the lower picture has no border.

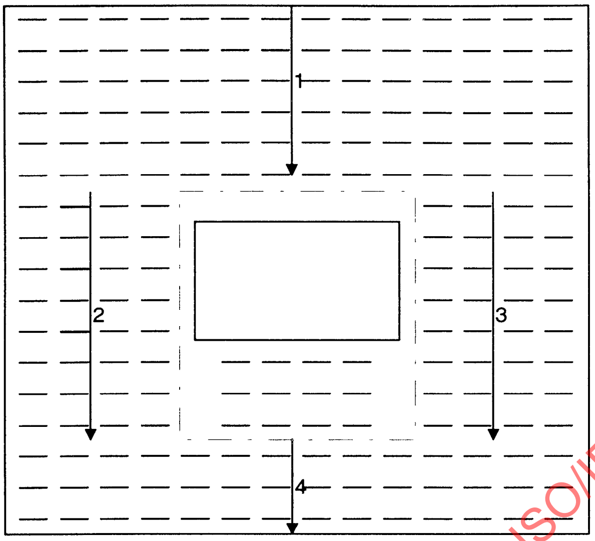
2) The picture is centred and has all its associated text on the left side, and the main text runs down on the right side.



T0816250-94/d48

The only modifications are that frame F13 shall have the permitted category including the category for the main text, which will be as specified by F0, its vertical position to be = 0.

3) The main text runs down on both sides of the figure, and the figure heading is below it. In addition, the heading has a border.



T0816250-94/d49

The reading order of the main text is 1, 2, 3, 4, as indicated.

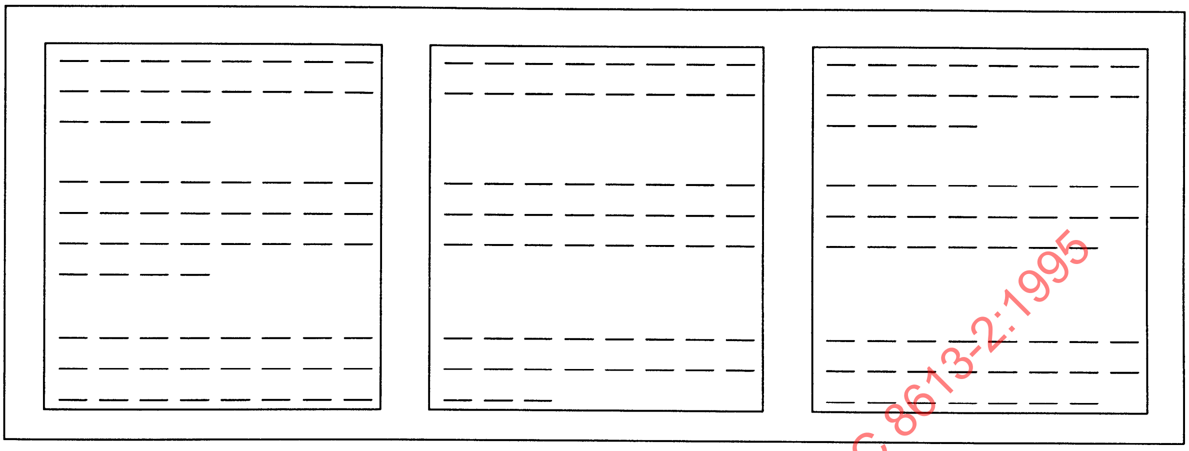
The figure and its heading are represented by a composite logical object with the picture and the figure text as the only two subordinate logical objects, the picture is the first of these.

The composite logical object specifies the attribute "layout object class" F12. There is an additional block in frame F12, not illustrated in Figure C.4, used to hold the heading of the figure. Frames F11 and F13 both have the same permitted category as required for the main text and their vertical position = 0.

C.1.6 Multi-lingual synchronized text

Informal description of feature illustrated:

Three columns of text on a page, each column containing text in a different language. There is required to be horizontal synchronization at various points, for example, paragraphs, clauses.



T0816260-94/d50

Assumptions:

Positions and widths of columns are fixed values.

The sets of information to be synchronized are grouped together within the logical structure by a composite logical object, and this object specifies new layout object F1, in order to create a new F1 and thus align the start of the related information.

The related material can cross page boundaries.

Generator for subordinates:

P	OPT REP F1
F1	SEQ (F11 F12 F13)
F11, F12, F13	None.

Variation:

If the text of one of the columns is in Japanese, say the third column.

Then the layout path in frame F13 would change to 180° and a number of subordinate frames would be specified, one for each line of Japanese text to be placed in this column. These frames would be balanced, within each of these subordinate frames the layout path is 270°. Therefore the Japanese text would take up as little space in the vertical dimension as possible.

For an illustration, see Figure C.5.

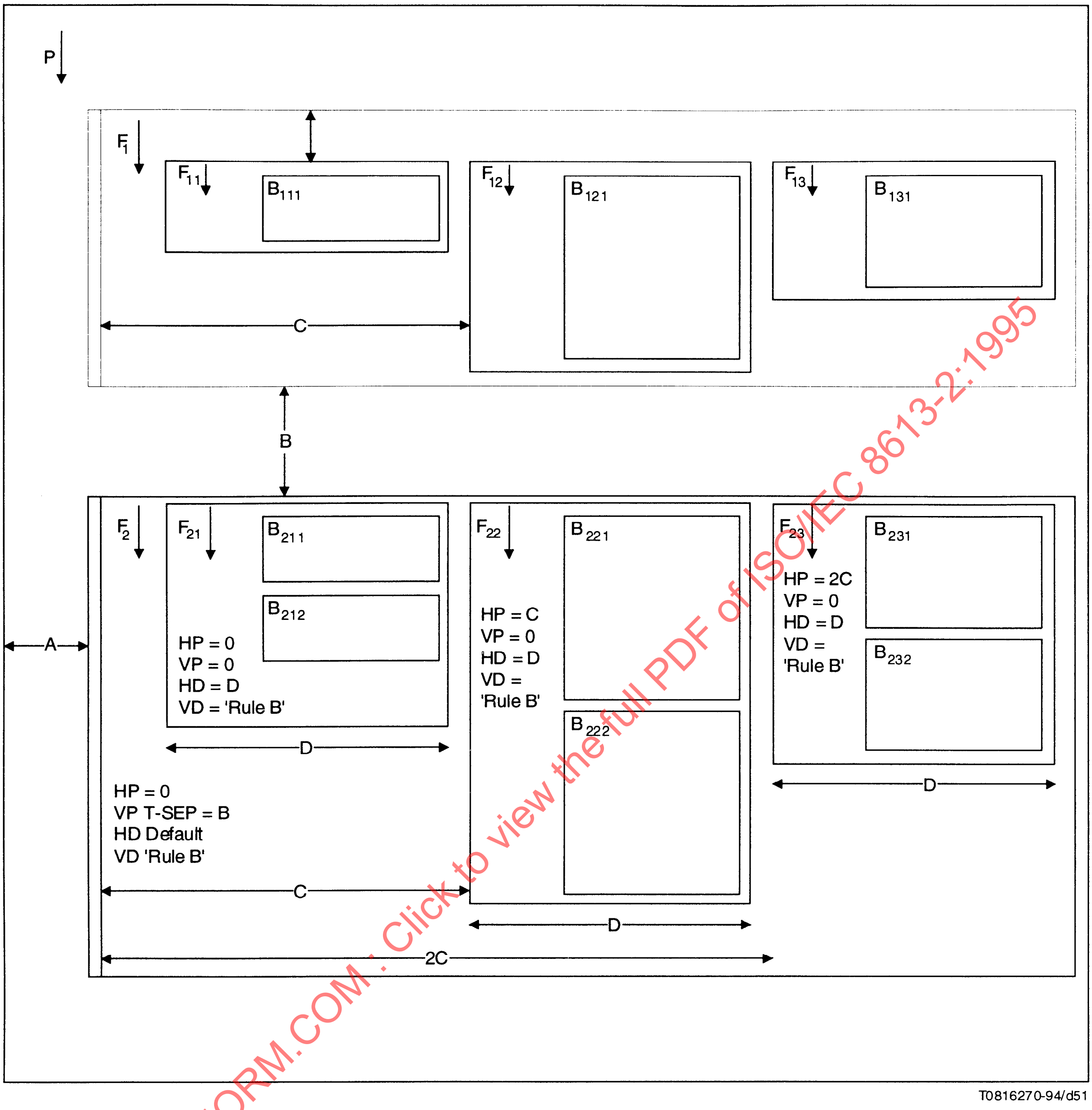


Figure C.5 – Multi-lingual synchronised text

C.1.7 Footnote placement

Informal description of feature illustrated:

Footnote placed at the bottom of the page, full width.

The footnote might be referred to from a multi-column layout of the main text.

If the footnote were referred to from the last line of the main text, then a new page may be forced for both the footnote and the footnote reference.

The footnote may be continued on the next page unless it is specified as being 'indivisible'.

Generator for subordinates:

P	SEQ (CHO (F0 F1))
F1	None

For an illustration, see Figure C.6.

