INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

T.412

(11/1988)

SERIES T: TERMINAL EQUIPMENT AND PROTOCOLS FOR TELEMATIC SERVICES

OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT – DOCUMENT STRUCTURES

Reedition of CCITT Recommendation T.412 published in the Blue Book, Fascicle VII.6 (1988)

NOTES

1 CCITT Recommendation T.412 was published in Fascicle VII.6 of the *Blue Book*. This file is an extract from the *Blue Book*. While the presentation and layout of the text might be slightly different from the *Blue Book* version, the contents of the file are identical to the *Blue Book* version and copyright conditions remain unchanged (see below).

2 In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Recommendation T.412

OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT - DOCUMENT STRUCTURES [1)]

CONTENTS

1 *Introduction*

1.1 Scope

1.2 Normative References

1.3 Definitions, symbols and abbreviations

2 *Architectural principles*

2.1 Architectural concepts

2.2 Structural model of a document

2.3 Descriptive representation of a document

2.4 Document processing model

2.5 Roles of the document architecture constituents in document processing

3 *Document structures*

3.1 Specific structures

3.2 Specific logical structures

3.3 Specific layout structure

3.4 Documents containing both specific structures

3.5 Generic structures

4 *Interface between the document architecture and content architectures*

4.1 General concept

4.2 Specification of a content architecture

4.3 Interface information

5 *Attribute definitions*

5.1 General principles of attributes

5.2 Attribute specification format

5.3 Shared attributes

5.4 Layout attributes

5.5 Logical attributes

5.6 Layout style attributes

5.7 Layout directives

5.8 Presentation style attributes

5.9 Content portion attributes

[1)] This text is aligned with the final text of the corresponding international standard ISO 8613-2.

6 *Reference model of the document layout process*

6.1 Overview

6.2 Content and layout structure generation

6.3 Layout references and categories

6.4 The effect of some attributes on the document layout process

6.5 Layout process for frames

6.6 Allocation of areas for blocks

6.7 Alternative representation

7 *Reference model of the document imaging process*

7.1 Imaging order

7.2 Intersection principles

7.3 General rules for positioning pages on presentation surfaces

8 *Document architecture levels*

8.1 Definition of document architecture classes

8.2 The constituents of document architecture classes

8.3 Minimum attribute sets

Annex A (normative) – Notation used to represent document structures

A.1 Notation for structure diagrams

A.2 Notation for expressions

Annex B (informative) – Examples of document structures

B.1 Introduction

B.2 Notation used to specify constituents of a document

B.3 Introduction to specimen document

B.4 Specific structures

B.5 Processable form document with generic logical structure and generic layout structure

B.6 Specific layout structure

Annex C (informative) – Examples of document architecture levels

C.1 Document architecture levels

C.2 Attributes applicable to FDA document architecture levels

C.3 Attributes applicable to PDA document architecture levels

C.4 Attributes applicable to FPDA document architecture levels

Annex D (informative) – Examples of particular document features

D.1 Layout of frames

D.2 Layout of blocks

D.3 Further constraints on layout

D.4 Texture and imaging order attributes

D.5 Bindings, content generator

Annex E (informative) – The defaulting mechanism

E.1 The defaulting mechanism as applicable to each defaultable attribute

E.2 Determination of content portions and their associated attributes

Annex F (informative) – Attribute summary tables

1 Introduction

1.1 *Scope*

1.1.1 The purpose of the T.410 Series of Recommendations is to facilitate the interchange of documents.

In the context of the T.410 Series, documents are considered to be 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, geometric graphics elements and raster graphics elements, all potentially within one document.

Note – The T.410 Series is designed to allow for extensions, including typographical features, colour, spreadsheets and additional types of content such as sound.

1.1.2 The T.410 Series applies to the interchange of documents by means of data communications or the exchange of storage media.

The T.410 Series provides 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.

The T.410 Series also provides for the interchange of ODA information structures used for the processing of interchanged documents.

Furthermore, the T.410 Series allows for the interchange of documents containing one or more different types of content such as character text, images, graphics and sound.

1.1.3 This Recommendation:

- 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 three document architecture classes;
- defines a notation used for illustrating and describing document structures;
- provides examples of document architecture levels;
- provides examples of document structures;
- provides examples of particular document attributes.

1.2 *Normative References*

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

- ISO 6937-2, 1983, Information processing – Coded character sets for text communication – Part 2: Latin alphabetic and non-alphabetic graphic characters.
- Recommendation X.208 (1988), Specification of abstract syntax notation one (ASN.1).

1.3 *Definitions, symbols and abbreviations*

1.3.1 *Definitions*

For the purpose of this Recommendation the definitions given in Recommendation T.411 apply.

1.3.2 *Conventions*

The following conventions are used throughout this Recommendation.

1.3.2.1 *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.

1.3.2.2 *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.

1.3.2.3 *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 3.1.2). 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.

1.3.2.4 *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 § 3.1.2). 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 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.

2 Architectural principles

2.1 *Architectural concepts*

For the purpose of the T.410 Series of Recommendations, a document is an amount of structured information that can be interchanged as a unit.

This Recommendation 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 *structure 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 § 2.3.5). The document profile includes information relating to the document as a whole (see § 2.3.6).

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

2.2 *Structural model of a document*

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

2.2.1 *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/T.412.

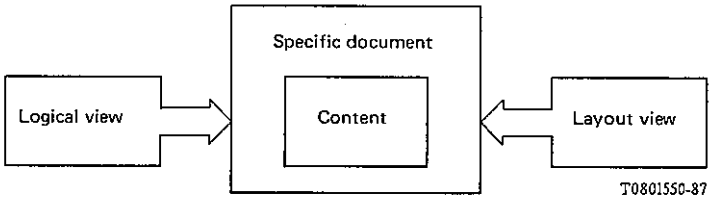


FIGURE 1/T.412

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.

2.2.2 *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 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 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 Recommendations of the T.410 Series. 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 § 4.

This interface allows the use of any of the content architectures defined in other Recommendations of the T.410 Series with the document architecture.

2.2.3 *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.

2.2.3.1 *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 T.410 Series of Recommendations 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.

2.2.3.2 *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 T.410 Series of Recommendations does not define particular document classes; however it provides the means by which document classes can be defined.

2.3 *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 T.410 Series of Recommendations 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 T.410 Series of Recommendations defines these constituents and attributes.

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

2.3.1 *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.

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

2.3.2 *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 § 2.3.4), for example a logo block or a standard paragraph, is called a *generic content portion description.*

2.3.3 *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.

2.3.4 *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 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 does 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 in 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 in termed a complete generic logical structure or a complete layout structure, respectively.

2.3.5 *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.

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.

2.3.6 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, interchange format class and document profile level 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 found in Recommendation T.414.

2.3.7 *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

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 § 2.3.9). In this case, the document profile indicates whether an external-document class description is referenced by the document, and if so, which one.

2.3.8 *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 examples of generic-documents.

2.3.9 *External-document class description*

An interchanged document that does not contain a generic structure can refer to an *external-document*. The external-document is identified in the interchanged document profile. 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.

2.3.10 *Resource-documents*

An object class description in a given interchanged document 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.

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

2.3.11 *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.

Paragraph 8 defines three document architecture classes that may be used in conjunction with document application profiles, as defined in Recommendation T.411.

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.

2.3.12 *Sets of constituents*

A document is represented by constituents, which are grouped into sets of constituents, and which have inter-relationships, as defined in this Recommendation.

The possible types of constituent in the descriptive representation of a document are shown in Figure 2/T.412.

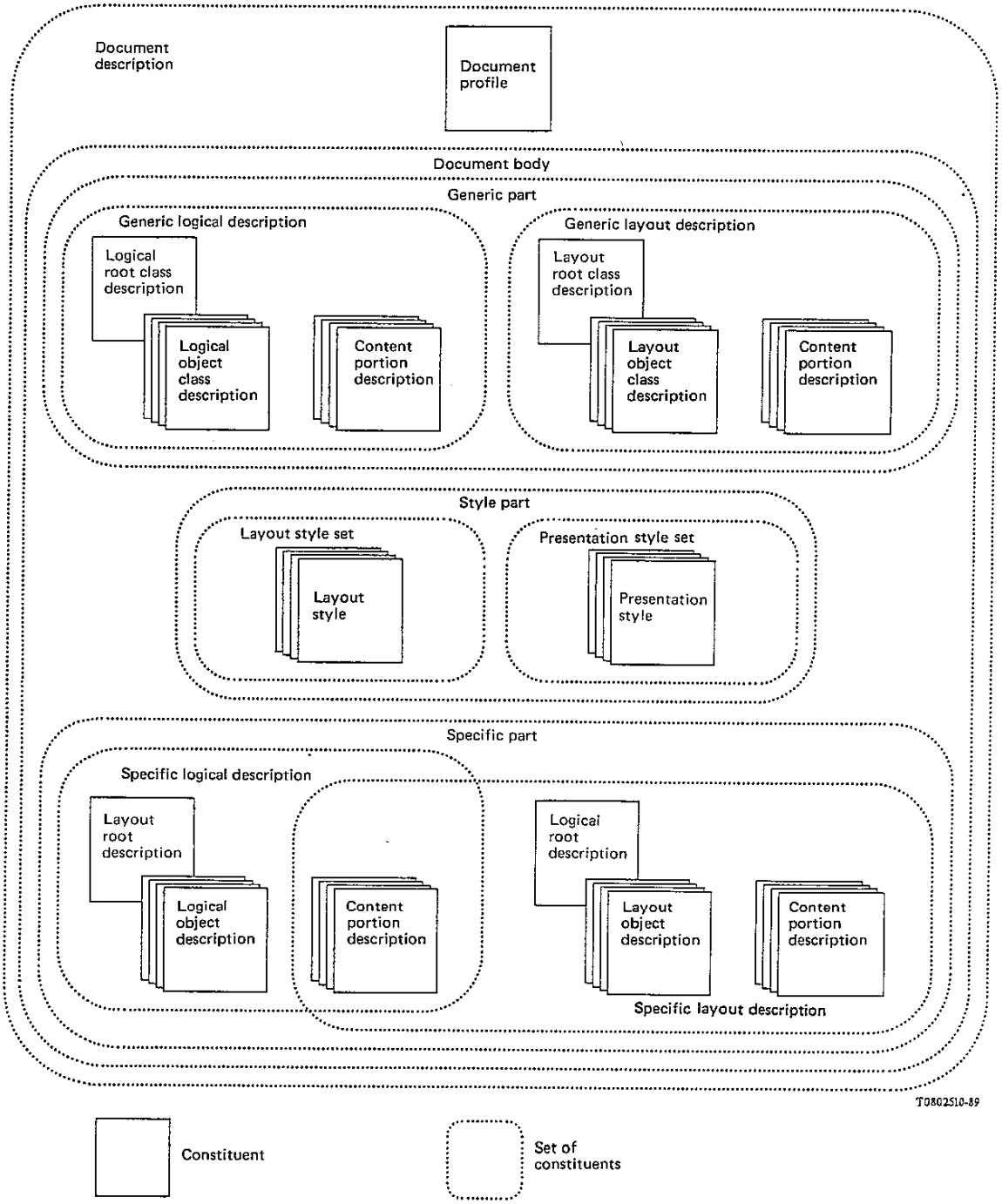


FIGURE 2/T.412

Descriptive model of a document

Figure 2/T.412 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 structure, and optionally style constituents;

   2) constituents representing the specific structure, and optionally style constituents;

3) constituents representing the generic structure, the specific structure, and optionally style constituents;

c) the constituents representing the generic structure 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 structure 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.

2.4 *Document processing model*

This paragraph describes a conceptual model for document processing.

It addresses only those aspects of document processing that are relevant to the document architecture defined in this Recommendation. 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 § 5.

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 document processing model (see Figure 3/T.412) is summarized in this paragraph. Three processes are shown:

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

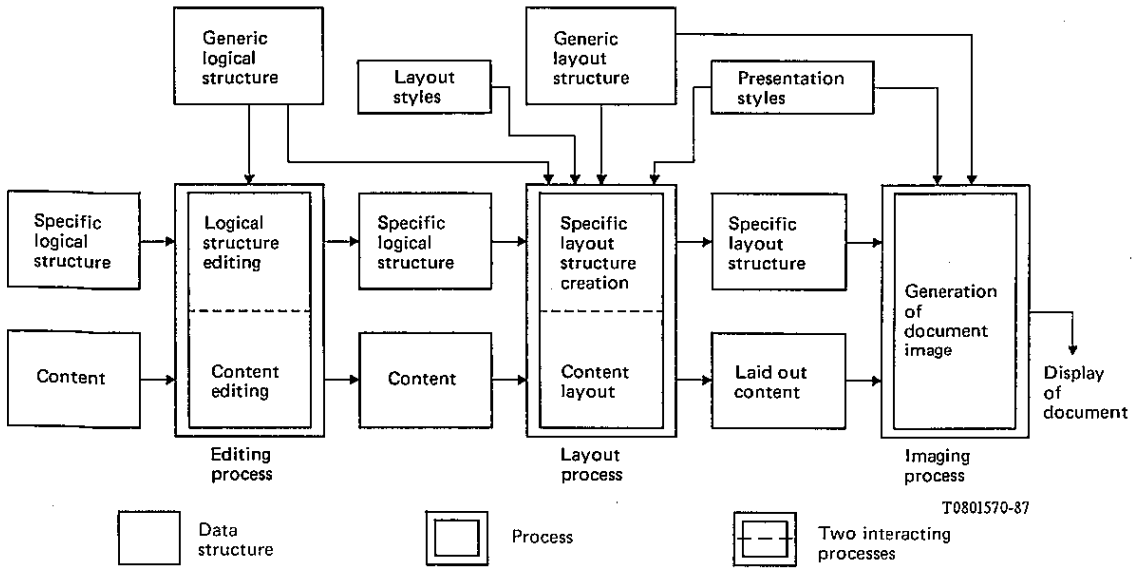


FIGURE 3/T.412

Document processing model

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 T.410 Series of Recommendations.