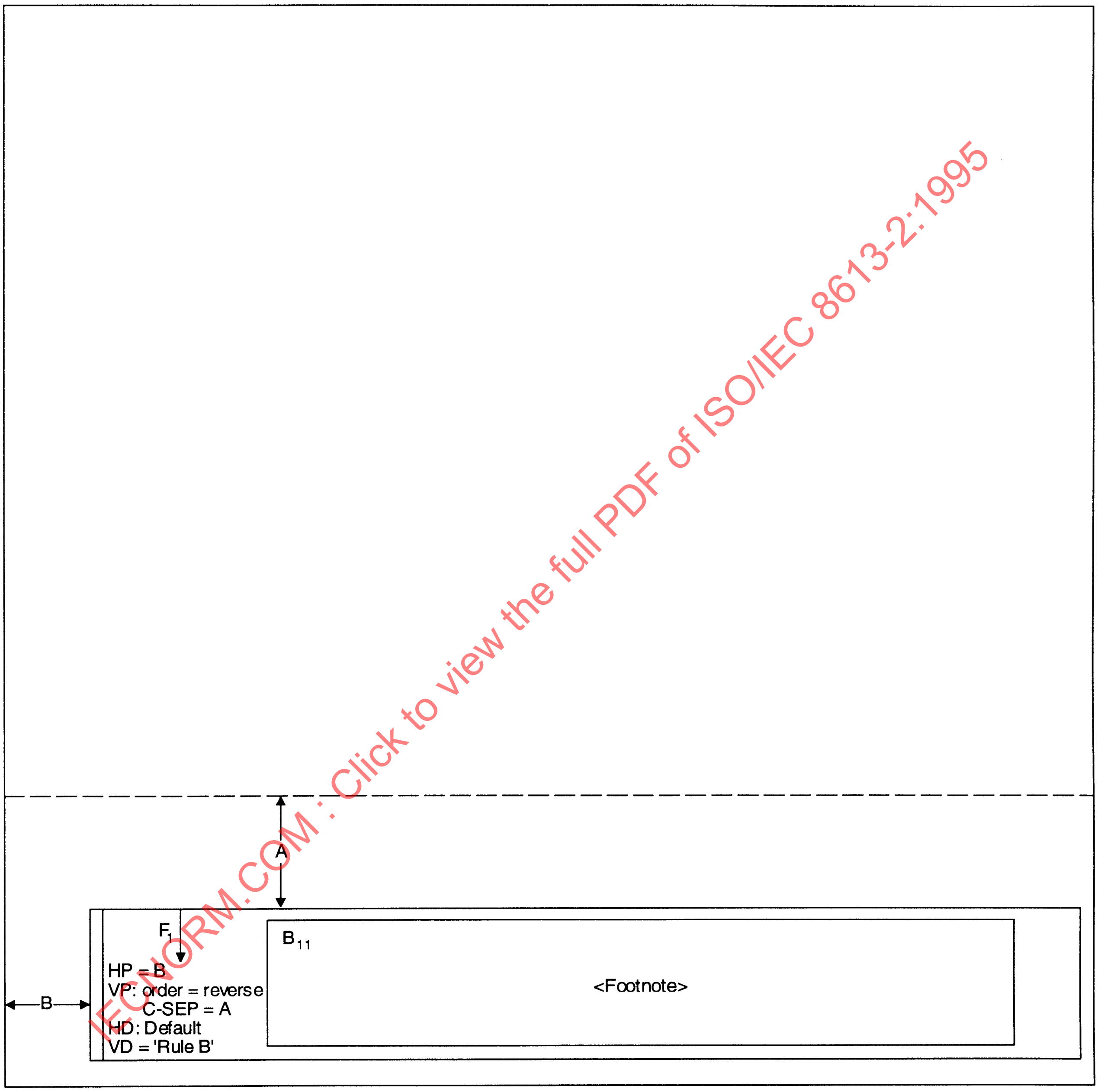


Figure C.6 – Footnote placement

C.1.8 Tabular layout

Informal description of feature illustrated:

Tabular layout, each row having just sufficient lines to contain the content. Table elements may be of any content architecture.

Assumptions:

The column positions and dimensions are fixed by the layout object class descriptions for the table. The generic layout structure does not provide any assistance for column widths and positions to vary. This is indicated by the notation

HP = *, HD = *.

However, the number of lines required for each row is variable, depending on the content of that row.

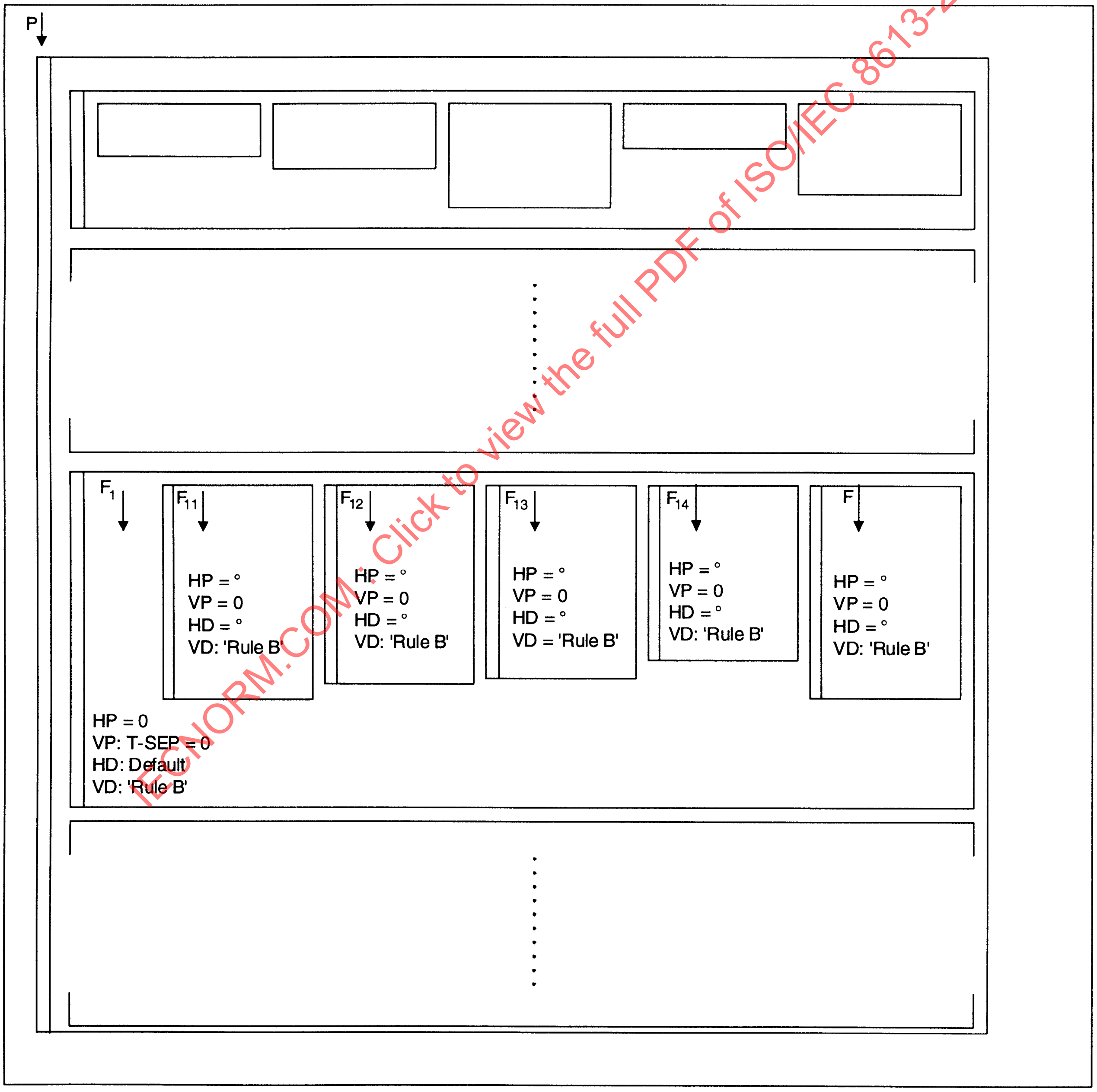
Generator for subordinates

P SEQ (CHO (... F 1 ...))

F1 SEQ (F11 ... F 1*i* ...)

F1i SEQ (F1*il* ... F 1*ij* ...)

For an illustration, see Figure C.7.



T0816290-94/d53

Figure C.7 – Tabular layout

C.2 Layout of blocks

This subclause contains illustrations of the use of the attributes:

- concatenation;
- offset;
- separation;
- layout path;
- fill order;
- block alignment.

C.2.1 Concatenation

Concatenation can be used to lay out an automatically generated chapter number with a chapter heading, see Figure C.8. Concatenation can also be used to lay out two or more parts of a paragraph where the parts have been split for reasons of logical structuring, for example because of a footnote reference, see Figure C.9.

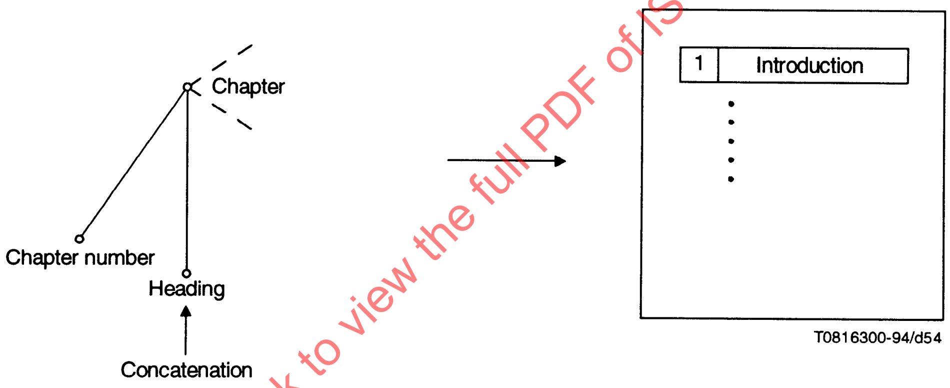


Figure C.8 – Concatenation of chapter number and title

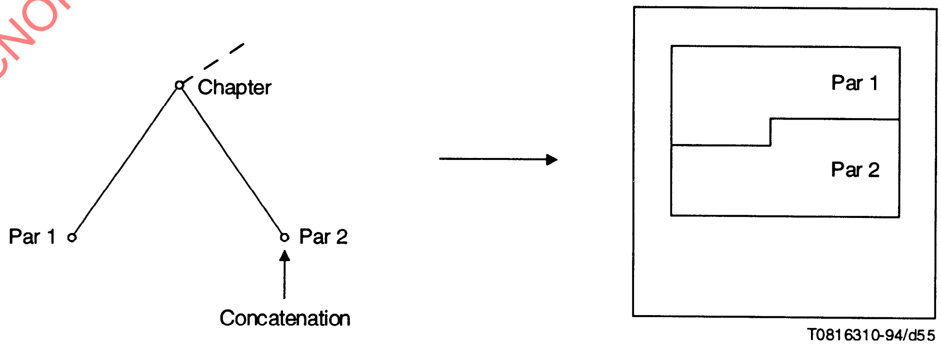


Figure C.9 – Concatenation of parts of a paragraph

C.2.2 Offset

Offset can be used to place a figure or some text at a minimum specified distance from the edges of the frame into which it is laid out.

In Figure C.10, the attribute "layout path" is assumed to have its default value of 270° and the attribute "fill order" is assumed to have the value 'reverse order'. If the attribute "fill order" had the value 'normal order' then the figure or text would have been placed towards the top of the page.

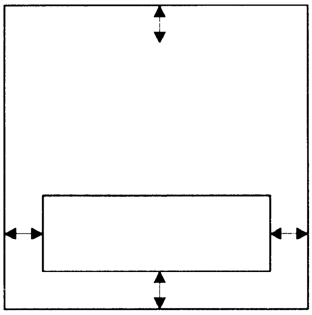


Figure C.10 – Illustration of attribute "offset"

C.2.3 Separation

Separation can be used to place a figure or some text into separate blocks with a minimum specified distance of separation, see Figure C.11.

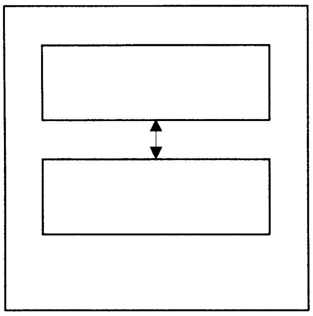


Figure C.11 – Illustration of attribute "separation"

C.2.4 Layout path

Layout path can be used to control the direction of placement of figures or text into blocks in a frame as they occur in logical order, for example top to bottom or left to right, see Figure C.12.

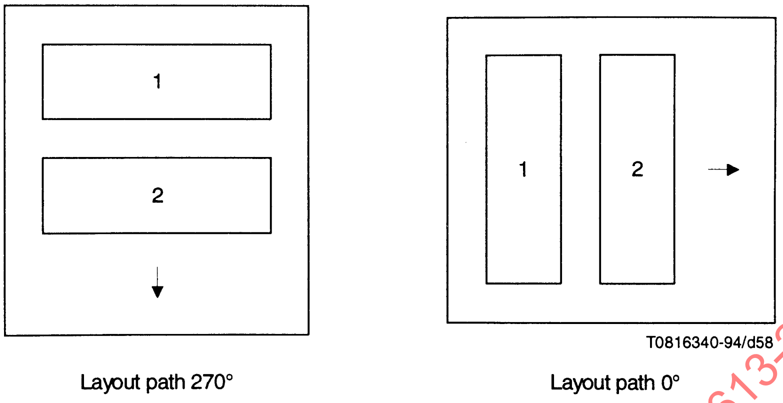


Figure C.12 – Layout path 0° and 270°

C.2.5 Fill order

Fill order can be used to lay out text or figures from the top and towards the bottom of a frame; alternatively, from the left, right or bottom of a frame and towards the right, left or top of the frame, respectively. The direction used is determined by the layout path, the order of layout is the sequential logical order.

Paragraphs of text are often laid out with 'normal order' fill order, footnotes are often laid out with 'reverse order' fill order. When a logical sequence of paragraphs and footnotes associated with 'normal order' and 'reverse order' respectively is laid out, with the default value of layout path 270°, then the paragraphs will be laid out from the top of the page, in their logical sequence in the layout path direction, followed by the footnotes grouped at the bottom of the page, also in their logical sequence in the layout path direction.

Figure C.13 illustrates an example of laying out two paragraphs and two footnotes that occur in the following sequential logical order:

- paragraph 1;
- footnote 1;
- footnote 2;
- paragraph 2.

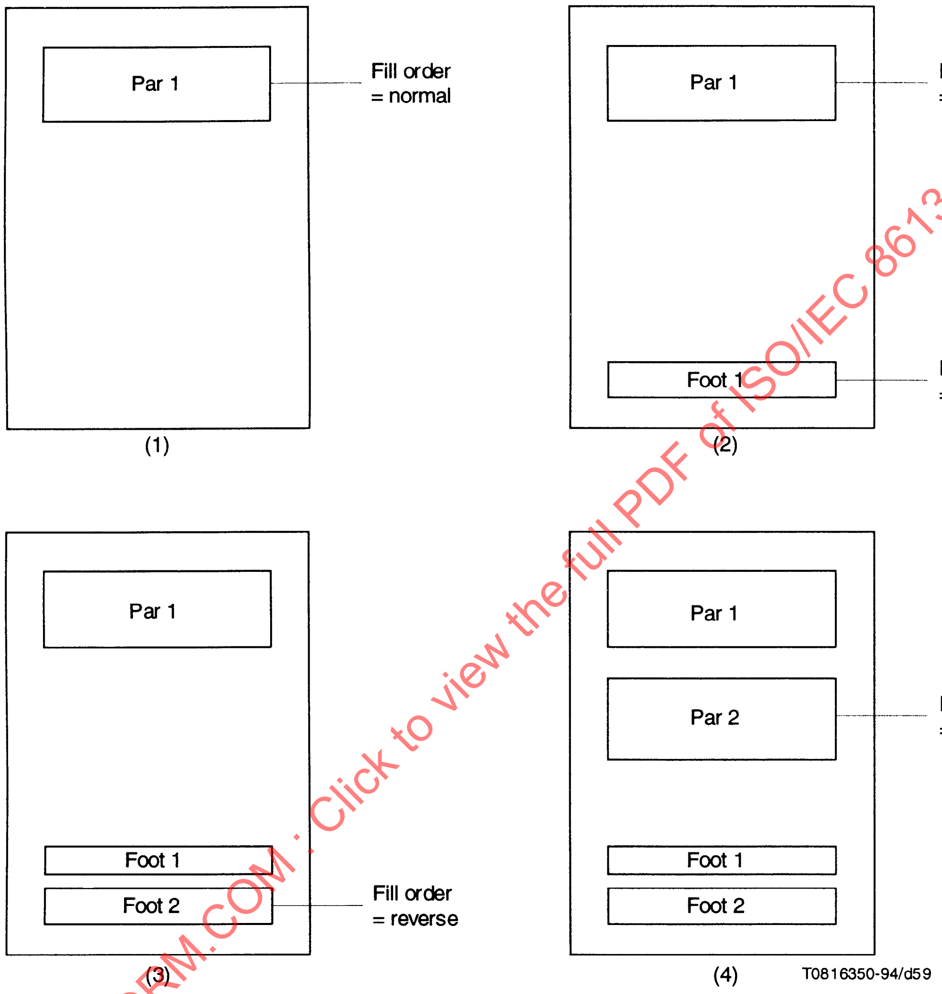


Figure C.13 – Illustration of fill order

C.2.6 Block alignment

Block alignment can be used to lay out a block of text or a figure 'left-hand aligned', 'right-hand aligned' or 'centred' in the direction orthogonal to that of the layout path. This could be used, for example, to centre the title of a chapter on the page.

Figure C.14 illustrates its use for aligning figures to the left or right. The figure illustrates that, with the default value of the attribute “layout path” 270°:

- 'left-hand aligned' results in alignment to the right of the immediately superior layout object;
- 'right-hand aligned' results in alignment to the left of the immediately superior layout object.

Figure C.14 illustrates also that the alignment is constrained by the specification of the attribute “offset”.

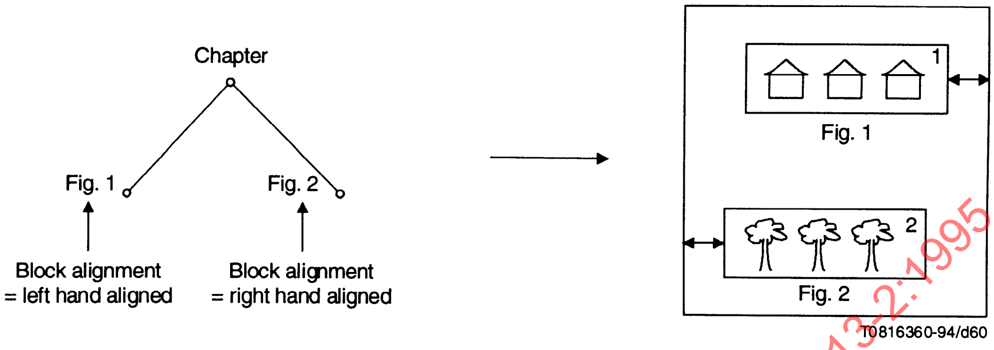


Figure C.14 Illustration of block alignment

C.2.7 Illustration of attributes used in combination

This subclause illustrates the combined use of the attributes “offset”, “separation” and “fill order”.

In this subclause, the following notation is used:

- OT, OLD, OR, OL:

  Values of the parameters “trailing offset”, “leading offset”, “right-hand offset” and “left-hand offset”, respectively, of the attribute “offset”, for the block indicated by the subscript.

- SL, ST:

  Values of the parameters “leading edge”, “trailing edge”, respectively, of the attribute “separation”, for the block indicated by the subscript.

- C:

  Values of the parameter “centre separation”, of the attribute “separation”, for the block indicated by the subscript.

Figure C.15 illustrates the combined use of the attributes "offset" and "separation" for two blocks that contain logical objects whose attribute "fill order" has the value 'normal order'.