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 T.410 Series of Recommendations does not place any constraints on such editing.

2.4.1 *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.

2.4.2 *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.

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

The document layout process is described in § 6. The content layout process depends on the content architecture pertaining to the content being laid out and is described in those Recommendations of the T.410 Series 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.

2.4.3 *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 § 7. However, the imaging process is a locally defined process. Hence, apart from defining the input information required by this process, which is the specific and generic layout structures, referenced presentation styles and the formatted content portions, this process is not formally defined in the T.410 Series of Recommendations.

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 Recommendations of the T.410 Series that are concerned with particular content architectures.

2.5 *Roles of the document architecture constituents in document processing*

2.5.1 *Editing process*

This paragraph 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.

2.5.1.1 *Generic logical structure*

The generic logical structure can be used to control the 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, except that it controls only 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.

2.5.1.2 *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.

2.5.1.3 *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 it must be in processable or formatted processable form. The content is changed by:

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

b) adding, modifying or deleting embedded control functions.

Editing of formatted form content is outside the scope of the T.410 Series of Recommendations. 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 T.410 Series of Recommendations.

2.5.1.4 *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 T.410 Series of Recommendations.

2.5.1.5 *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.

2.5.1.6 *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 § 2.5.1.2). This involves the addition, deletion or modification of the layout directive attributes contained in a layout style. This can affect the layout characteristics of all the logical objects whose representation refers to the style and can affect the relationships that the objects have with other objects in the document. The result of changing the layout styles is that it may cause changes in the specific layout structure produced by the document layout process.

2.5.1.7 *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.

2.5.2 *Layout process*

This sub-section 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 § 6 of this Recommendation.

2.5.2.1 *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.

2.5.2.2 *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 directives which direct the document layout process and presentation attributes which direct the content layout process. These layout directives and presentation attributes are specified by referencing a layout style or a presentation style, respectively.

2.5.2.3 *Layout styles*

During the layout process, layout styles provide layout directives which direct the document layout process.

2.5.2.4 *Presentation styles*

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

2.5.2.5 *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.

2.5.2.6 *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.

2.5.2.7 *Specific layout structure*

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

2.5.2.8 *Generic content*

Generic content portions associated with logical object classes may be in formatted, processable or formatted processable form. Whenever such a logical object class specifying a generic content portion is referred to from a basic logical object in a specific logical structure, 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 portion 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.

2.5.3 *Imaging process*

This paragraph 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.

2.5.3.1 *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.

2.5.3.2 *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.

2.5.3.3 *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 change the precedence of the layout objects by specifying an imaging order which is different from the sequential layout order.

2.5.3.4 *Layout styles*

In the imaging process, layout styles play no role.

2.5.3.5 *Presentation styles*

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

3 Document structures

3.1 *Specific structures*

3.1.1 *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 directly associated (see 3.1.3). 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 § 3.2 and 3.3, 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.

3.1.2 *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.

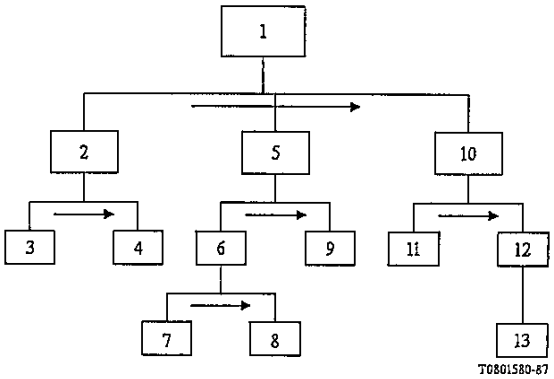
This sequential order is such that each object in the structure is succeeded by all of its immediate subordinates, before any other objects with the same immediate superior. Each of the immediate subordinates is followed by all of its immediate subordinates, before proceeding to the next immediate subordinate 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 that the logical objects are intended to be processed by the layout process (see § 6).

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 § 7.1).

An example of the sequential order of a structure is shown in Figure 4/T.412, where the numbers indicate the sequential order.



Note – Arrows define sequential order.

FIGURE 4/T.412

Example of sequential order

3.1.3 *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.

3.2 *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 T.410 Series of Recommendations 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.

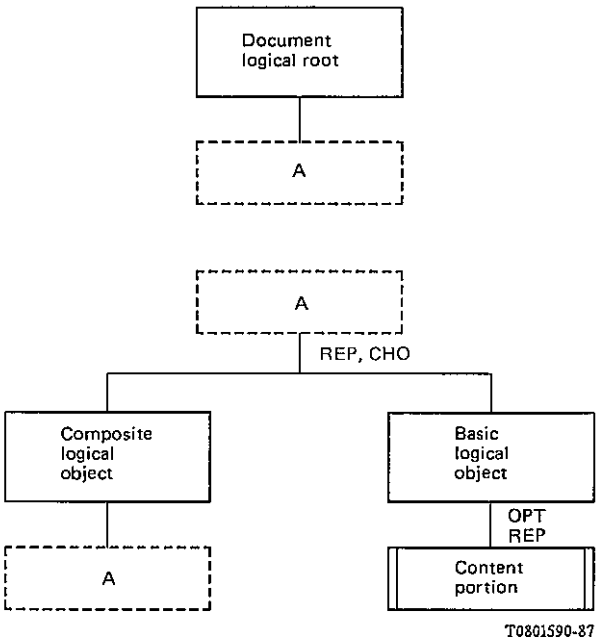
Logical objects provide the means to specify the specific logical structure of a document that is relevant to a particular application of the T.410 Series of Recommendations. For example, the specific logical structure may be required in the processing of a document, such as editing and laying out the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

Paragraph 5 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/T.412.



Note – The notation used in this Figure is that defined in Annex A.

FIGURE 5/T.412

Permissible specific logical structures

3.2.1 *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.

3.2.2 *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 any given basic logical object and the document logical root is variable.

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 T.410 Series of Recommendations. For example, these relationships may be used to specify which sections, paragraphs and diagrams relate to a certain chapter in a document.

3.2.3 *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 directly associated with a basic logical object. In the case that zero content portions are directly associated, the content is either specified in the form of an expression for generating content or is derived from a basic object class description.

3.3 *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.

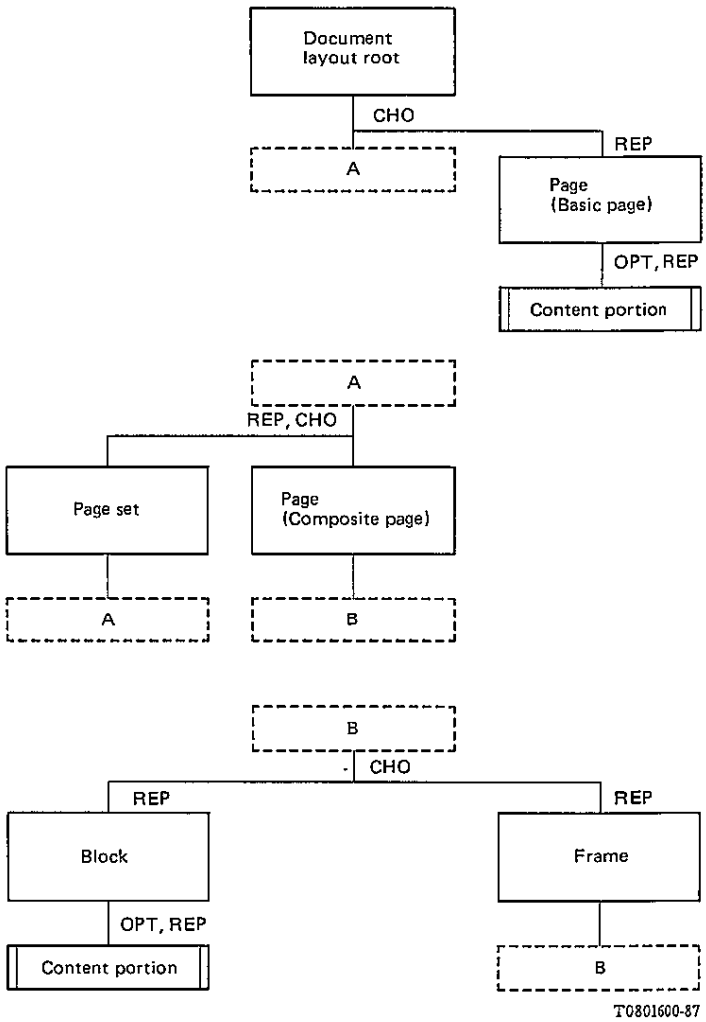
Paragraph 5 defines the attributes that are used to describe the characteristics of layout objects, including the use of presentation styles.

3.3.1 *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/T.412.



Note – The notation used in this figure is that defined in Annex A.

FIGURE 6/T.412

Permissible specific layout structures

3.3.1.1 *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.

3.3.1.2 *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 any given page and the document layout root is variable.

3.3.1.3 *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.

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

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

If a page is a composite object, then its immediate subordinates consist of either any number of frames or any number of blocks.

3.3.1.4 *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. 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. The number of hierarchical levels between any given block and the enclosing page is variable.

3.3.1.5 *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 directly associated with a block. In the case that zero content portions are directly associated, the content is derived from a basic object class description.

3.3.2 *Positioning layout objects*

This sub-section 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.

3.3.2.1 *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/T.412. This Figure also defines names for 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.

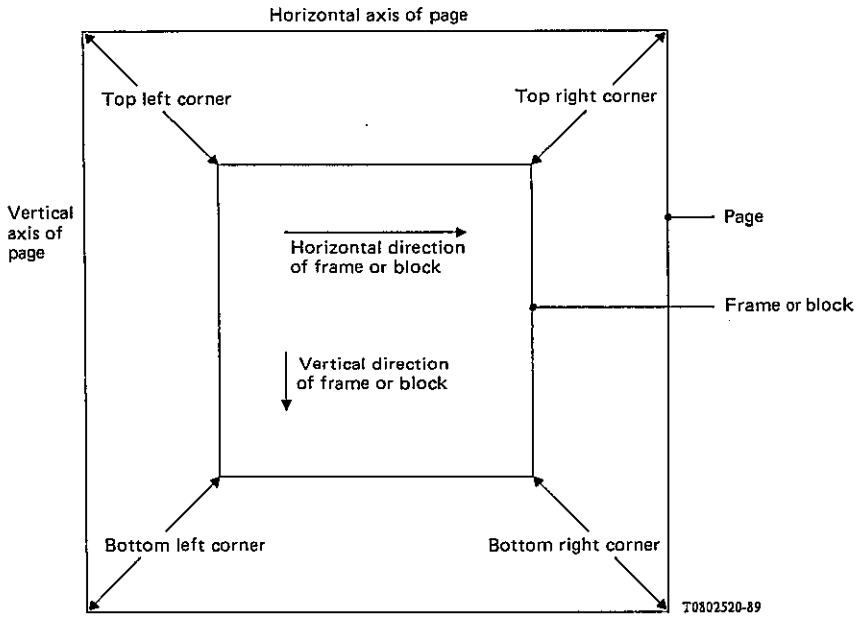


FIGURE 7/T.412

The horizontal and vertical directions of layout objects

3.3.2.2 *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 co-ordinate system. Frames which are immediately subordinate to another frame are positioned relative to the reference point of that frame.

3.3.2.3 *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 co-ordinate system. Blocks that are immediately subordinate to a frame are positioned relative to the reference point of that frame.

3.3.3 *Naming of edges of frames and blocks*

In the following definition the layout path (see § 5.4.2.2, 6.5 and 6.6) 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^{o} anti-clockwise relative to the direction of the layout path.

The names of the edges of frames and blocks are illustrated in Figure 8/T.412.

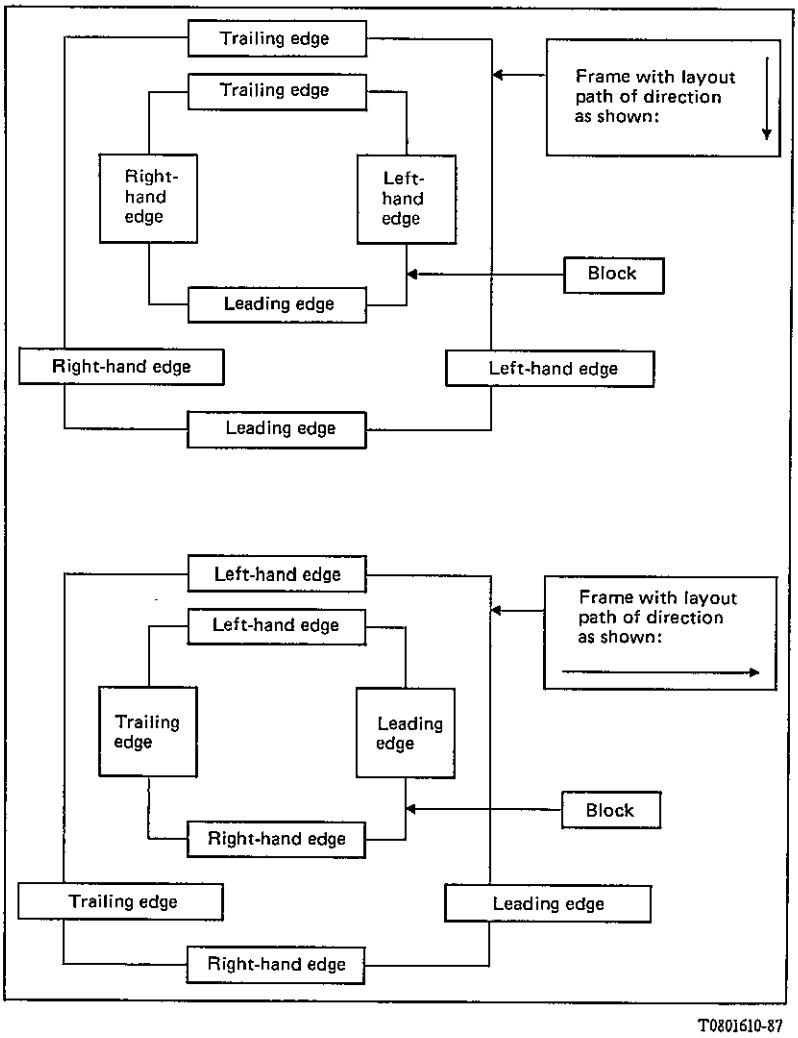


FIGURE 8/T.412

Naming of the edges of blocks and of their immediately superior frames

3.3.4 *Measurement*

3.3.4.1 *Basic measurement unit*

For the purpose of conveying the originator's intentions, all dimensions and positions are expressed in basic measurement units (BMU). The value of the *basic measurement unit* 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.