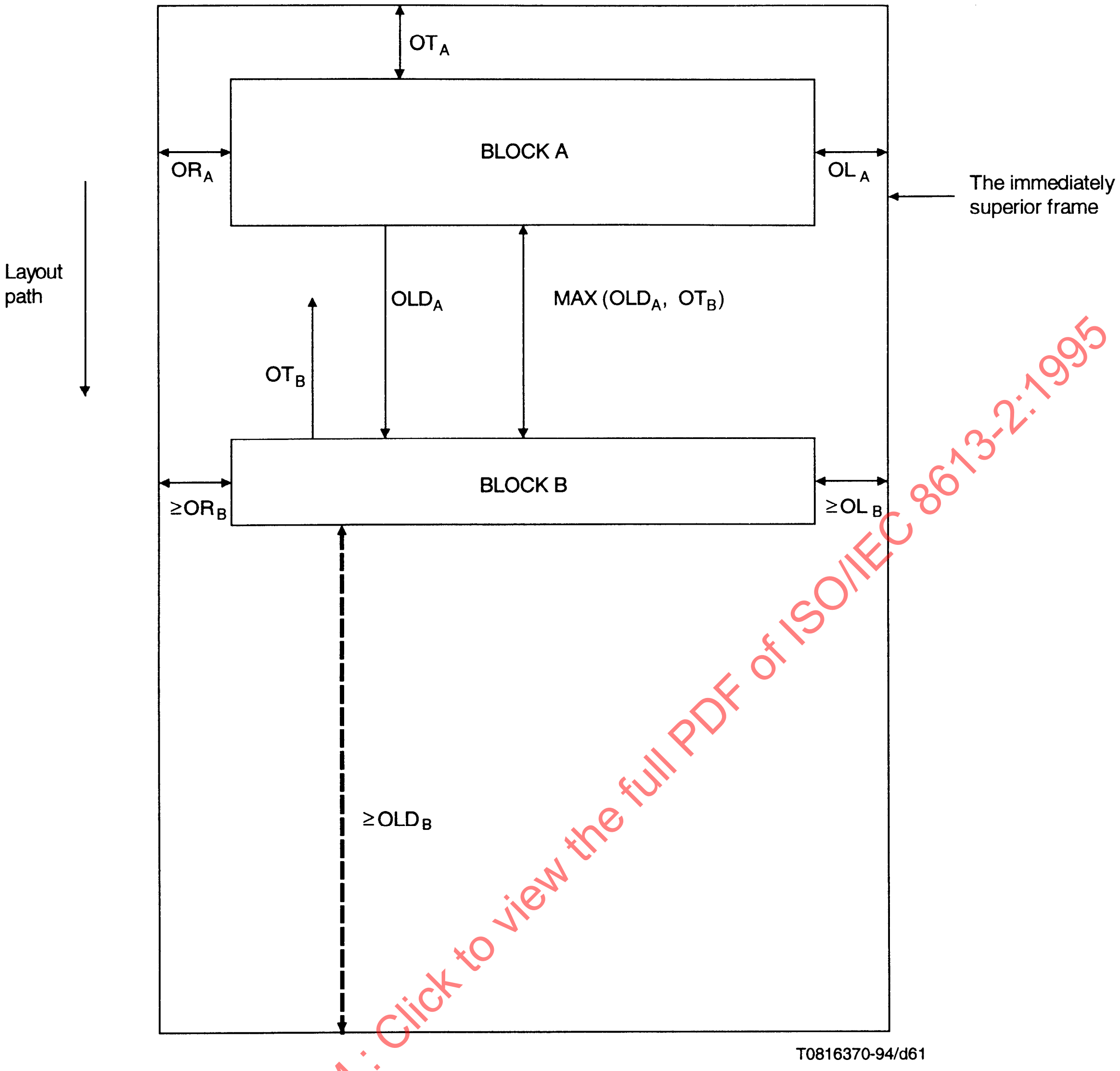
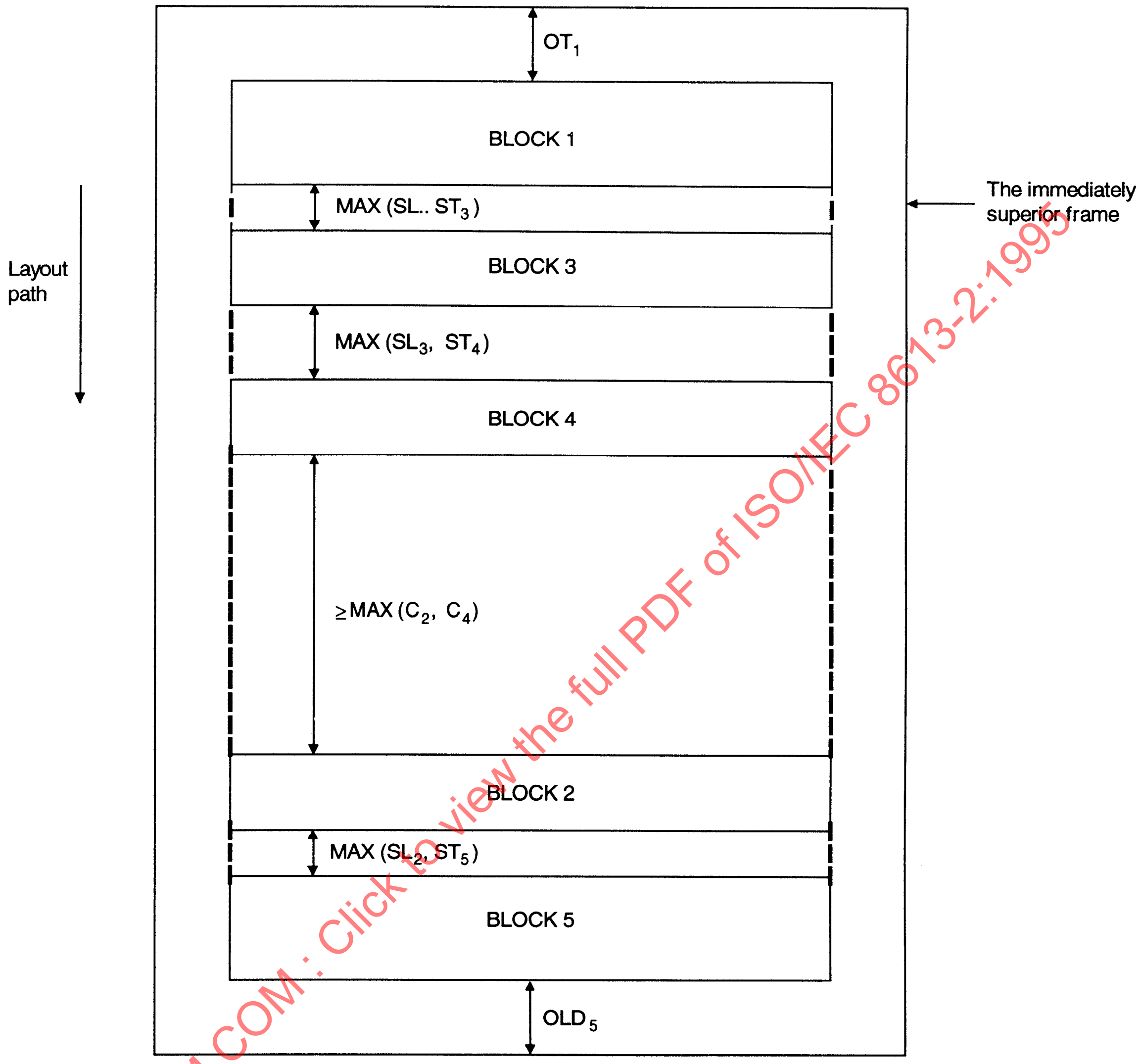


Figure C.15 – Illustration of the attribute "separation"

Figure C.16 illustrates the combined use of the attributes “offset”, “separation” and “fill order” and shows blocks that contain the content of logical objects that specify both ‘normal order’ and ‘reverse order’ values of the attribute “fill order”.

Figure C.16 is an example of the layout that would result when blocks are laid out in the order 1, 2, 3, 4, 5 where 1, 3, 4 specify normal fill order and 2, 5 specify reverse fill order.



T0816380-94/d62

Figure C.16 – Illustration of layout of blocks in normal and reverse order

Figure C.17 illustrates the combined use of the attributes "offset", "separation" and "fill order" and shows blocks that contain the content of logical objects that specify both 'normal order' and 'reverse order' values of the attribute "fill order".

Figure C.17 illustrates the determination of the available area during the part of the layout process concerned with positioning of blocks.

In Figure C.17, block A has been laid out in 'normal order', whereas block B has been laid out in 'reverse order'. When block C has been laid out, block C will be placed as close to block A as is possible within the indicated available area.