3.3.4.2 *Scaled measurement units*

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 Recommendation T.414). 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.

3.3.5 *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/T.412 and 10/T.412).

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.

3.3.5.1 *Borders of blocks*

A border around a block is illustrated by Figure 9/T.412.

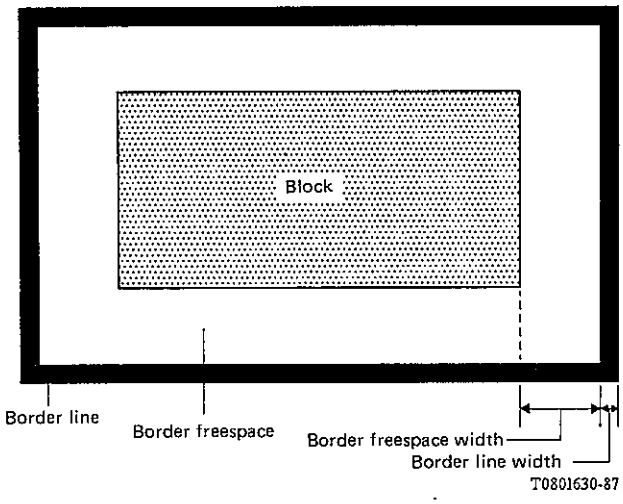


FIGURE 9/T.412

Border around a block

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 must take the border into account in determination of the available area.

3.3.5.2 *Borders of frames*

A border within a frame is illustrated by Figure 10/T.412.

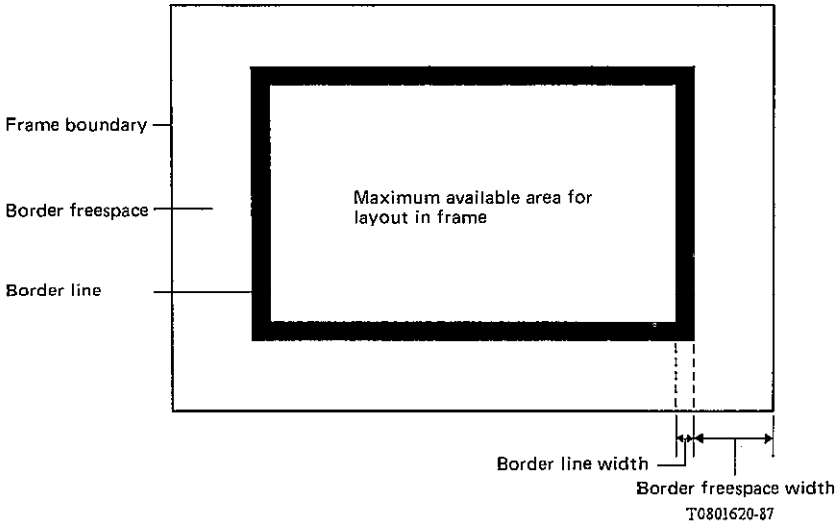


FIGURE 10/T.412

Border within a frame

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 must take the border into account in determination of the available area.

3.4 *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 these 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 may 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/T.412 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/T.412 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.

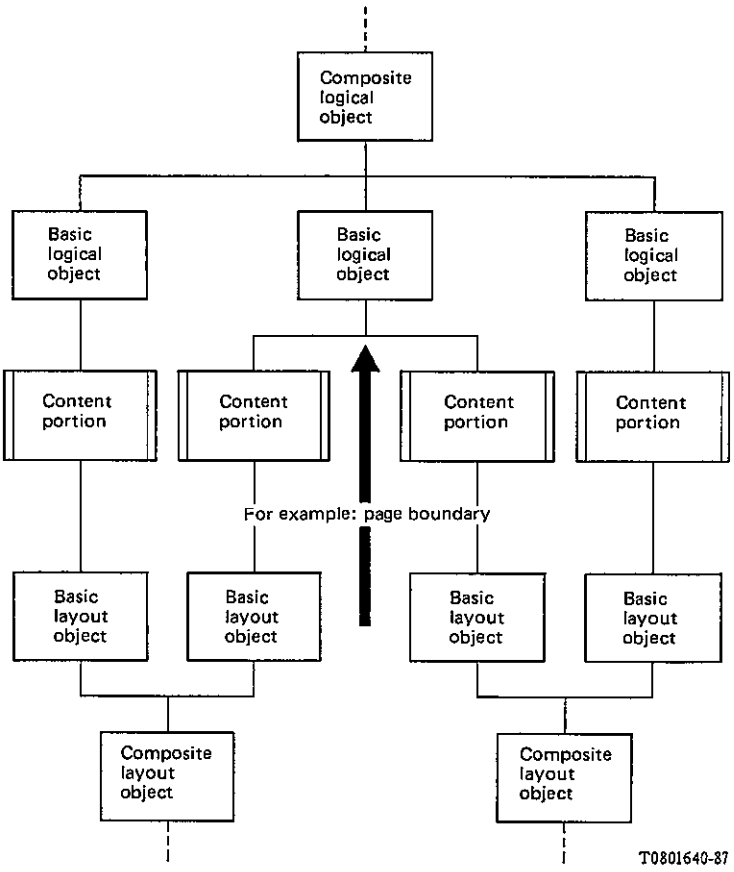


FIGURE 11/T.412

Example of relationship between logical and layout objects and associated content portions

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

3.5 *Generic structures*

3.5.1 *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 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 §§ 3.5.6 and 3.5.7.

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.

3.5.2 *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 constrain 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 or may 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 constrain 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.

3.5.3 *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 § 5.4.2.3).

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 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 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 § 3.5.7).

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

3.5.4 *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 § 5.3.2.1). 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 T.410 Series of Recommendations does not define any particular generic structures or object classes. However, it allows the definition of a potentially infinite variety of document classes.

3.5.5 *Content in generic structures*

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

- by specification of one or more generic content portions;
- by specification of 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 description 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.

3.5.6 *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, a reference to an object class in a resource-document which has content specified, or a content generator.

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

3.5.7 *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 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.

3.5.8 *Resource-document*

A logical object class description may contain a reference to a logical object class description in the resource-document (see § 2.3.10). 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.

4 Interface between the document architecture and content architectures

4.1 *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;
- of 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 section:

- *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 12/T.412.

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).

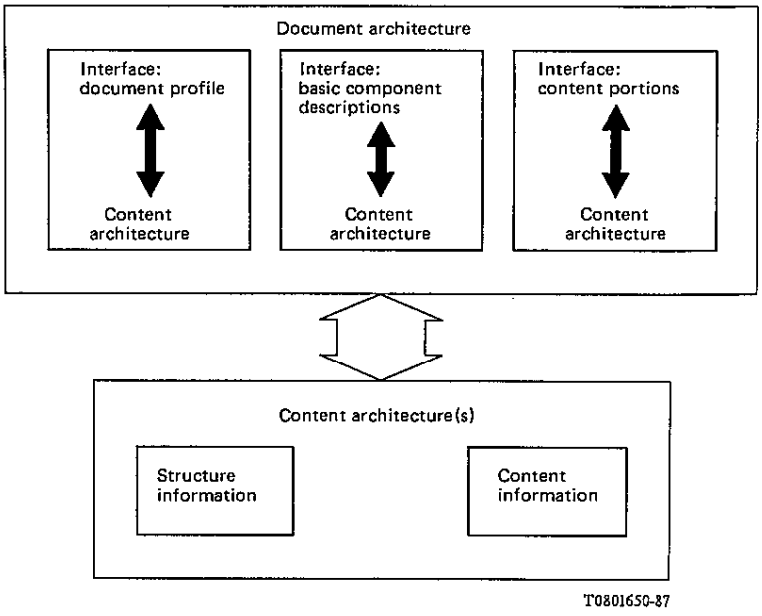


FIGURE 12/T.412

Document architecture - Content architecture interface

4.2 *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(e), through the interface information.

4.3 *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.

4.3.1 *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 Recommendation T.411.

4.3.2 *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.

4.3.3 *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.

5 Attribute definitions

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

5.1.1 *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.

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 Recommendation T.414. The general principles of other attribute categories are described in this paragraph and the attributes are defined in §§ 5.3-5.9.

Note – The Tables in Annex F summarise all attributes defined in this section, in accordance with their categorisation.

5.1.1.1 *Attributes of components*

Attributes of component descriptions are further categorised as follows:

- shared attributes, these can be included in both logical and layout component descriptions (see § 5.3);
- layout attributes, these can be included in layout component descriptions only (see § 5.4);
- logical attributes, these can be included in logical component descriptions only (see § 5.5);

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 § 5.1.1.4), are independent of the content architecture pertaining to any component descriptions.

5.1.1.2 *Layout style attributes*

A layout style consists of:

- a layout style identifier;
- a user-visible name;
- user-readable comments; and,
- 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.

Layout styles are defined in § 5.6.

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 directives to that logical component description which contains the reference. Precedence rules are specified in §5.1.2.4 and 5.7.12.

The layout directives are a set of attributes which specify information for the document layout process (see § 6). The layout directives guide the document layout process in generating the specific layout structure.

Layout directive attributes are defined in § 5.7.

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.

5.1.1.3 *Presentation style attributes*

A presentation style consists of:

- a presentation style identifier;
- a user visible name;
- user-readable comments;
- transparency;
- colour;
- border; and,
- sets of presentation attributes.

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.

Presentation styles are defined in § 5.8.

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 § 5.1.2.4.

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.

5.1.1.4 *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 § 5.3.4).

The presentation attributes are defined in Recommendations of the T.410 Series that are concerned with individual content architectures.

5.1.1.5 *Content portion description attributes*

A content portion description consists of:

- content identifier-logical;
- content identifier-layout;
- type of coding;
- alternative representation;
- coding attributes; and,
- 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 § 5.9. Other attributes of content portions depend on the content architecture and details are specified in those Recommendations of the T.410 Series that deal with individual content architectures.

Content portion descriptions are defined in § 5.8.

5.1.2 *Attribute values*

5.1.2.1 *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 § 5.1.2.4. 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).

Attributes may be structured into parameters and sub-parameters, each attribute definition includes a specification of any such structuring.

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 predefined set of values.

Such a 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 an attribute value of type c) is the sub-parameter "fill order" of the attribute "position" (see § 5.4.1.1) which can have a value equal to a member of the set {'normal order', 'reverse order'}. An example of an attribute value of type d) is the attribute "presentation style". An example of an attribute value of type e) is the parameter "binding name" of the attribute "bindings".

Alternatively, if the value of an attribute 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 § 5.1.3).

5.1.2.2 *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 must 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 mechanism specified in § 5.1.2.4.
- 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 § 5.1.2.4, 5.1.2.5).

A document application profile may specify non-standard default values for attributes (see Recommendation T.414). If this is the case they are declared in the document profile by means of the document profile attribute "document application-profile defaults".

5.1.2.3 *Default value lists*

Certain attributes that are classified as defaultable may be specified in a default value list. Default value lists can 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 in the specific layout and specific logical structures, to be specified in constituents that correspond to objects at higher levels of the hierarchical structure. The use of a default value list forms part of the defaulting mechanism defined in § 5.1.2.4.

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

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

5.1.2.4 *Determining values for defaultable attributes of objects*

The values of defaultable attributes of objects can 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 higher levels 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 this Recommendation itself.

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

For example, 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 can be specified that the defaulting rules are to apply to each parameter independently. When this is possible it is explicitly specified in the attribute definition, in the part of the definition which specifies the default values.

a) If an attribute value is specified for the object description concerned, then that value is used.
b) If the object description concerned refers to a style, and that style contains a value for the attribute, then that value is used.
c) If the object description concerned refers to an object class description, and that object class description contains 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 contains a reference to a style which in turn contains a value for the attribute, 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 contains 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 contains a reference to a style which in turn contains a value for the attribute, then that value is used.

g) If a default value list specified at a higher level of the hierarchical structure is applicable to the object (that is, it is specified for this object type), and that 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.

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

At each level, if a default value list exists for both an object description and an object class description, then only default values which are not specified in the default value list of the object description are taken from the default value list of the object class description.

At each level, if the default value list exists for an object class description in the resource-document that is referred to by an object class description that is referred to by an object description, then only default values which are not specified in the object description or in the object class description in the interchanged document are taken from the object class description in the resource-document.

At each level, the default value list may specify an attribute value directly, or indirectly via a style. In the case that an attribute value is specified in both ways in a single default value list, the value specified explicitly in the default value list is used.

h) If no value is determined by the preceding steps a)-g), and a default value is defined for this attribute by the document profile attribute "document application profile defaults" (see § 4 of Recommendation T.414) then that value is used.

j) If no value is determined by the preceding steps a)-h), then the default value defined in the T.410 Series of Recommendations is used.