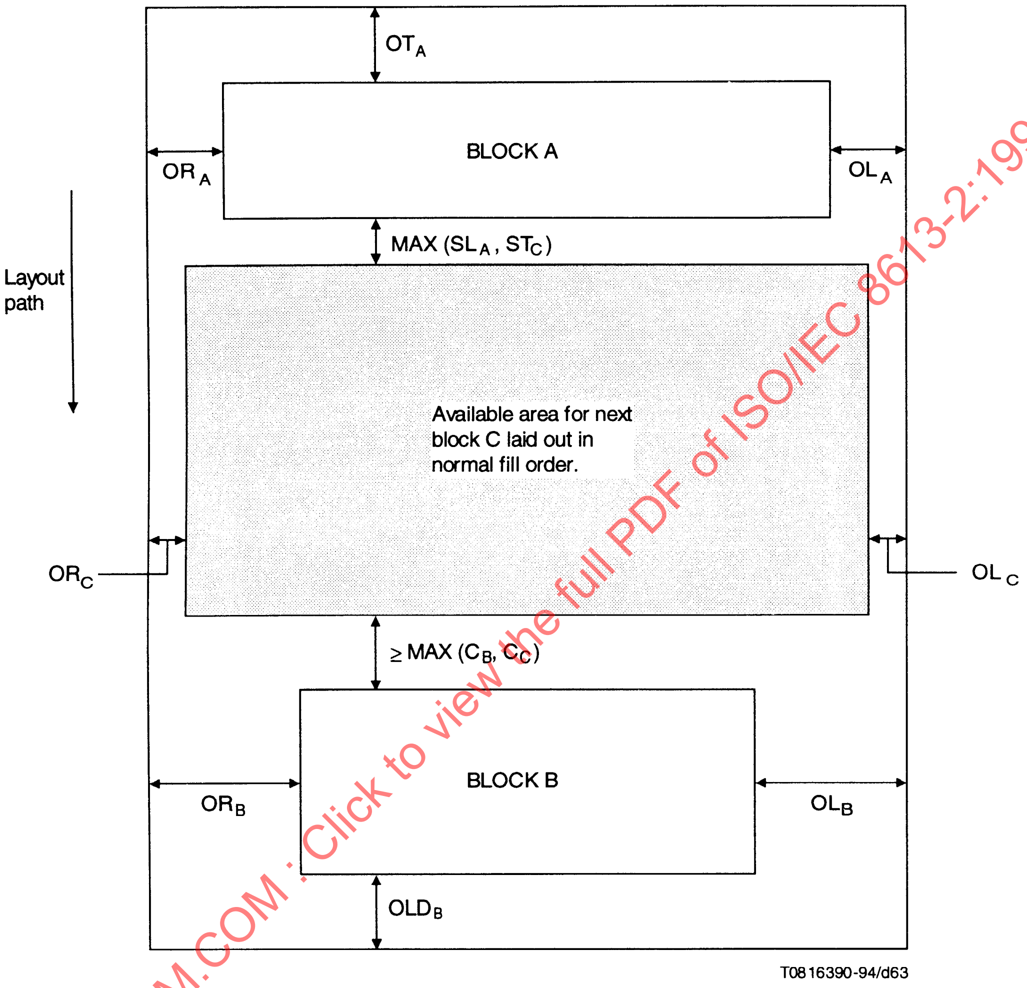


Figure C.17 – Illustration of layout of logical objects in normal order

Figure C.18 illustrates the determination of the available area during the part of the layout process concerned with positioning of blocks.

In Figure C.18, block A has been laid out in 'normal order', whereas block B has been laid out in 'reverse order'. In order to determine the available area for block C, block B has temporarily been moved as far as possible in the direction opposite to the layout path. When block C has been laid out, both blocks B and C will be moved as far as possible in the direction of the layout path, ensuring that the values of the attributes "offset" and "separation" for blocks A, B, C are all complied with, in particular, the value of the attribute "offset" for block C.

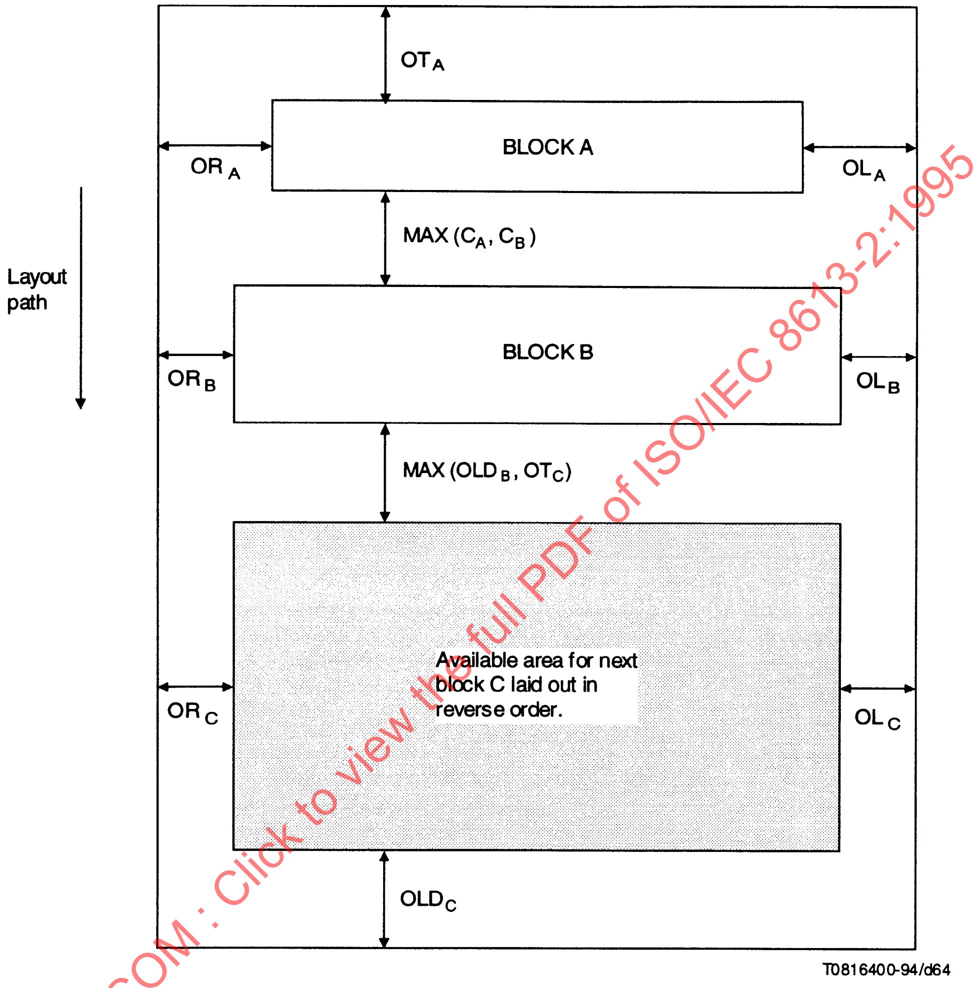


Figure C.18 – Illustration of layout of logical objects in reverse order

C.3 Further constraints on layout

This clause contains illustrations of the use of the attributes:

- layout object class;
- synchronization;
- new layout object;
- same layout object;
- indivisibility;
- balance.

C.3.1 Layout object class

The attribute "layout object class" may also be used to define the appropriate layout areas for particular logical groups of content. The difference from layout category is that the identified logical object is placed in its entirety into a single instance of the layout object. For example, in C.1.4 and Figure C.3, the picture is placed into F11 by specifying the layout object class, this ensures that each picture that occurs causes a new instance of F11 and therefore of F1. Also, in C.1.5 and Figure C.4, the picture and the associated pieces of text are placed into frames F11, F12 and F13 by specifying the attribute "layout object class".

C.3.2 Synchronization

The attribute "synchronization" may be used to lay out margin notes beside the associated text. This is illustrated in Figure C.19.

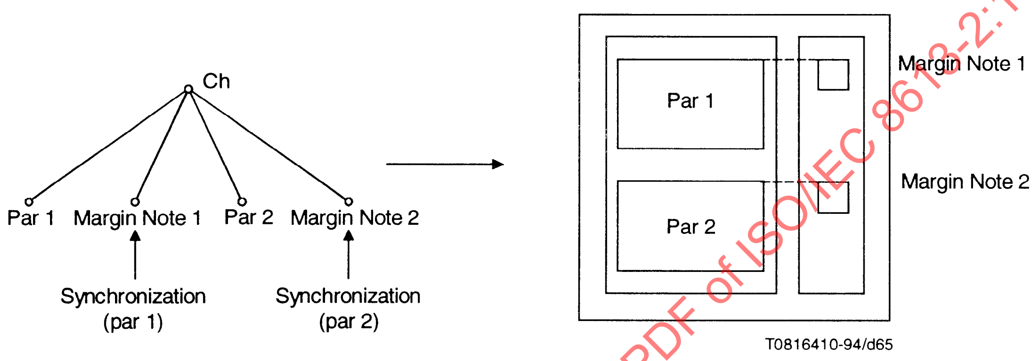


Figure C.19 – Illustration of synchronization

C.3.3 New layout object

The attribute "new layout object" may be used to constrain a chapter to start on a new page, see Figure C.20.

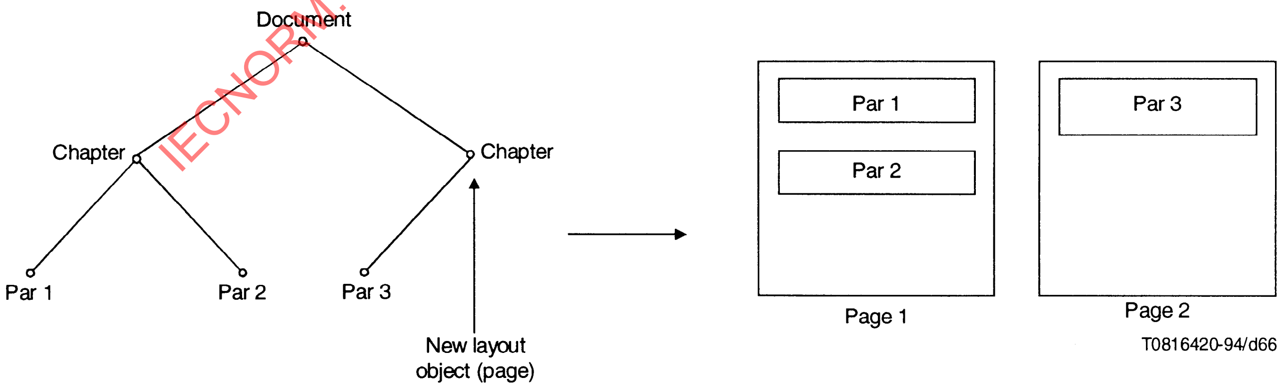


Figure C.20 – Illustration of new layout object

C.3.4 Same layout object

The attribute "same layout object" may be used to constrain a chapter heading and the beginning of the first paragraph of the chapter to be laid out together, for example, to avoid a page break directly after the chapter heading, see Figure C.21.