5.1.2.5 *Determining values of attributes of content portions*

The content associated to 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 an attribute value is specified for the content portion description concerned, then that value is used;

b) If no value is determined by step a), and a default value is defined for this attribute by the document profile attribute "document application profile defaults" (see Recommendation T.414) then that value is used;

c) If no value is determined by the preceding steps a), b), then the default value defined in the T.410 Series of Recommendations is used.

5.1.3 *Expressions*

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

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

The expression permitted in the attribute "generator for subordinates" is defined in § 5.3.2.1. The other possible expressions are defined in this sub-sub-section.

There are three types of expressions:

a) string expressions (see § 5.1.3.1);

b) numeric expressions (see § 5.1.3.2);

c) object identifier expressions (see § 5.1.3.3).

A string expression or a numeric expression may refer to a binding value (see § 5.1.3.4).

5.1.3.1 *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 § 5.1.3.4);

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.

Each of these functions has a parameter consisting of a numeric expression (see § 5.1.3.2).

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.

5.1.3.2 *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 § 5.1.3.4);

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 specified object, within the set of objects that are immediately subordinate to the immediate superior of the specified object and belong to the same object class as the specified 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).

5.1.3.3 *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 end the specification of one or two parameters as 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 § 5.1.3.5).
- SUPERIOR-OBJECT: a function with a 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 a parameter, which produces the identifier of the object description that immediately precedes the object description specified by the parameter in the sequential order, (see § 3.1.2). 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.)

5.1.3.4 *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 having the expression applies 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 § 5.3.5.3).

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 § 5.1.3.3, 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 selections functions:

- CURRENT-OBJECT;
- CURRENT-INSTANCE;
- SUPERIOR;
- PRECEDING.

The parameters of *binding selection functions* are as defined in § 5.1.3.3, 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 § 5.1.3.3 and § 5.1.3.5.

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 § 5.1.3.3.

The differences are that, if the result of the function is the identifier of an object description that does not have an attribute "bindings" including a binding with 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 that has an attribute "bindings" including a binding with 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 that has a binding with the specified binding name, 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 can contain an expression, which may itself refer to further expressions.

5.1.3.5 *Current instance function*

The current-instance function can be used in object identifier expressions (see § 5.1.3.3) and in binding reference expressions (see § 5.1.3.4). 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 any content is laid out 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 any content is laid out 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.)

5.2 *Attribute specification format*

The attribute definitions in this section are structured as follows:

Constituents

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

Classification

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

Structure

States the structuring of the attribute into parameters and sub-parameters, 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 and sub-parameters 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 Recommendation T.415.

Default values

States the default values of the attribute [see § 5.1.2.4 j)]. If the attribute is structured into parameters and sub-parameters 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. The rationale for these exceptional cases is issues such as particular optimisations available with restricted usage of the attribute and backwards compatibility with previous standards.

5.3 *Shared attributes*

The attributes defined in this paragraph can 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 clauses.

5.3.1 *Identification attributes*

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

5.3.1.1 *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 value is always determined by one of the steps a) or c) in § 5.1.2.4.

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 § 5.3.3.2). 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.

5.3.1.2 O*bject 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 values assigned to the first integer are:

- 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 a single 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 Recommendation T.415), 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 CCITT Recommendations.

5.3.1.3 *Object class identifier*

Constituents

Object class descriptions.

Classification

Mandatory.

Permissible values

A sequence of non-negative integers. The values assigned to the first integer are:

- 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 end 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 must be unique.

5.3.2 *Construction attributes*

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

5.3.2.1 *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 object class description, 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.

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 must be one of the values that can be generated by the attribute "generator for subordinates" 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 as 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 may 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 may not be evaluated, when the containing construction term is evaluated.

5.3.2.2 *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 § 5.1.3.1.

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 the single one 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 directives 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.

5.3.3 *Relationship attributes*

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

5.3.3.1 *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 § 5.3.1.3).

5.3.3.2 *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.

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 § 5.3.1.2). The sequence contains integers corresponding to each immediately subordinate object description and the same integer may 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 § 7.1, 7.2), unless overridden by the attribute "imaging order" (see § 5.4.3.1).

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 Recommendation T.415), 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 Recommendation T.415.

Note – This exceptional case is provided for compatibility with CCITT Recommendations.

5.3.3.3 *Content portions*

Constituents

Basic component descriptions.

Classification

Non-mandatory.

Permissible values

A sequence of one or more non-negative integers.

Definition

This attribute specifies which content portions are 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 § 5.9.1). The sequence contains integers corresponding to each content portion of the component concerned and the same integer may 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 must 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 has associated generic content portion descriptions;
b) the basic object description specifies a content generator, this is only possible 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 Recommendation T.415), 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 Recommendation T.415.

Note – This exceptional case is provided for compatibility with CCITT Recommendations.

5.3.3.4 *Resource*

Constituents

Object class descriptions.

Classification

Non-mandatory.

Permissible values

A string of characters from the minimum subrepertoire of ISO 6937-2.

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 § 2.3.10 and § 3.5.9).

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.

5.3.3.5 *Presentations style*

Constituents

Basic component descriptions.

Classification

Non-mandatory.

Permissible values

Either a presentation style identifier or "null".

Default value

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.

5.3.4 *Content architecture class attributes*

5.3.4.1 *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.

Default value

'Formatted character content architecture', as defined in Recommendation T.416.

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.

5.3.4.2 *Content type*

Constituents

Basic layout component descriptions.

Classification

- Non-mandatory for object class descriptions;
- defaultable for object descriptions.

Permissible values

'Formatted raster graphics content architecture', as defined in Recommendation T.417.

Default value

'Formatted raster graphics content architecture'.

Exceptions

This attribute provides an alternative to the attribute "content architecture class" as a means to specify the content architecture class of the content associated with the basic component. The attribute follows the same rules for combination with other attributes as are specified in the T.410 Series of Recommendations for "content architecture class".

This attribute is ignored when a value is specified for the attribute "content architecture class" or, in the case of an object description, when a value of "content architecture class" applies that is derived from any of the steps a) to f) of § 5.1.2.4.

Note – This attribute is provided for compatibility with CCITT Recommendations.

5.3.5 *Miscellaneous attributes*

5.3.5.1 *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. The character set is that specified in the document profile attribute "comments character sets". The default character set is the minimum subrepertoire of ISO 6937-2. In addition to the graphic character set, the control functions carriage return and line feed may be included in the character string. Code extension control functions for the designation and invocation of graphic character sets may also be included.

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 Recommendation, nor for any content layout or imaging processes, defined in other Recommendations of the T.410 Series.

5.3.5.2 *Application comments*

Constituents

Component descriptions.

Classification

- Non-mandatory for object class descriptions;
- defautable for object descriptions.

Permissible values

An octet string.

Default value

'empty string'

Definition

This attribute shall be used for application dependent comments.

The attribute has no significance for the reference models of the layout or imaging processes defined in this Recommendation, nor for any content layout or imaging processes, defined in other Recommendations of the T.410 Series.

This character sequence is not part of the document content. It shall be possible to process the document ignoring the value of this attribute.

5.3.5.3 *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. The character set is that specified in the document profile attribute "comments character sets". The default character set is the minimum subrepertoire of ISO 6937-2. In addition to the graphic character set, the control functions carriage return and line feed may be included in the character string. Code extension control functions for the designation and invocation of graphic character sets may also be included.

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 directly access an object by name. The attribute has no significance for the reference models of the layout or imaging processes defined in this Recommendation, nor for any content layout or imaging processes, defined in other Recommendations of the T.410 Series.

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". However, such values are not defined by this Recommendation.

5.3.5.4 *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 6937-2.

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 § 5.1.3;
- in the case of layout object descriptions, a string literal, a numeric literal, or an object identifier.

Default values

Each named binding is independently defaultable. For each possible 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 ultimately relate to the value of an attribute.

In the T.410 Series of Recommendations this attribute shall only be used in the evaluation of the content specified by the attribute "content generator".

5.3.5.5 *Default value lists*

Constituents

Composite component descriptions.

Classification

Non-mandatory.

Structure

A set of one or more lists of attributes, such that each list applies to a different subordinate object type.

Within such a set of lists, there should be only one list that pertains to a particular object type.

Permissible values

One or more default value lists.

Definition

This attribute specifies default attribute values for subordinate object descriptions.

Table 1/T.412 lists the attributes that may be included in a list for each object type.

TABLE 1/T.412

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 Content type Dimensions Transparency Colour Page position Medium type Presentation attributes
Frame	Position Dimensions Border Layout path Permitted categories Transparency Colour
Block	Presentation style Content architecture class Content type Position Dimensions Border Transparency Colour Presentation Attributes
Composite logical object	Protection Layout style
Basic logical object	Presentation style Content architecture class Content type Protection Layout style

In the case of a page, if the list applies to a composite page, the attribute "content architecture class", "content type", "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.

5.4 *Layout attributes*

The attributes defined in this clause are applicable to layout components only.

5.4.1 *Property attributes*

These attributes specify the characteristics that are applicable to layout components.

5.4.1.1 *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", "vertical position" or, the parameter "variable position", which has four sub-parameters: "offset", "separation", "alignment" and "fill order".

The sub-parameter "offset" is structured into four sub-sub-parameters, "leading offset", "trailing offset", "left-hand offset", "right-hand offset".

The sub-parameter "separation" is structured into three 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 subparameter. 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 a parameter is specified without including a value for one or more of its sub-parameters or sub-sub-parameters then the following defaults are defined for the unspecified sub-parameters or sub- sub-parameters:

For the parameter "fixed position":

- "horizontal position": 0
- "vertical position" : 0

For the parameter "variable position":

- for the sub-parameter "offset":
  - "leading offset": 0
  - "trailing offset": 0
  - "left-hand offset": 0
  - "right-hand offset": 0
- for the sub-parameter "separation":
  - "leading edge": 0
  - "trailing edge": 0
  - "centre separation": 0
- for the sub-parameter "alignment", the default value is 'right-hand aligned'
- for the sub-parameter "fill order", the default value is 'normal order'

Definition

This attribute specifies the position of the object relative to the object at the next higher level in the hierarchical structure (i.e. either 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 must 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 the 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 as a multiple of scaled measurement units.

The case of variable position may only be specified for frame class descriptions referred to in construction expressions 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 one 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 § 3.3.3.)

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 default value(s) are assumed.

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 immediately 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 must be equal to, or greater than, the greater of the value of "leading edge" for the first of the frames in the direction of layout path and the value of "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 must be equal to, or greater than, the greater of the values of "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 § 3.3.3.)

If the parameter "variable position" does not specify a value for this sub-parameter, or for any of its sub-sub-parameters, then the default value(s) are assumed.

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', 'cent-red', 'left-hand aligned'.

Subject to satisfying constraints on placement specified by the sub-parameter "offset" 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 § 3.3.3).

It is possible to independently align each of a set of sibling frames within their common immediately superior layout object.

If the parameter "variable position" does not specify a value for this sub-parameter then the default value is assumed.

d) *fill order*

The 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 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 sibling frames 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 sibling frames 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 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 § 3.3.3.)

If the parameter "variable position" does not specify a value for this sub-parameter then the default value is assumed.

This attribute is subject to the precedence rule that satisfying constraints on the positioning of other sibling frames which contain any content laid out earlier in sequential logical order than any in this frame, has precedence over satisfying constraints on the positioning of this frame.

5.4.1.2 *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" includes 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 (see exception case);

Case 2:

- "horizontal dimension": one of four sub-parameters:
- "vertical dimension": one of four sub-parameters:
  - "fixed dimension": positive integer;
  - "Rule A": 'applies', with two optional sub-sub-parameters:
    - "minimum dimension": positive integer,
    - "maximum dimension": positive integer;
  - "Rule B": 'applies' with two optional sub-sub-parameters:
    - "minimum dimension": positive integer,
    - "maximum dimension": 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 § 7.3)

Definition

In either case, this attribute consists of an ordered pair of parameters, corresponding to the dimensions in the horizontal and vertical directions of the component in scaled measurement units. The first parameter of the pair specifies the dimension in the horizontal direction, the second parameter specifies the dimension in the vertical direction.

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 "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 dimension concerned is to take its default value.

By the use of the sub-parameters in a), b) and c), each dimension of a frame can have either a fixed size or a variable size.

Rule A and Rule B may only he specified for frame class descriptions referred to in construction expressions only from other frame class descriptions. Consequently, block and page component descriptions, frame descriptions and frame class descriptions referred to in construction expressions specified by page class descriptions may not specify these rules. Further, Rule A may only be specified for the dimension in same direction as the layout path of the immediately superior frame.

In the case of "Rule A" or "Rule B", two further sub-sub-parameters can 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 "minimum dimension" then the value will be that specified by "minimum dimension". If the value determined by the rule would otherwise be larger than the value of "maximum dimension" then the value will be that specified by "maximum dimension".

The minimum and maximum dimensions 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.

Exceptions

The sub-parameter "variable page height" may only be specified for basic page components, and for the vertical dimension only.

Note – The sub-parameter "variable page height" is provided for compatibility with existing CCITT Recommendations.

5.4.1.3 *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

Zero or more of the parameters "left-hand-edge", "right-hand-edge", "trailing-edge" and "leading-edge".

Each parameter either has the value 'null' or consists of zero or more of the sub-parameters "border line width", "border line type", "border freespace width".

Permissible values

For each of the four parameters:

- 'null';

or, a combination of:

- "border line width": any non-negative integer
- "border line type": one of the values 'solid', 'dashed', 'dot', 'dash-dot', 'dash-dot-dot', 'invisible'.

- "border freespace width": any non-negative integer

Default values

For each of the four parameters:

- "border line width": 20
- "border line type": "solid"
- "border freespace width": 0

Definition

This attribute specifies a border, consisting of a border line and border freespace, for the edges of a frame or block component (see § 3.3.5). 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.

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 should be drawn for the edge.

5.4.2 *Formatting attributes*

These attributes specify the information applicable to the formatting of the document.

5.4.2.1 *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 these requirements are not satisfied then the attribute balance should be ignored.

All object descriptions and object class descriptions identified by this attribute must have the same value for layout path and the same set of "permitted categories". If the same value for layout path or the same set of permitted categories are not specified then the attribute balance should be ignored.

The value 'null' indicates that no constraints on the sizes of the subordinate layout objects are specified by this instance of this attribute.

5.4.2.2 *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°'.

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".

5.4.2.3 *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 only contain 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 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".

5.4.2.4 *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 6937-2, each being the name of a layout category.

Default value

'null'.

Definition

This attribute specifies the layout categories permitted for logical objects the content of which is to be laid out within the frame.

A layout category has a name 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.

A frame can specify any number of layout categories. The content associated with a basic logical object is constrained to be placed within frame(s) that specify the same layout category as that specified by the logical object. This provides for the layout process to consist of a number of separate layout streams (see § 6).

The value of this attribute is the set of names of the layout categories permitted.

This attribute is only significant 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 is permitted to be placed in the frame, including content associated with logical objects which do not specify a layout category.

5.4.3 *Imaging attributes*

These attributes specify the information applicable to imaging the document.

5.4.3.1 *Imaging order*

Constituents

Composite page or frame object descriptions.

Classification

Non-mandatory.

Permissible values

A sequence of one or more non-negative integers.

Definition

This attribute specifies the precedence for imaging of 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 § 5.3.1.2). The sequence contains integers corresponding to each immediately subordinate object description and the same integer may 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.

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 § 7.2.

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 § 5.3.3.2).

5.4.3.2 *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.

When two or more frames and/or blocks intersect, the effect of combination is determined from the imaging order, as described in § 7.1.

Transparency is in effect only when blocks and/or frames actually intersect. The use in combination of the attributes "imaging order", "transparency" and "colour" is described in § 7.2.

5.4.3.3 *Colour*

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, 'colourless', 'white'.

Default value

'colourless'

Definition

This attribute defines the colour of a page, frame or block.

The value 'white' shall only be specified if the attribute "transparency" has 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 § 7.1. The use in combination of the attributes "imaging order", "transparency" and "colour" is described in § 7.2.

5.4.3.4 *Page position*

Constituents

Page component descriptions.

Classification

- Non-mandatory for object class descriptions;
- defaultable for object descriptions.

Permissible values

A pair of non-negative integers.

Default value

Such that edge losses are minimised.

Definition

This attribute specifies the position of the layout object page within a nominal page (see § 7.3).

The value of this attribute is an ordered pair of integers 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 first integer of the pair specifies the horizontal distance, the second specifies the vertical distance.

The use of this attribute is further described in § 7.3.

5.4.3.5 *Medium type*

Constituents

Page component descriptions.

Classification

- Non-mandatory for object class descriptions;
- defaultable for object descriptions.

Structure

Two parameters, "nominal page size" and "side of sheet".

Permissible values

For the parameter "nominal page size": a pair of positive integers.

For the parameter "side of sheet", a set of data elements defined for the parameter, 'recto', 'verso', 'unspecified'.

Default values

The two parameters are independently defaultable.

The default values are:

- "nominal page size": dimensions for ISO A4 (see § 7);
- "side of sheet": 'unspecified'.

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 first integer of the pair specifies the horizontal dimension, the second specifies the vertical dimension.

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.

The use of this attribute is described further in § 7.

5.4.4 *Presentation attributes*

Constituents

Basic page or block component descriptions, presentation styles.

Classification

- Non-mandatory for object class descriptions;
- non-mandatory for presentation styles;
- defaultable for object descriptions.

Permissible values

Any set of presentation attributes applicable to a particular content architecture class, as defined in those Recommendations in the T.410 Series which specify the individual content architectures.

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.

5.5 *Logical attributes*

The attributes defined in this sub-section are applicable to logical components only.

5.5.1 *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 all content portions referred to from that description.

5.5.2 *Layout style*

Constituents

Logical component descriptions.

Classification

Non-mandatory.

Permissible values

A layout style identifier or 'null'.

Default value

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 this logical component description.

5.6 *Layout style attributes*

5.6.1 *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.

5.6.2 *Attributes which can be specified for layout styles*

The following attributes can be specified for layout styles:

- layout style identifier (see § 5.6.1);
- user-readable comments (see § 5.3.5.1);
- user-visible name (see § 5.3.5.3);
- layout directive attributes (see § 5.7);

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" and "user-visible name" 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.

5.7 *Layout directives*

A layout directive is an attribute of a layout style which guides the generation of a layout structure from a logical structure.

Layout directives 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 § 6 for a specification of the reference document layout process).

Some of the attributes may he 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 directives are applied subject to their conforming to the layout object class descriptions.

5.7.1 *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 this logical object within the available area(s) (see § 2.4.2), subject to satisfying constraints on placement specified by the attribute "offset" (see § 5.7.8). The alignment specified by this attribute is in the direction perpendicular 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^{o}, '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).

A 'null' value indicates that no constraints on the layout of the logical object are specified by this instance of the attribute.

5.7.2 *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' or 'non-concatenated'.

Default value

'non-concatenated'

Definition

This attribute specifies whether or not the content associated with a basic logical object to which it applies and the content associated with an earlier basic logical object in the logical sequential order which has the same content architecture class, layout category and fill order, are to be concatenated. That is, the attribute indicates whether the content associated with the object and the earlier object are to be treated as an unbroken stream. There may be other logical objects between the two in sequential logical order, but these must not specify the same content architecture class, layout category and fill order.

The value 'concatenated' specifies that the layout of the content associated with the component must if possible be continued in the same basic layout object as used with the content associated with the earlier basic logical component in sequential logical order that has the same value of the attributes "content architecture class", "layout category" and "fill order".

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

Each content architecture specifies in the relevant Recommendation of the T.410 Series 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.

5.7.3 *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) containing content associated with this logical object are to be laid out within their immediately superior layout object, relative to the direction of layout path of that superior object.

The value 'normal order' specifies that the blocks are to be positioned after each other in the direction of layout path, in the sequential logical order of the logical objects whose content they contain.

These 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 to be positioned after each other in the direction of layout path, in the sequential logical order of the logical objects whose content they contain.

These 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 § 3.3.3.)

5.7.4 *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 shall if possible be laid out within a single layout object which is of a specified object class or 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 must be of the specified layout object class, this must be of object type page set, page or frame.

If the value is as in b) above, the layout object must be of the specified layout category.

If the value is as in c) above, the layout object must be of object type page.

A 'null' value indicates that no constraints on the layout of the logical object are specified by this instance of this attribute.

5.7.5 *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 string of characters from the minimum subrepertoire of ISO 6937-2, this being a layout category identifier.

Default value

'null'

Definition

This attribute specifies the name of the layout category of the logical object.

A layout category is a name which may be associated with basic logical component descriptions and with frame component descriptions in order to specify and restrict the layout objects into which the content associated with basic logical objects may be placed.

A logical component can specify only a single layout category. The content associated with a logical object is constrained to be placed within frame(s) that include the name of this layout category in the list of such names 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 § 6.3.1), each layout stream pertaining to a particular layout category.

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 matching layout categories for each of the logical objects.

Also, the content associated with basic logical objects which have the same layout category may be laid out in frames generated from different layout object classes, providing that each frame specifies a matching layout category.

If the attribute value is 'null', then the logical object does not specify any layout category and the associated content is only permitted to be laid out into frames for which no permitted categories have been specified, i.e. for which the attribute "permitted categories" has the value 'null'.

5.7.6 *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 must 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.

This attribute may only be used to specify layout object classes of object type document layout root, page set, page or frame.

This attribute can 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 must 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 can 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 only valid 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.

5.7.7 *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 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 or layout category or object type.

The current layout position in the specific layout structure is that of the basic layout object in which was laid out the end of the content associated with the preceding logical object with the same layout category as:

- for a basic logical object, the object for which the attribute is specified;
- for a composite logical object, the first basic logical object in sequential logical order subordinate to the logical object for which this attribute is specified.

In the case that no preceding logical object had the same layout category, the current layout position is the document layout root.

If the value is as in a) above, the next layout object must be of the specified layout object class, this must be of object type page set, page or frame.

If the value is as in b) above, the next layout object must be of the specified layout category.

If the value is as in c) above, the next layout object must be of object type page.

A 'null' value 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.

5.7.8 *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 parameters, "leading offset", "trailing offset", "left-hand offset", "right-hand offset"

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 § 2.4.2) within the immediately superior frame or page in which the content associated with the basic logical object can 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 § 3.3.3.)

For each edge, the corresponding parameter specifies the amount of offset for that edge in scaled measurement units.

5.7.9 *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

The value of this attribute consists of two parameters.

Permissible values

With regard to the first parameter:

- when this attribute is applied to a logical object description, the value is the identifier of another logical object description, or 'null';
- when this attribute is applied to a logical object class description, the value is an object identifier expression, or 'null'.

If the first parameter has the value 'null' then the second parameter is ignored, otherwise the value of the second parameter shall be one of:

a) the identifier of a layout object class description;

b) the identifier of a layout category;

c) 'object type page';

Default value

First parameter: 'null'; second parameter: ignored.

Definition

This attribute specifies that the start of the content associated with the logical object and the end of the content associated with another logical object, specified by the first parameter, shall be laid out, if possible, within a single layout object which is of a specified layout object class or layout category or object type, as specified by the second parameter.

If this cannot be fulfilled, then the start of the content associated with the logical object shall be laid out in the earliest layout object in sequential layout order which:

- follows the layout object in which the end of the content associated with the specified logical object is laid out;
- is of the specified layout object class, layout category or object type;
- permits the layout of the start of the content associated with the logical object.

The current layout position of all layout streams pertaining to the layout categories that apply to the logical object, and its subordinates, if any, for which the attribute applies are moved forward to the beginning of the layout object identified by the second parameter. Any layout stream with current layout position within or after the layout object identified by the second parameter is not affected.

If the value is as in a) above, the layout object must be of the specified layout object class, this must be of object type page set, page or frame.

If the value is as in b) above, the layout object must be of the specified layout category.

If the value is as in c) above, the layout object must be of object type page.

A 'null' value of the first parameter indicates that no constraints on the layout of the content associated with the logical object are specified by this instance of this attribute.

5.7.10 *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 parameters, "leading edge", "trailing edge", "centre separation".

Permissible values

For each parameter, a non-negative integer. One or more of the parameters can be specified in any instance of this attribute.

Default values

Each of the three parameters are independently defaultable. The default value for each parameter is: 0

Definition

This attribute specifies minimum amounts of separative between the block(s) used to lay out the content associated with the basic logical object 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 must 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 fill order 'normal order' and 'reverse order'.

Thus, a constraint on the separation of two adjacent blocks which have different fill orders is that the separation must 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 § 3.3.3.)

5.7.11 *Synchronization*

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

When this attribute is specified for a logical object description, the value is either the identifier of another logical object description, or 'null'.

When this attribute is specified for a logical object class description or in a layout style, the value is either an object identifier expression, or 'null'.

Default value

'null'.

Definition

This attribute specifies that the content associated with the component and with another specified component are to be laid out aligned along a line orthogonal to the direction of the layout path.

When the attribute specifies a reference to another logical object, the trailing edge of the first block containing content associated with that other logical object and the trailing edge of the first block containing content associated with the logical object to which the attribute applies should be synchronized, i.e. the lines along the trailing edges coincide.

The two blocks containing the beginning of the content associated with the two logical objects involved must be placed into distinct lowest level frames. The direction of fill order for these blocks must be the same, and the direction of layout path must be the same in these frames for this attribute to be in effect. If not, this attribute will 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.

5.7.12 *Interactions and precedences among the layout directives*

This sub-clause describes the order in which the different layout directives are to be taken into account in the layout process.

The interactions and precedences described here provide a general implementation guideline, but 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 T.410 Series of Recommendations 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. It is outside the scope of Recommendations of the T.410 Series to indicate how a document containing conflicting or inconsistent information is to be laid out.

For composite logical objects the following layout directive attributes are applicable, listed in the order of decreasing precedence:

- layout object class;
- new layout object;
- same layout object;
- synchronization;
- indivisibility.

This set of layout directives should not only be taken into account at a particular level but also at hierarchically related levels. This means that each layout directive applicable to a lower level logical object description should result in a valid layout as defined by the layout directives applicable to logical object descriptions higher in the hierarchy. That is, a layout directive applicable to logical object descriptions higher in the hierarchy has precedence over any layout directives applicable to logical object descriptions at lower levels.

For basic logical objects the following layout directive attributes are applicable, listed in the order of decreasing precedence:

- layout object class;
- layout category;
- new layout object;
- same layout object;
- fill order;
- concatenation;
- offset;
- separation;
- synchronization;
- indivisibility;

- block alignment.

The layout directives applicable to basic logical object descriptions should also be valid within the set of layout directives applicable to superior logical object descriptions.

There follows a description of the individual layout directives in order of decreasing precedence:

a) *Layout object class*

The specification made by this attribute should be given highest precedence.

When this attribute specifies a lowest level frame, then the following attributes are ignored:

- new layout object;
- same layout object;
- indivisibility.

b) *Layout category*

The specification made by this attribute should be fulfilled.

c) *New layout object*

The specification made by this attribute should 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 should fulfill the specification.