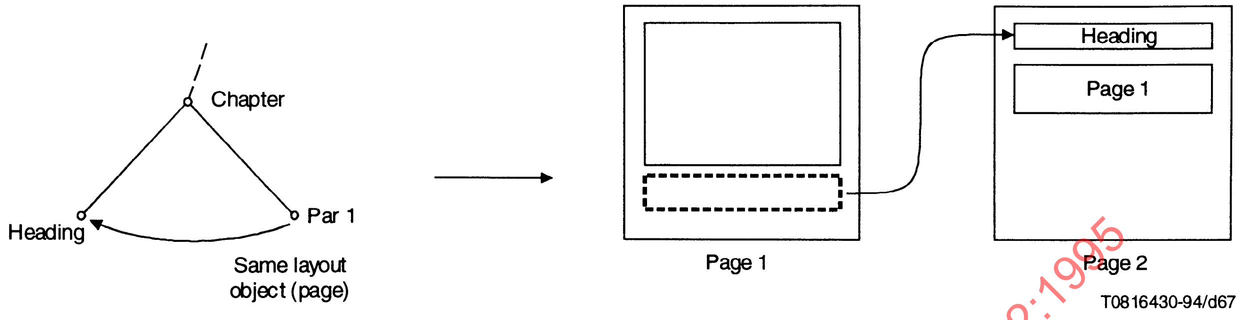


Figure C.21 – Illustration of same layout object

C.3.5 Indivisibility

The attribute "indivisibility" may be used to constrain parts of the content such that they are not split during the layout process, for example a figure or a table, see Figure C.22.

In conjunction with the attribute "same layout object", it could be used to specify that a footnote is to be placed entirely on the same page as the footnote reference. The use of "same layout object" by itself would specify only that the text of the footnote is to start on the same page.

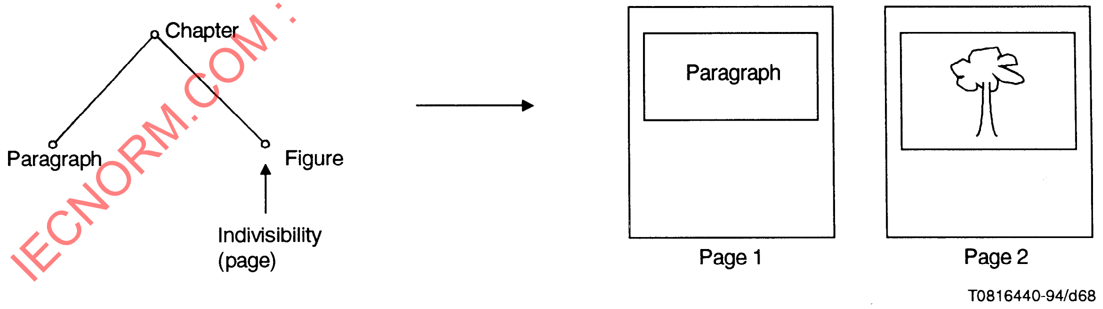


Figure C.22 – Illustration of indivisibility

C.3.6 Balance

The attribute "balance" may be used to constrain the last page of a chapter that is laid out in 2 column layout to be such that the two columns of text of the chapter on the last page are approximately equal in height, see Figure C.23.

A further example is included in C.1.6, variation 1.

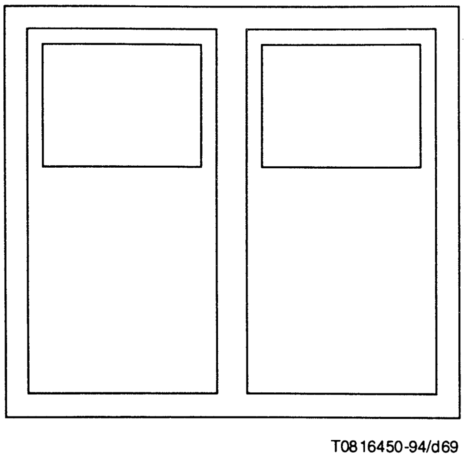


Figure C.23 – Illustration of balance

C.4 Texture and imaging order attributes

The attributes "transparency" and "colour" together define the texture of pages, frames and blocks. Together with the attribute "imaging order", they describe how to combine the images of the content of overlapping layout objects.

C.4.1 Example 4.1

Figure C.24 illustrates three overlapping blocks 1, 2 and 3, with the imaging order 1, 2, 3, thus:

- block 2 overlays block 1;
- block 3 overlays block 1 and block 2.

The texture of the blocks is as follows:

- block 1 has the texture: colour of media, opaque;
- block 2 has the texture: colour of media, transparent;
- block 3 has the texture: colour of media, opaque.

The content of block 3 is all visible, with the background colour that of the media.

The content of block 2 is:

- not visible in the area of intersection with block 3;
- combined with the content of block 1 in their remaining area of intersection;
- visible, with background colour that of the media, in the area in which it does not intersect block 1 or block 3.

The content of block 1 is:

- not visible in the area of intersection with block 3;
- combined with the content of block 2 in their remaining area of intersection;
- visible, with background colour that of the media in the area in which it does not intersect block 2 or block 3.

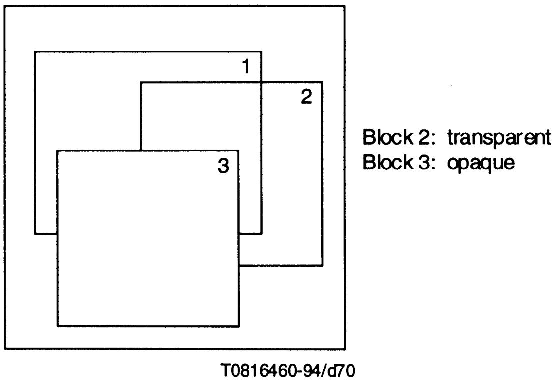


Figure C.24 – Illustration of overlapping blocks

C.4.2 Example 4.2

Figure C.25 shows an example of an overlay order consisting of a page P with two frames F1, F2, and 3 blocks, B1, B2 and B3. The attribute "imaging order" is not specified and thus the imaging order is that of the sequential layout order, which is as represented by the tree structure of Figure C.25. The final image is shown in Figure C.26. In this example, all blocks have the attribute "layout texture" with the value 'colour of media', 'opaque'.

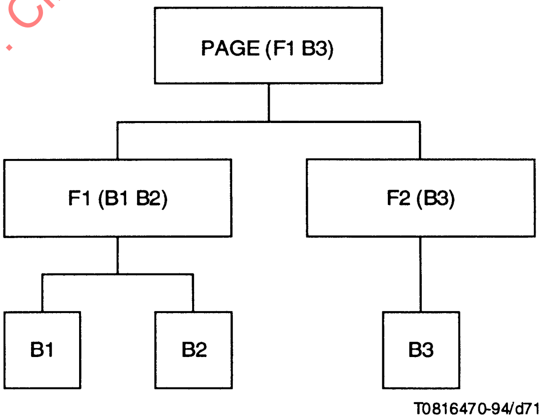


Figure C.25 – Sequential layout order

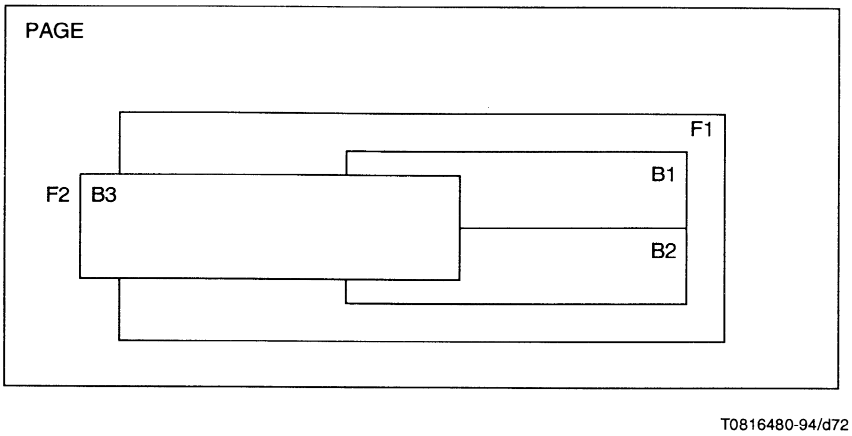


Figure C.26 – Final image

C.4.3 Example 4.3

In this example, multiple levels of frames are considered. The page is defined by the tree structure shown in Figure C.27.

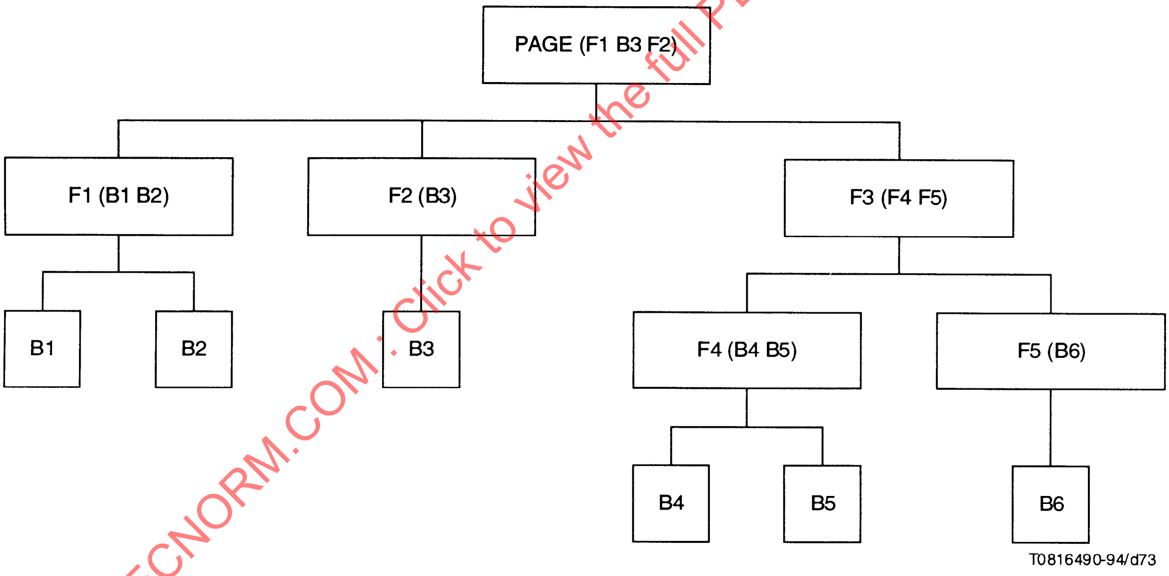


Figure C.27 – Sequential layout order

The imaging order either specified explicitly or defined by the sequential layout order is:

- page P: imaging order = Fl, F2, F3;
- frame F1: imaging order = B1, B2;
- frame F3: imaging order = F4, F5;
- frame F4: imaging order = B4, B5.

The values of the attribute "layout texture" are as follows:

- blocks B1, B4 and B6: colour of media, transparent;
- blocks B2, B3 and B5: colour of media, opaque.

To image the proposed page we consider the imaging order given at the page level which makes the frame F1 and its subordinates objects the first to be imaged. Within this frame, B1 is the first block imaged, according to the imaging order below the frame level.

The next block to be imaged is B2, the second and last block that is subordinate to the frame F1. As this block is opaque, the content of B1 is suppressed within the area of intersection.

This is illustrated in Figure C.28. To simplify this and subsequent figures, frames are not shown in all cases.

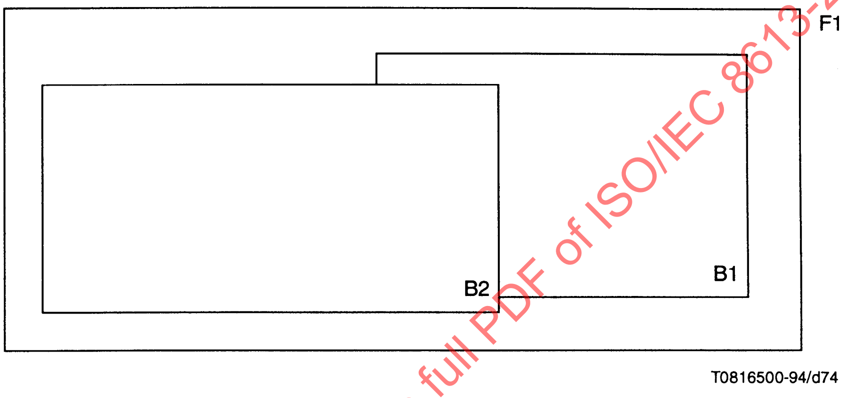


Figure C.28 – Image within frame F1

The imaging of the blocks subordinate to F1 is now complete. The next step of the imaging process considers the frame F2, which contains just one block, B3, as specified by the imaging order at the page level. As the value of the attribute "layout texture" is 'colour of media', 'opaque', the content of block B2 is suppressed within the area of intersection with block B3 (see Figure C.29).

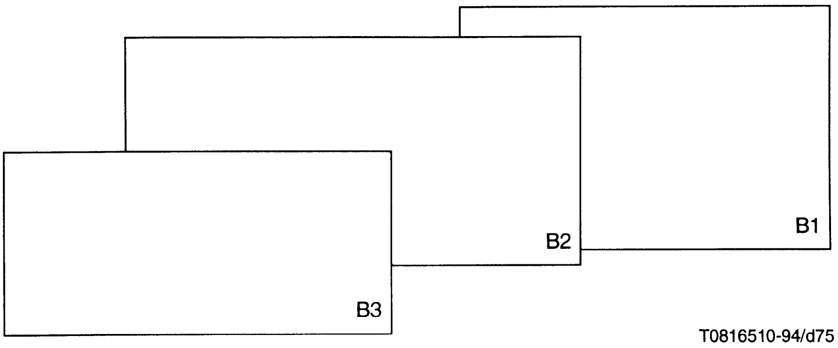


Figure C.29 – Imaging of blocks B1, B2, B3

The last branch of the layout structure, the frame F3 is now to be imaged. The imaging order specified by this frame indicates that the frame F4 and its subordinate objects are the first to be imaged. Block B4, which has "layout texture" 'colourless, transparent', has its content added to the current composite image as illustrated in Figure C.30. This block does not intersect with other blocks that have previously been imaged.

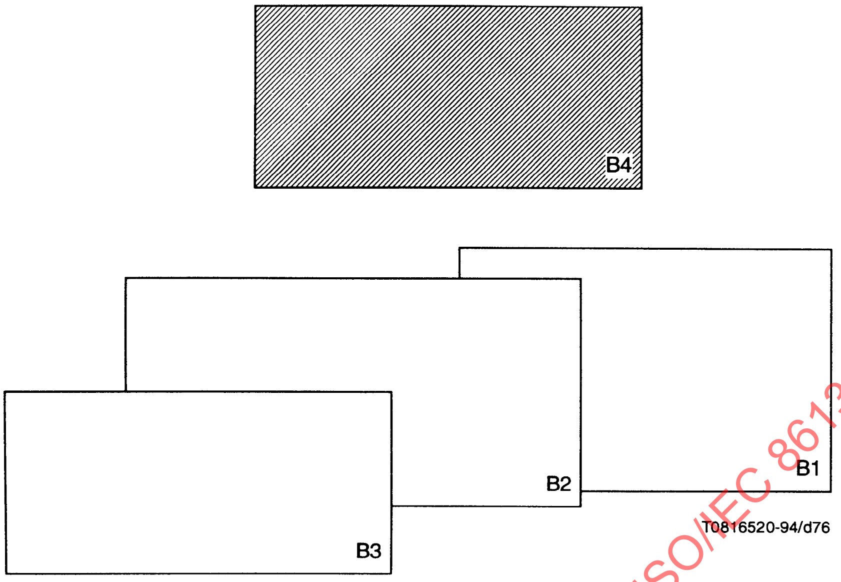


Figure C.30 – Image of block B4 added

The block B5 in frame F4 is now imaged. This block has "layout texture" 'colourless, opaque', and the content of B2 and B4 are suppressed in their area of intersection with B5 (see Figure C.31).

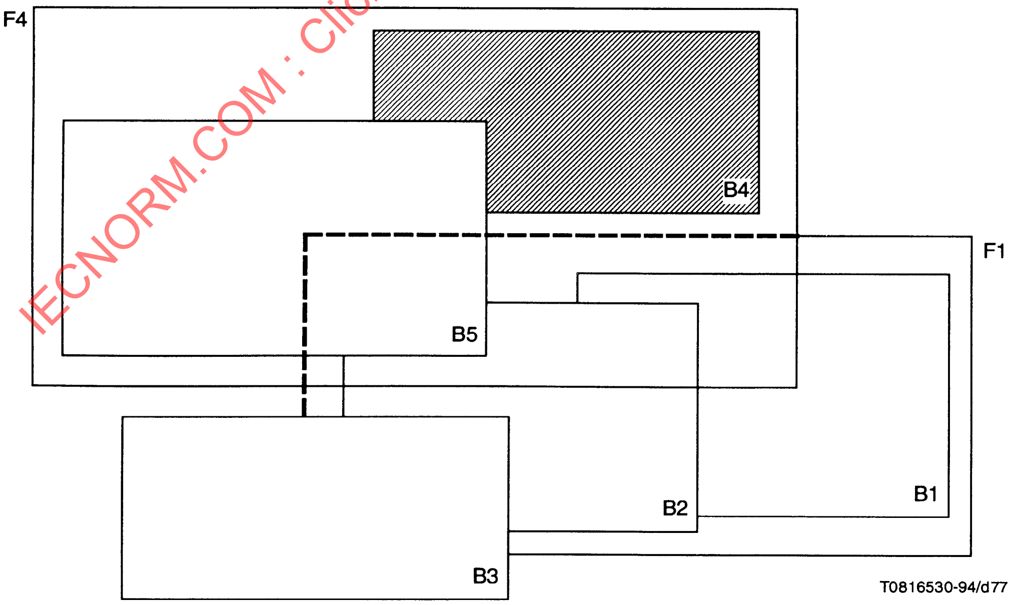


Figure C.31 – Image of block B5 added

The final block of the page, which has to be the last imaged as defined by the imaging order, is B6. As this block has "layout texture" 'colourless, transparent', its content is combined with the intersecting blocks B1, B2, B3, B4 and B5. This completes the imaging process and the finished page is shown in Figure C.32.