If the attribute "layout object class" has specified a lowest level frame for this or a superior logical object, then this attribute should be ignored.

d) *Same layout object*

The specification made by this attribute should be ignored when:

- 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 to 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.

e) *Fill order*

The specification made by this attribute should be fulfilled.

f) *concatenation*

When this attribute specifies the value 'non-concatenated', the specification made by this attribute should be fulfilled.

When this attribute specifies the value 'concatenated', the specification made by this attribute should be ignored and the content associated with the logical component should 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 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 apply then the content associated with the logical object should be concatenated if there is adequate available area.

The attributes "separation", "offset", "border", "colour", "transparency", and "block alignment", are 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".

g) *Offset*

The specification made by this attribute should be fulfilled except when the attribute is applied to basic logical objects that are concatenated to preceding logical objects.

h) *Separation*

The specification made by this attribute should be fulfilled except when the attribute is applied to basic logical objects that are concatenated to preceding logical objects.

i) *Synchronization*

When this attribute specifies a value other than 'null', the specification should be fulfilled provided that all conditions for synchronization are possible without violating any layout directives of higher precedence.

k) *Indivisibility*

When this attribute specifies a value other than 'null', the specification should be fulfilled provided that all conditions for indivisibility are possible without violating any layout directives of higher precedence.

l) *Block alignment*

The specification made by this attribute should 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".

5.7.13 *Interaction among attributes affecting the layout process*

All layout directives 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" should 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" should be fulfilled without violating any of the layout directives.

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" can 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 effect their position and dimensions.

5.8 Presentation style attributes

5.8.1 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.

5.8.2 *Attributes which can be specified for presentation styles*

The following attribute can be specified for presentation styles:

- presentation style identifier (see § 5.8.1);
- user-readable comments (see § 5.3.5.1);
- user-visible name (see § 5.3.5.3);
- presentation attributes (see § 5.4.4);
- border (see § 5.4.1.3);
- transparency (see § 5.4.3.2);
- colour (see § 5.4.3.3).

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" and "user-visible name" 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" can 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.)

5.9 *Content portion attributes*

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

5.9.1 *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 § 5.3.3.3).

The value of each attribute consists of a sequence of integers, as defined in § 5.3.1.2 and § 5.3.1.3 for an identifier of a basic component, with an additional integer 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 descriptive can 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 Recommendation T.415), 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 CCITT Recommendations.

5.9.2 *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 specified in Recommendations of the T.410 Series that deal with 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 § 5.9.4).

Exception case

The value is also permitted to be an integer in the case of formatted raster graphics content architecture.

Note – This exceptional case is provided for compatibility with CCITT Recommendations.

5.9.3 *Content information attributes*

5.9.3.1 *Content information*

Constituents

Content portion descriptions.

Classification

Non-mandatory.

Permissible values

An octet string.

Representation

Defined in the Recommendations of the T.410 Series that deal with individual content architectures.

Definition

This attribute specifies that part of the content portion description which is composed of content elements for example, graphic characters, pixels) governed by a content architecture.

5.9.3.2 *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 the content portion.

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 6937-2.

In addition to the graphic character set, the control functions carriage return and line feed may be included in the character string.

5.9.4 *Coding attributes*

Constituents

Content portion descriptions.

Classification, structure, permissible values, representation, default values

Defined in the Recommendations of the T.410 Series that deal with individual content architectures.

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.

6 Reference model of the document layout process

This section 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 content layout process, which controls the layout of content portions within basic layout objects is not described here but in those Recommendations of the T.410 Series relating to particular content architectures.

6.1 *Overview*

The document layout process consists of 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 accordance with their sequential order in the specific logical structure.

The reference model of the document layout process only handles layout into frames, it does not handle the case of documents containing basic pages.

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 directives 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 and document layout processes 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 specified with fixed positions or specified with non-fixed positions.

All blocks with the same immediate superior frame are either specified with fixed positions and dimensions, i.e. from the generic layout structure, or with non-fixed positions and dimensions, i.e. content dependent.

It is assumed that a document which is defined in accordance with the T.410 Series of Recommendations 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. It is outside the scope of the T.410 Series of Recommendations to indicate how a document containing conflicting or inconsistent information is to be laid out.

6.2 *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 without the use of a 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 must 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.

6.2.1 *Laying out content of a document*

For the layout process, content of a document can be either related 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.

6.2.1.1 *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 directly referenced by a basic logical object;

b) content in a generic content portion that is associated with a basic logical object class in a generic logical structure or resource-document;

c) content specified by the attribute "content generator" that is specified for the basic logical object;

d) content specified by the attribute "content generator" that is specified for a basic logical object class in a generic logical structure or resource-document.

In all cases, the content may be either in processable, formatted or formatted processable form.

If case a) applies, the document layout process causes 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 and specific layout structures.

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 portion will be divided into two or more content portions such that the basic logical object now references 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 no new layout object needs to be created because 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 the basic logical object derives its content information from content information in a generic object class, as in case b), then the document layout process causes the generation of a new content portion that is only associated with the specific layout structure. That is, this new content portion is 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 a new content portion that is only associated 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.

6.2.1.2 *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 resource-document;

b) content specified by the attribute "content generator" to be applied to a basic object class description in the generic layout structure or resource-document;

c) content specified in the generic logical structure that is referred to from the generic layout structure (application of "logical source");

In case a), the content portion is already formatted and the position and dimensions of the block containing it 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 only have a reference to the generic layout object class description with the generic content portion 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 only associated 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 § 5.4.2.3), 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 order for the layout process to be deterministic the attribute "generator for subordinates" for this logical object class and also for all of its subordinates is only allowed to specify a construction expression composed from construction terms consisting only of required construction factors and/or sequence constructions which use only required construction factors.

The created logical object, or hierarchy of logical objects, is not added to the specific logical structure.

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 a way that the created logical objects are entirely laid out 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 only associated to the specific layout structure.

6.2.1.3 *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 not common to both the specific logical and specific layout structures. All occurrences of the attribute "content identifier - layout" are also removed from the content portions associated with the specific logical structure. In addition, any content portions that were divided as a result of the document layout process are recombined into a single content portion.

The reformatting is then performed as described for initial formatting in § 6.2.1.1 and 6.2.1.2.

6.2.2 *Specific layout structure generation*

The process of creating the specific layout structure is governed by the attribute "generator for subordinates" (see § 5.3.2.1) which is applicable to layout object class descriptions and which guides and restricts the layout sub-structures, if any, that may be generated immediately subordinate to layout objects of that class.

This attribute contains a construction expression which specifies the immediately subordinate objects that may be generated when that layout object class is used and in what order these subordinate objects may be generated. The immediate subordinates of a composite layout object are created in the sequential layout order specified by the construction expression. 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 evaluated as described in § 5.3.2.1.

The result will be a hierarchic specific layout structure with a well defined sequential layout order that can serve as input to an imaging process.

6.3 *Layout references and categories*

There are two principal methods of controlling the generation of a specific layout structure from the logical structure, and the allocation of the content of basic logical objects to layout objects within that layout structure; one is provided by layout categories, the other by layout references.

6.3.1 *Layout categories*

A layout category can be specified by the attribute "layout category" (see § 5.7.5), which applies to basic logical object descriptions, and by the attribute "permitted categories" (see § 5.4.2.4), 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 only placed into frames which specify a matching layout category.

The effect of specifying different layout category identifiers for different basic logical objects is to partition those objects into different *layout streams*, each of which corresponds to a particular layout category. These different layout streams are laid out into sequences of frames having appropriate layout category identifiers. The content of logical objects of the same layout stream is laid out such that the order of the content within the specific layout structure is the same as its order within the specific logical structure.

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 layout category that applies.

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 frame is needed for the content of a particular layout stream, then the layout process searches the layout structure for the earliest lowest level frame in sequential layout order which already exists, or can be created, and includes the layout category of the layout stream in its permitted categories and satisfies other constraints that may be present due to attributes of the logical and layout structures. The search for such a frame starts from the layout object identified by the current layout positions. When a suitable lowest-level frame has been identified, the current layout position of the layout stream is moved forward to this frame.

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 a block within the lowest level frame at the current layout position of the layout stream identified by the layout category of that logical object.

When the content of a logical object is split by the layout process into several layout objects, this above layout of content is repeated, as in the preceding description, for each part of the content, possibly with intervening searches for a new current layout position.

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".

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 directives 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 its beginning. 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 the specific logical structure (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 of the specific logical structure. 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 logical source attribute, respectively, applies. If necessary this content is laid out when the end of the layout process is reached.

The layout process places no constraints on the layout object class descriptions which specify the permitted category, thus the content of basic logical objects may be laid out in frames generated by different layout object class descriptions, provided that each frame specifies the appropriate layout category identifier.

In the case that frames specify multiple permitted categories then the current layout position of more than one layout stream can identify the same lowest level frame, thus in such cases basic logical objects with different layout category identifiers may be laid out in the same frame.

If a lowest level frame specifies no permitted categories ('null' value), then the layout process treats it as if it had specified 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 basic logical objects of any layout category may be laid out in such a frame.

If a basic logical object has no specified layout category, then it is allocated to a separate layout stream maintained for this purpose, the layout process will lay it out only in lowest level frames which specify no permitted categories.

The use of different layout streams is illustrated in Figure 13/T.412.

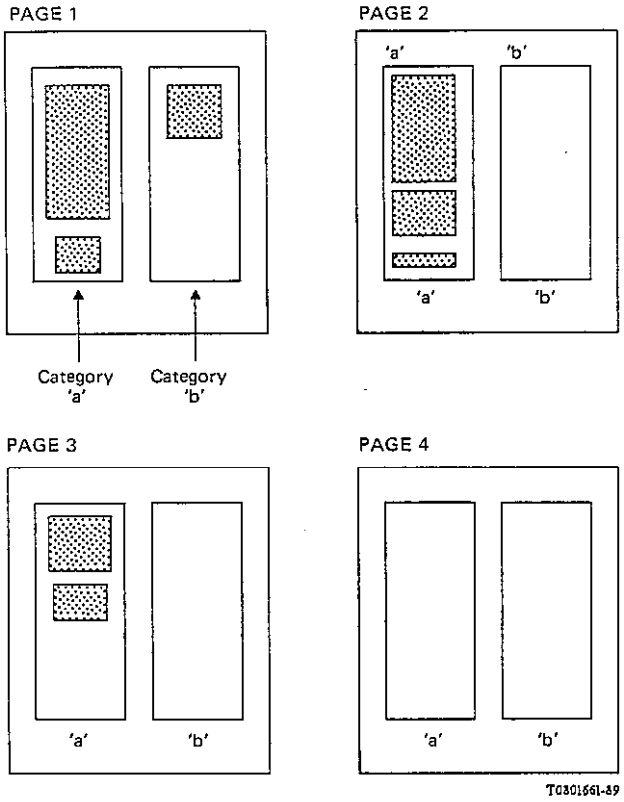


FIGURE 13/T.412

Illustration of layout streams

In this example, each page contains frames of permitted category 'a' and 'b'. If the next basic logical object has category '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 category '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 category 'b'.

Thus frame 'b' in page 1 is the current layout position for laying out the content of the next basic logical object of category 'b'; and frame 'a' in page 3 is the current layout position for basic logical objects of category 'a'.

In subsequent sub-sections it is shown how some additional attributes impose additional constraints concerning the frame in which a basic logical object is laid out.

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 should be satisfied when laying out a basic logical object into a 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", should be created by the layout process to contain all the content associated with all the subordinates of the composite logical object. Layout categories applicable to subordinate basic logical objects are only valid within the layout sub-structure subordinate to this layout object.

Both a layout object class and a layout category may be specified for a basic logical object and in this case both attributes must be satisfied when carrying out the layout process.

6.3.2 *Layout object class*

Layout reference is specified by the layout directive attribute "layout object class" (see § 5.7.6) which identifies a layout object class description. When processing a logical object description specifying this attribute the layout process creates a layout object of the specified layout object class.

The content of the logical object, or if it is a composite object the content of all of its subordinate basic objects, is entirely laid out within this single layout object. No other part of the document is laid out within this layout object.

Thus, 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 must be laid out in an instance of a particular page set.

6.4 *The effect of some attributes on the document layout process*

In addition to the constraints imposed by the attributes "layout object class" and "layout category" on the document layout process, a number of other attributes impose additional constraints on this process. This sub-section describes the effect of some of these, namely:

- new layout object;
- same layout object;
- indivisibility;
- synchronization;
- balance.

6.4.1 *New layout object*

The attribute "new layout object" (see § 5.7.7) is used to control the layout process such that the content associated with the logical object with which the attribute is associated, should be laid out at the beginning of a particular instance of a layout object. It specifies that the content of the first affected layout stream should be laid out in a layout object, which does not contain any preceding content. In addition it also synchronizes a set of layout streams.

Examples of use are that a section should start on a new page; a figure at the beginning of a frame, or a particular paragraph at the top of a column.

The layout process determines the current layout position of the first affected layout stream. This stream is the first encountered layout stream in the sequential logical order 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, 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 position is found to be in a layout object fulfilling these criteria, no new layout object of specified layout object class, layout category or object type need to be created. Otherwise the layout process has to 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 position for all layout streams specified by 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 of these layout streams belonging to the logical object or its subordinates are laid out.

In acting on an instance of this attribute, the layout process only finds or creates one layout object of the specified layout object class, layout category or object type and this only for the first affected layout stream.

As an illustration, consider in Figure 12/T.412 the case in which the first affected layout stream is of layout category 'b' and the attribute "new layout object" specifies a layout object of layout category '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.

6.4.2 *Same layout object*

The attribute "same layout object" is used to control the layout process such that the content associated with the logical object with which the attribute is associated and the content associated with another specified logical object, is laid out within a particular layout object. It specifies that the beginning of the content of the logical object for which the attribute applies should be laid out in the same layout object of a specified object class, layout category or object type as the end of the content of a referenced logical object. In addition it also synchronizes a set of layout streams.

A typical use of this attribute is to control the layout process such that a footnote should start on the same page as its reference.

The layout process requires that the logical object referred to shall precede, in the sequential logical order, the logical object for which the attribute applies.

The synchronization aspect of this attribute moves the current layout positions of all layout streams of the logical object for which the attribute applies, forward in the sequential layout order to the layout object specified by the attribute. These current layout positions are moved before any content of these layout streams belonging to the logical object or its subordinates are laid out. Any layout stream with current layout position within or after the specified layout object is not affected.

The layout process determines the layout position of the end of the content of the referenced logical object and the current layout position of the content to be laid out. If these two layout positions are both within the same layout object of specified object class, layout category or object type, the effect of this attribute is fulfilled.

If this should not be 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 so as to satisfy the constraint specified by this attribute. However, for each layout stream (layout category), the sequential layout order must not be changed, i.e. it must correspond to the sequential logical order.

6.4.3 *Indivisibility*

The attribute "indivisibility" is used to control the layout process such that the content associated with the logical object with which the attribute is associated, is laid out within a particular layout object. It specifies that all content associated with the logical object for which the attribute applies should be laid out in the same layout object of a specified object class, layout category or object type.

Typical uses of this attribute are to control the layout process such that a figure and the whole figure caption should be laid out on the same page; that a certain paragraph must not be split by a column break or page break; or that two paragraphs should be laid out completely on the same page.

The layout process determines if the current layout positions of all affected layout streams are within the same layout object of specified object class, layout category or object type, and that all content of the affected layout streams that belongs to the logical object (if basic) or to the subordinates of the logical object (if composite), for which the attribute applies can be laid out completely within that layout object. If this is the case, then this attribute is fulfilled.

If this should not be 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 (layout category), the sequential layout order must not be changed, i.e. it must correspond to the sequential logical order.

If the current layout positions of all affected layout streams are not within a layout object for which the attribute can be fulfilled, this attribute moves the current layout positions of all affected layout streams of the logical object for which the attribute applies, forward in the sequential layout order to the layout object specified by the attribute. These current layout positions are moved before any content of these layout streams belonging to the logical object or its subordinates are laid out.

6.4.4 *Synchronization*

The attribute "synchronization" is used to control the layout process such that the logical object with which the attribute is associated, and another specified logical object have their content laid out in different blocks with trailing edges aligned along a line. It specifies that the block containing the beginning of the content of the logical object or, if composite, its subordinates for which the attribute applies should be laid out with its trailing edge along the line of the trailing edge of the block containing the beginning of the content of the logical object or, if composite, its subordinates of the logical object referred to in the attribute.

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, namely:

- The referenced 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 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, however, the reference layout process does not identify any particular algorithms by which the layout may be modified. However, for each layout stream/layout category the sequential layout order must not be changed, i.e. corresponds 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:

- either move the block containing the beginning of the content of the logical object, if basic, or of its first subordinate, if composite, for which the attribute applies such that its trailing edge is aligned with the trailing edge of the block containing the beginning of the content of the logical object, if basic, or of its first subordinates, if composite, referred to;
- or, alternatively it will move the block or blocks containing the content of the referred logical object or its subordinates forward in the direction of layout path such that the attribute can be fulfilled.

If the content of either 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 both the attribute "synchronization" and the attribute "fill order" with value 'reverse order' have been applied to a logical object, the reference layout process outlined does not specify an exact algorithm.

6.4.5 *Balance*

The attribute "balance" is used to control the layout process such that the content in a specified set of layout objects contains appropriate amount of content such that the leading edge of each layout object specified should be approximately along a single line.

A typical use of this attribute is for balancing multi-column text at the end of a section.

The attribute only applies to those object or object class descriptions in which it is specified if these correspond to objects which only have composite layout objects as its immediate subordinates. Thus the attribute does not apply when specified in a component description of a lowest level frame.

The layout process determines if all conditions for this attribute are fulfilled (see § 5.4.2.1). If this is not the case, the attribute will be ignored.

In the case that 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 (layout category), the sequential layout order must not be changed, i.e. corresponds to the sequential logical order. Normally, the layout process will move content from the layout objects earlier in the sequential layout order to those later in the specified set in order to "balance" the content so that their leading edges will be approximately aligned.

6.5 *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 positions.

a) a constant position;

b) a position derived from evaluation of a rule.

Frames immediately subordinate to a page are 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 § 6.6. The two cases for layout of frames within a superior frame are described in this sub-section.

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 of their immediate superior.

In the case that the attribute "border" is specified, then for each edge the border allowance is the sum of the "border line width" and "border freespace width" for the frame. If a border is not specified for one or more of the edges, then for these edges the border allowance is 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 immediately subordinate frames.

6.5.1 *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”, “vertical position”. If either of the sub-parameters “horizontal position”, “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.

6.5.2 *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 "alignment", "fill order", "offset" and "separation". These sub-parameters specify constraints on the placement of the frame which are used to uniquely determine the position of the frame.

In this case the layout of frames is also controlled by means of the layout attribute "layout path" (see § 5.4.2.2) 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 assumes that placement of the variably positioned immediate subordinates of a frame occurs in their sequential layout order.

6.5.2.1 *Determination of the area for placement of frames*

The area within a frame 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.

In all cases the area 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 greater of the border allowance and the offset for the edge, specified by the relevant one of the sub-sub-parameters "trailing offset", "leading offset", "left offset" and "right offset" determines the constraint on the area for placement within the immediately superior frame.

Two cases must be taken into account, as described in § 6.5.2.1.1 and 6.5.2.1.2.

6.5.2.1.1 *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 additionally constrain the distance between the leading edge of the last laid out frame and 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 "center separation" of the sub-parameter "separation" is used to additionally constrain the distance between 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 § 6.5.2.1.

6.5.2.1.2 *Placement of frames in reverse order*

For placement of frames in reverse order, the reference model for the layout process 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 subordinates.

b) For the purpose of determining the constraints on its dimension, the frame is temporarily positioned as close to the trailing edge of the immediately superior frame as is possible under these constraints and those defined in § 6.5.2.1.

c) Finally, 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 step a) (determination of additional constraints) are:

- If there is no other frame currently within the immediately superior frame; then no additional constraints to those described in § 6.5.2.1 (i.e. border and offset) need to be considered.
- 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 area available for placement. That distance is constrained to be no less than the maximun 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, 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 attribute "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.
- 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.

6.5.2.2 *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 this frame has to be evaluated.

The constraints on 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 has to be evaluated.

The constraints on 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.

6.5.2.3 *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 "fixed dimension" parameter of the dimension attribute, the frame is aligned according to the sub-parameter "alignment" of the "variable position" parameter within the area available subject to the constraints specified by the sub-sub-parameters "right-hand offset" and "left-hand offset" of the sub-parameter "offset".

6.6 *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 §§ 5.4.2.2, 5.7.3, 5.7.8, 5.7.10, 5.4.1.4 and 5.7.2, 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.

6.6.1 *Determination of the available area*

The available area within an immediately superior 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 immediately superior frame is specified to have variable dimensions with a size determined by the content (that is, by the use of the sub-parameters "rule A" or "rule B", see § 5.4.1.2), 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 maximized 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 § 5.4.1.2).

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 from the attribute "border" applicable to the block. If for one or more of the edges a border is neither specified for the frame nor for the block, then for these edges the border allowance is zero scaled measurement units.

In all cases the available area is constrained to be both within the border allowance and also 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 attributes "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.

6.6.1.1 *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 needs to be considered.

If there exists one or more blocks laid out in normal order but none in reverse order, then the attribute "separation" is used to further constrain 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 "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 is used as described in § 6.6.1.

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 further constrain 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 distances of the available area from the immediately superior page or frame, and from the other block(s) laid out in normal order (if any) are further constrained as described above.

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.

6.6.1.2 *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 needs 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 further constrain 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 "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 is used as described in § 6.6.1.

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 is used to further constrain 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 is used as described in § 6.6.1.

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 § 6.6.1.

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.

6.6.1.3 *Block alignment orthogonal to 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.

6.7 *Alternative representation*

If an alternative representation is specified then the layout process does not define the situation when this representation will be used. This is the subject of particular implementations.

If the alternative representation is used then all the layout directives 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 Recommendation T.416. The character set is specified by the document profile attribute "alternative representation character sets".

7 Reference model of the document imaging process

This section 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 § 2.3.11). 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 this section is to aid the understanding of the semantics of the attributes affecting the presentation of the document structure but it does not 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 but in those Recommendations of the T.410 Series relating to particular content architectures.

7.1 *Imaging order*

The *imaging order* determines the precedence of layout objects for imaging in the layout object to which they are immediately subordinate. Thus, this order determines how the image of the document is resolved for displaying on the presentation surface.

The imaging order of layout objects that are immediately subordinate to the same layout object is determined by the attribute "imaging order" of that common superior object. The imaging order of layout objects that are not immediately subordinate to a common layout object is determined by the imaging order specified by their lowest common superior layout object.

When the attribute "imaging order" 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.

7.2 *Intersection principles*

Within a page, frames and blocks may be positioned in such a way that they *intersect* partially or fully, that is they share common areas. In all cases, subordinates are fully contained within their superiors (see § 3.3.1).

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 T.410 Series of Recommendations has the following restrictions:

- "colour" is either 'colourless' or 'white';
- "transparency" is either 'transparent' or 'opaque';
- the texture of pages, frames and blocks is either:
  - 'white', 'opaque';
  - 'colourless', 'opaque';
  - 'colourless', 'transparent'.

For pages, 'colourless', 'opaque' and 'colourless', 'transparent' are equivalent.

'colourless', 'opaque' is intended for hard copy, it allows the colour of page, frame and block areas to be that of the media.

When frames or blocks intersect their intersection is governed by the following rules:

- Their overlay sequence is given by the imaging order, which is the same as the sequential layout order unless an imaging order is explicitly specified. 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.
- For imaging purposes the border 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.

7.3 *General rules for positioning pages on presentation surfaces*

This paragraph is concerned with the rules for positioning pages on presentation surfaces.

7.3.1 *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 sender of a document, is a rectangular area called the *nominal page.*

Thus the 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 must 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.

7.3.2 *Positioning of the page*

The position of the 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 14/T.412. Horizontal positions are measured positively from the vertical axis to the right and vertical positions are measured positively from the horizontal axis downwards.

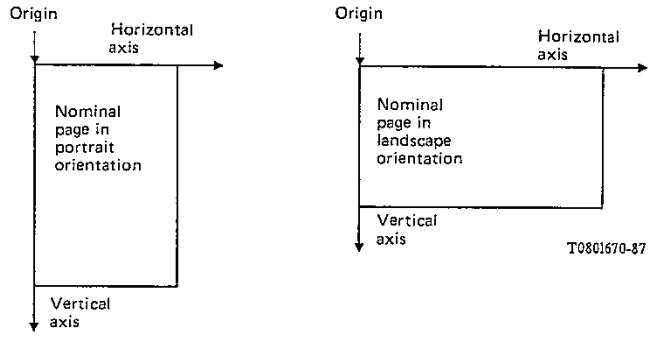


FIGURE 14/T.412

Nominal page coordinate 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 14/T.412). 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 the portrait orientation, the graphic elements within the page may be rotated so that the nominal page is intended to be viewed in the landscape orientation.

The reference point for the positioning of a 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 page position is specified, then the position of the page relative to the nominal page is completely determined.

If the 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 is 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 should be positioned on the nominal page such that the possibility of information loss is minimised.

- If the page dimensions are equal to those of the nominal page, then the page is 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 should be positioned relative to the nominal page such that the possibility of information loss is minimised.

In all four cases, it is intended that the page is positioned such that its edges are parallel to the edges of the assured reproduction area.

Although the provisions of this Recommendation would, in principle, permit text to be positioned anywhere on the nominal page, the originator should rely on text reproduction by the recipient only within the assured reproduction area.

7.3.3 *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 15/T.412, 16/T.412, 17/T.412, 18/T.412 and 19/T.412, respectively.

The dimensions in these Figures are expressed in basic measurement units (BMUs), (see § 3.3.4.1).