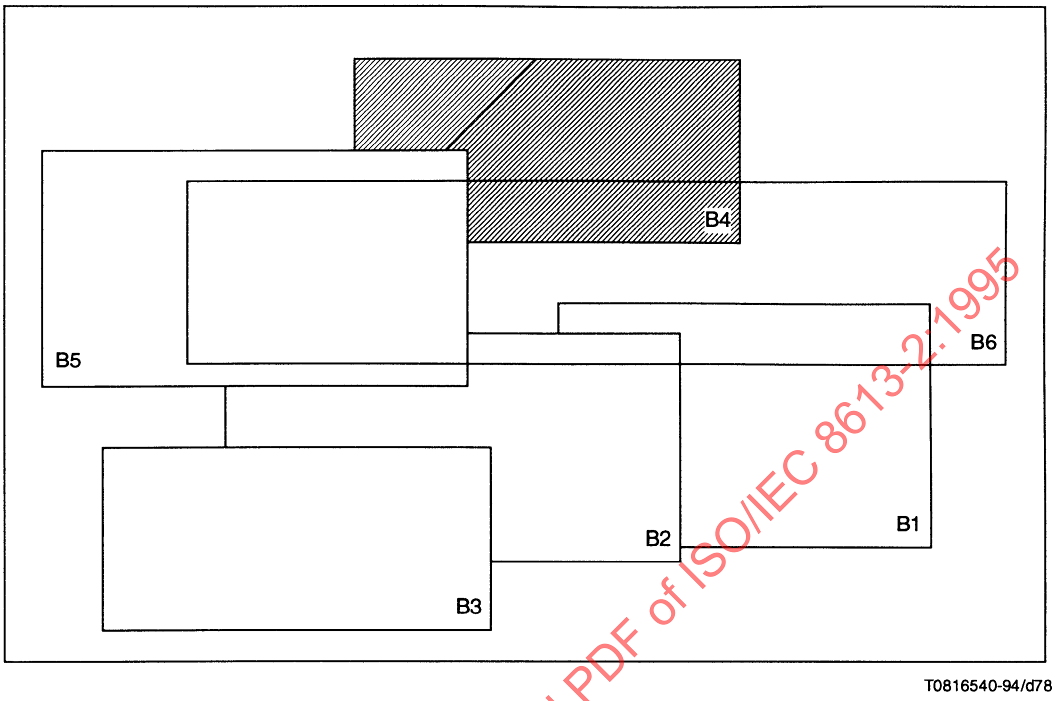


Figure C.32 – Final image

C.4.4 Example 4.4, example of an application

This feature could be applied, for example, to a map with an overlaid transparent grid. Both the map and the grid are overlaid with an opaque key at a fixed position. The map could have a generic description in which the key and the grid could be generic content portions while the particular map would be specific content.

C.5 Bindings, content generator

The attribute "bindings" together with the attribute "content generator" may be used to create automatically numbering schemes for layout and logical objects, for example, pages, sections, chapters, footnotes, or other application specific logical structures, for example, "part numbers".

Figure C.33 illustrates the automatic creation of chapter numbers.

In this illustration, each chapter has a subordinate logical object "Nr" which represents the chapter number.

Each of these logical objects specifies an attribute "bindings" which specifies a binding name, binding value pair of parameters. The binding name is also "Nr" (by coincidence). The binding value is defined by an expression which defines the value in terms of the previous chapter number, incremented by one.

All expressions in this example are specified using the notation specified in Annex A.

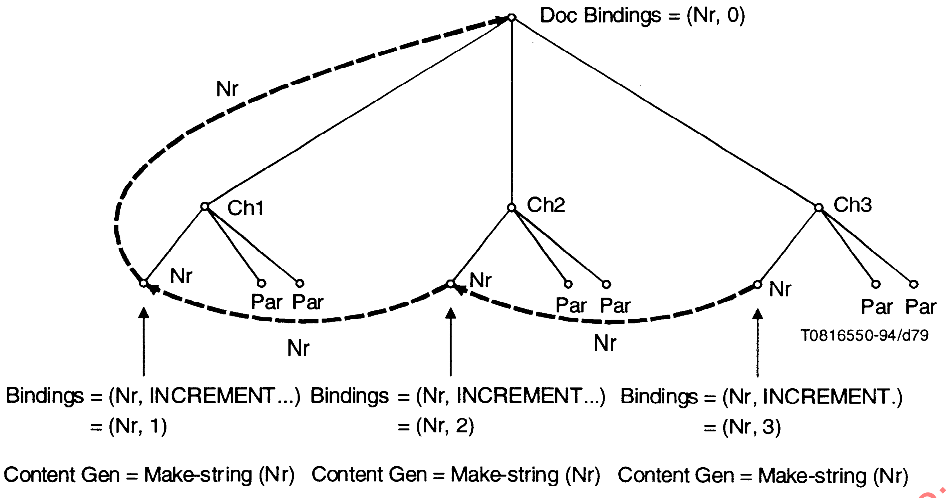

Doc Bindings = (Nr, 0)
Nr
Ch1
Ch2
Ch3
Nr
Par Par
Nr
Par Par
Nr
Par Par
Nr
Nr
T0816550-94/d79
Bindings = (Nr, INCREMENT...)
= (Nr, 1)
Bindings = (Nr, INCREMENT...)
= (Nr, 2)
Bindings = (Nr, INCREMENT.)
= (Nr, 3)
Content Gen = Make-string (Nr)
Content Gen = Make-string (Nr)
Content Gen = Make-string (Nr)

Figure C.33 – Use of bindings and content generator for numbering chapters

The expression specified by the binding value is:

```
INCREMENT
    (BINDING_REFERENCE
        (PRECEDING (CURRENT OBJECT))
        (Nr))
```

The evaluation of the binding-reference will search backwards in sequential logical order through the specific logical structure until a binding with the binding name “Nr” is located. The value of this binding will then be taken and incremented by one to form the value of the current binding.

In order to generate the chapter number, the basic logical object “Nr” also specifies a content generator. This content generator will use the string expression:

```
MAKE_STRING
    (BINDING_REFERENCE
        (CURRENT_OBJECT)
        (Nr))
```

This expression returns a string representation of the chapter number, which will be incorporated as the “content information” in a content portion of the specific layout structure for the document.

The chapter number binding values may be initialised by specifying a binding value at the document root level, to set the initial value of the bindings with binding name “Nr” to integer value zero.

This is set to zero so that the first chapter number will be one. The binding name, binding value pair specified at the document logical root is (Nr, 0), where 0 is a numeric literal and “Nr” is the binding name.

Layout and presentation characteristics of the chapter number may be specified by referencing layout or presentation styles from the basic logical object.

Annex D

The defaulting mechanism

(This annex does not form an integral part of this Recommendation | International Standard)

This annex summarises some aspects of the defaulting mechanism.

D.1 The defaulting mechanism as applicable to each defaultable attribute

Table D.1 summarises the steps of the defaulting mechanism described in 9.1.2.4 as these apply to the various attributes.

D.2 Determination of content portions and their associated attributes

D.2.1 Determination of content associated with basic logical objects

During the layout process the content associated with a basic logical object is determined according to the first of the following rules which is applicable:

1) One or more content portions are specified for the basic logical object and in at least one of them the attribute "content information " is specified.

In this case, the content associated with the basic logical object is formed by the concatenation of the strings specified for the attribute "content information" for each content portion, in the sequential order of the content portions.

The content portion attributes are derived from the corresponding content portions.

2) One or more content portions are specified for the basic logical object, and none of the content portions specifies the attribute "content information". The attribute "content generator" is specified for the basic logical object.

In this case, the content associated with the basic logical object is formed by evaluation of the attribute "content generator".

The content portion attributes are derived from the first content portion in sequential order.

3) One or more content portions are specified for the basic logical object, and none of the content portions specifies the attribute "content information". The attribute "content generator" is not specified for the basic logical object.

In this case, the content associated with the basic logical object is a "null" string.

The content portion attributes do not apply.

4) No content portions are specified for the basic logical object. The attribute "content generator" is specified for the basic logical object.

In this case, the content associated with the basic logical object is formed by evaluation of the attribute "content generator".

The content portion attributes are derived using the rules for determining values of attributes of content portions (see 9.1.2.5).

5-8) If the basic logical object description refers to an object class description which either specifies content portions or specifies the attribute "content generator", then apply rules 1)-4), using the content portions and/or attribute "content generator" specified by the object class description.

9-12) If the basic logical object description refers to an object class description which refers to an object class description in the resource-document which either specifies content portions or specifies the attribute "content generator", then apply rules 1)-4), using the content portions and/or attribute "content generator" specified by the object class description in the resource-document.

13) If no content is determined by rules 1)-12), then the content associated with the basic logical object is a "null" string.

The content portion attributes do not apply.

Table D.1 – Applicable steps for defaulting of attributes

Attribute Name	Steps from 9.1.2.4								
object type	a		c						
content architecture class	a		c		e		g	h	j
user-readable comments[1]	a		c		e				j
application comments[1]	a		c		e				j
user-visible name[1]	a		c		e				j
bindings	a		c		e				j
sealed	a	b	c	d	e	f	g	h	j
position	a		c		e		g		j
dimensions	a		c		e		g	h[2]	j
border	a	b	c	d	e	f	g	h	j
balance	a		c		e				j
layout path	a		c		e		g	h	j
layout stream categories	a		c		e				j
layout stream sub-categories	a		c		e				j
permitted categories	a		c		e		g		j
transparency	a	b	c	d	e	f	g	h	j
colour	a	b	c	d	e	f	g	h	j
colour of layout object	a	b	c	d	e	f	g	h	j
object colour table	a	b	c	d	e	f	g	h	j
content background colour	a	b	c	d	e	f	g	h	j
content foreground colour	a	b	c	d	e	f	g	h	j
content colour table	a	b	c	d	e	f	g	h	j
page position	a		c		e		g	h	j
medium type	a		c		e		g	h	j
presentation attributes	a	b	c	d	e	f	g	h[3]	j
protection	a		c		e		g		j
block alignment		b		d		f	g	h	j
concatenation		b		d		f	g		j
fill order		b		d		f	g		j
floatability range		b		d		f	g		j
indivisibility		b		d		f	g		j
layout category		b		d		f	g		j
layout object class		b		d		f	g		j
logical stream category		b		d		f	g		j
logical stream sub-category		b		d		f	g		j
new layout object		b		d		f	g		j
offset		b		d		f	g		j
same layout object		b		d		f	g		j
separation		b		d		f	g		j
synchronization		b		d		f	g		j
type of coding	a		c					h	j
coding attributes	a		c					h[3]	j[4]

[1] These attributes are not defaulted from attributes specified for styles.

[2] Only for the case that the attribute applies to a component of object type page. Otherwise the step does not apply.

[3] The presentation attributes and coding attributes for which a document application profile may specify non-standard default values are defined in those Specifications of the ITU-T Rec. T.410-Series | ISO/IEC 8613 that deal with individual content architectures.

[4] In the case that the attribute is defined to be defaultable.