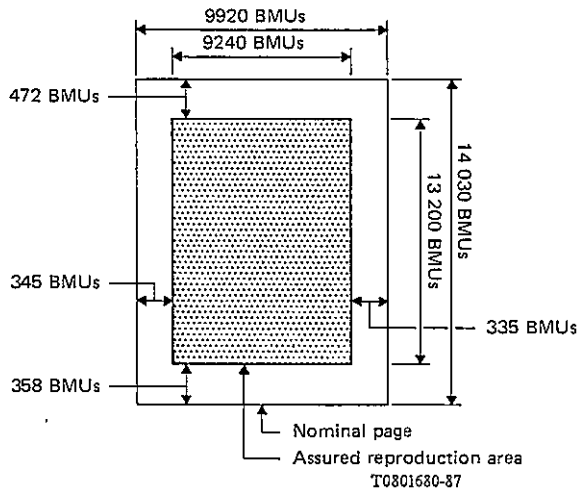


FIGURE 15/T.412

Dimensions and assured reproduction areas for ISO A4 page size

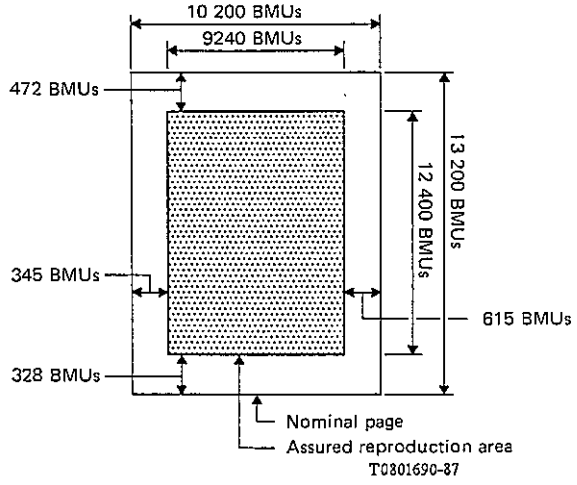


FIGURE 16/T.412

Dimensions and assured reproduction areas for North American Letter page size

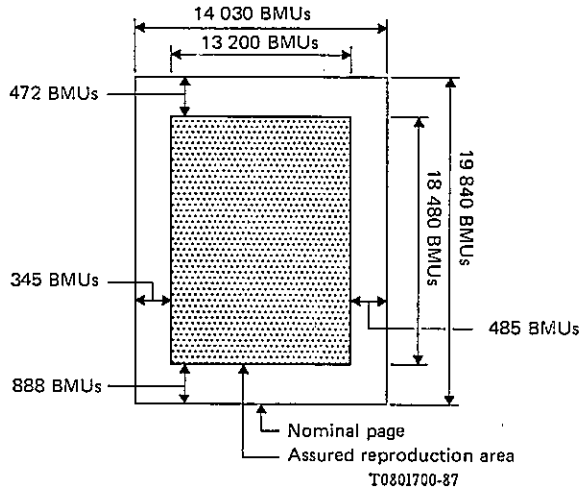


FIGURE 17/T.412

Dimensions and assured reproduction areas for ISO A3 page size

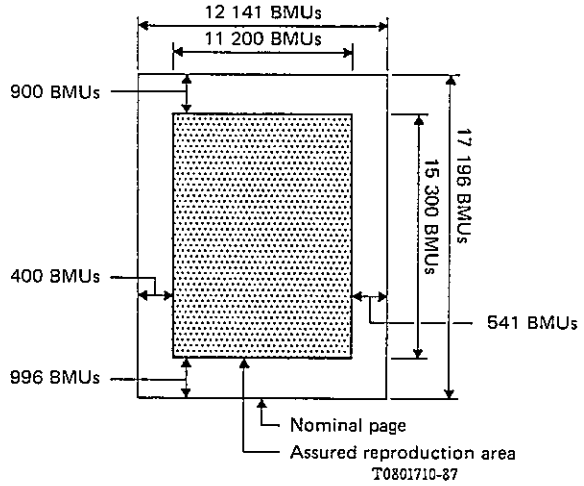


FIGURE 18/T.412

Dimensions and assured reproduction areas for Japanese legal page size

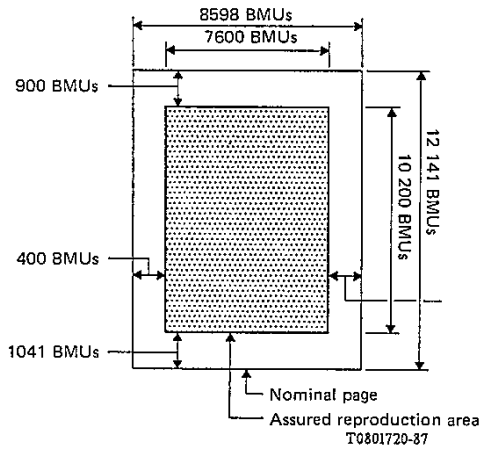


FIGURE 19/T.412

Dimensions and assured reproduction areas for Japanese letter page size

7.3.4 *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.

7.3.5 *Positioning of pages on soft copy media*

For positioning of pages on soft-copy media, the specific characteristics of the soft-copy devices 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 is not defined in the T.410 Series of Recommendations.

8 Document architecture levels

8.1 *Definition of document architecture classes*

This sub-section defines the document architecture classes that may be used in specific applications using the T.410 Series of Recommendations.

Three document architecture classes are distinguished, namely:

- formatted document architecture class;
- processable document architecture class;
- formatted processable document architecture class.

Each class defines:

- the structures that must, or may optionally, be used in documents that pertain to that class;

- the styles permitted in documents that pertain to that 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 the permitted structures that can be used in documents that pertain to that class. However, a document application profile may place restrictions concerning which object types can be used in that application.

For each object type, there is a minimum set of attributes that must be supported by all document application profiles allowing the use of components of that object type. These minimum sets are defined in § 8.3. The document application profile specifies which other attributes may be used for that application.

Recommendation T.411 contains a complete definition of the rules for defining document architecture levels in document application profiles.

8.2 *The constituents of document architecture classes*

The structures that must and that may optionally be present in documents that pertain to the three document classes are defined in Table 2/T.412.

TABLE 2/T.412

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)	Must be present	Not permitted	Optional
PDA	Optional	Must be present	Optional (complete only)	Not permitted	Optional	Optional
FPDA	Optional	Must be present	Must be present (complete only)	Must 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 must be present and others are optional. A structure that must be present shall be present in any document that pertains to that class. An optional structure may or may 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 may or may not be used in documents conforming to that document application profile.

The structures which are present in any particular document, and whether generic structures are partial or complete is specified by attributes of the document profile.

In addition, processable and formatted processable document architecture classes may permit the use of layout styles. Presentation styles may be permitted in any of the three document architecture classes.

8.3 *Minimum attribute sets*

8.3.1 *The minimum sets of attributes applicable to logical component descriptions*

The minimum set of attributes applicable to logical objects is:

- Object type
- Object identifier
- Object class
- Subordinates
- Content portions
- Content architecture class or Content type

The minimum set of attributes applicable to logical object classes is:

- Object type
- Object class identifier
- Content portions
- Content architecture class or Content type

In any document, one or other, but not both, of the attributes "content type" and "content architecture class" should be included in the minimum set of attributes applicable to basic logical components.

8.3.2 *The minimum sets of attributes applicable to layout document descriptions*

The minimum set of attributes applicable to layout objects is:

- Object type
- Object identifier
- Object class
- Content portions
- Content architecture class or Content type
- Position
- Dimensions

The minimum set of attributes applicable to layout object classes is:

- Object type
- Object class identifier
- Content portions
- Content architecture class or Content type
- Position
- Dimensions

In any document, one or other, but not both, of the attributes "content type" and "content architecture class" should be included in the minimum set of attributes applicable to basic layout components.

The attributes "layout path", "logical source" and "permitted categories" are not to be used in documents conforming to formatted document architecture class.

ANNEX A

(to Recommendation T.412)

(Normative)

Notation used to represent document structures

A.1 *Notation for structure diagrams*

The notation described below is intended to be an aid for 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: this indicates a sequence construction, that is the immediately subordinate objects are to be generated in order from left to right as written;

- AGG: this indicates an aggregate construction, that is the immediately subordinate objects can be generated in any order;

- CHO: this indicates a choice construction, that is only one can be chosen to form the immediately subordinate object.

In addition, one or two mnemonic symbols can 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 0 or 1 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 1 or more times.

- OPT REP: the use of the symbols together indicates that an object, or group of objects, can occur 0, 1 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 can 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/T.412 is intended to be added in the main structure in Figure A-1/T.412 at the point indicated by CONNECTOR Z.

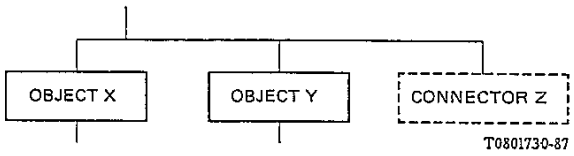


FIGURE A-1/T.412

Example of document structure notation: main structure

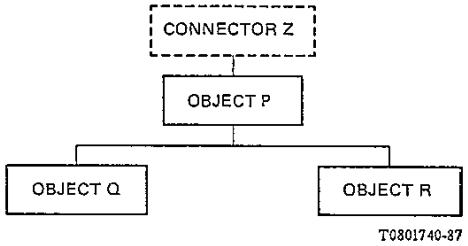
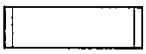


FIGURE A-2/T.412

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 generic content portions.

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 is not usually indicated on the diagrams since this should be obvious in the majority of cases.

A.2 *Notation for expressions*

This paragraph contains 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 paragraph includes notations for:

- construction expressions;
- string expressions;
- numeric expressions;
- object identifier expressions;
- bindings;
- references to binding values.

There are two aspects concerning the definition of this notation. First 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 side is to be substituted for the non-terminal symbol on the left;

b) | the alternative operator, used to separate alternatives;

c) < > used to delimit a non-terminal symbol in an expression;

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) ... a 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 sub-paragraph. 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 § 5.3.2.1). This paragraph defines a human readable notation for construction expressions.

The terminal symbols used are as follow:

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 type>

<object class identifier> ::= -- *any character string from the set of characters:*
hyphen -;
the capital letters, A, B .... Z;
the small letter a, b .... z;
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 § 5.3.1.3).

A.2.3 *Notation for string expressions*

This paragraph defines a human readable notation for string expressions, as defined in § 5.1.3.1.

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 must 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 paragraph defines a human readable notation for numeric expressions (see § 5.1.3.2).

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 letter a, b, ... z;
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 § 5.3.1.2).

A.2.5 *Notation for object identifier expressions*

This paragraph 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> ::= -- *any character string from*
the set of characters:
hyphen -;
the capital letters A, B, ... Z;
the small letter a, b, ... z:
digits 0 9

<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> ::= -- *any character string from*
the set of characters:
hyphen -;
the capital letters A, B, ... Z;
the small letter a, b, ... z:
digits 0 9

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 is according to the format defined in §§ 5.3.1.2 and 5.3.1.3, 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 § 5.3.5.4).

This paragraph 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 ISO 6937-2, 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> ::= -- *any character string from the set of characters: hyphen -; the capital letters A, B, ... Z; the small letter a, b, ... z; 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 § 5.3.1.2).

A.2.7 *Notation for references to binding values*

This paragraph 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 object identifier expressions 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> ::= -- *any character string from the minimum subrepertoire of ISO 6937-2, being the value of the binding name parameter.*

<object identifier> ::= -- *any character string from the set of characters: hyphen -; the capital letters A, B, ... Z; the small letter a, b, ... z; digits 0 .... 9*

<binding selection function> ::= <current-object function>
| <current-instance functions>
| <superior function>
| <preceding function>

<current-object function> ::= CURRENT-OBJECT
| CURR-OBJ

<current-instance function> ::= -- *see § A.2.6.*

<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 § 5.3.1.2).

ANNEX B

(to Recommendation T.412)

(Informative)

Examples of document structures

B.1 *Introduction*

This Annex presents examples of the application of the document architecture specified in this Recommendation 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 § 6 and to content layout processes which are described in those Recommendations of the T.410 Series 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 and B.4.2) describe how the specimen document can 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 can be used to provide two examples of document classes. In the first of these examples, the generic logical structure alone can be regarded as the document class from which the specific logical structure, previously described in the second example, can 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 sub-section 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 § 5.3.1.3. 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 § 5.3.1.2.

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 the number of control functions as specified by the letter is 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/T.412 to B-3/T.412 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/T.412 to B-6/T.412 illustrate the layout structure of this document by outlining various blocks within each page.

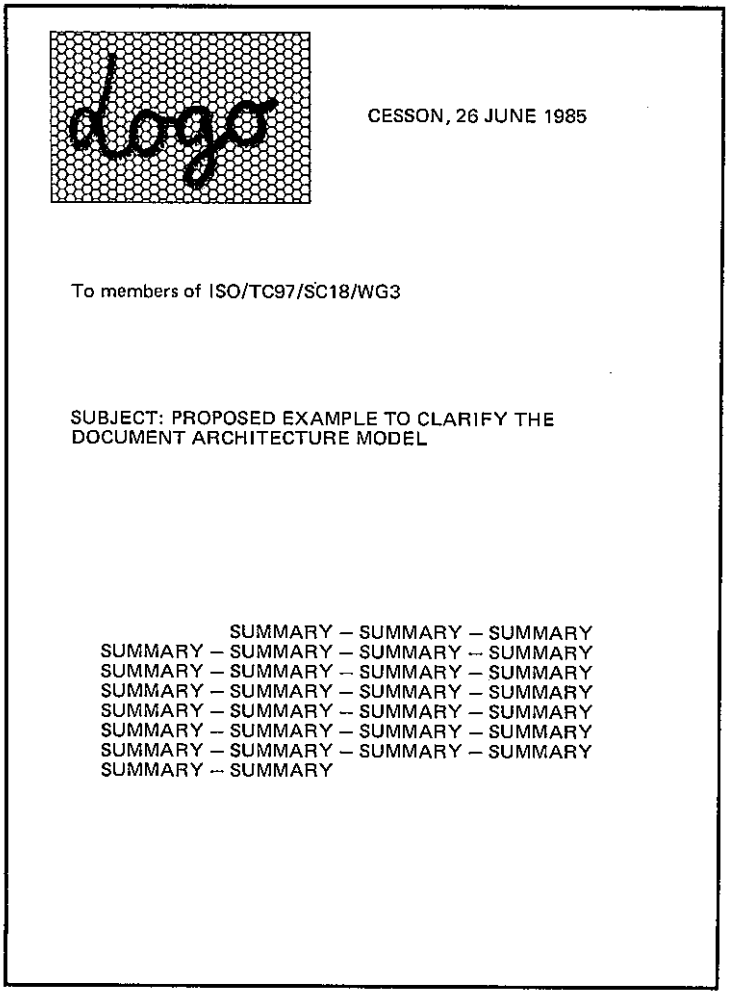


FIGURE B-1/T.412

Specimen document "letter" (1)

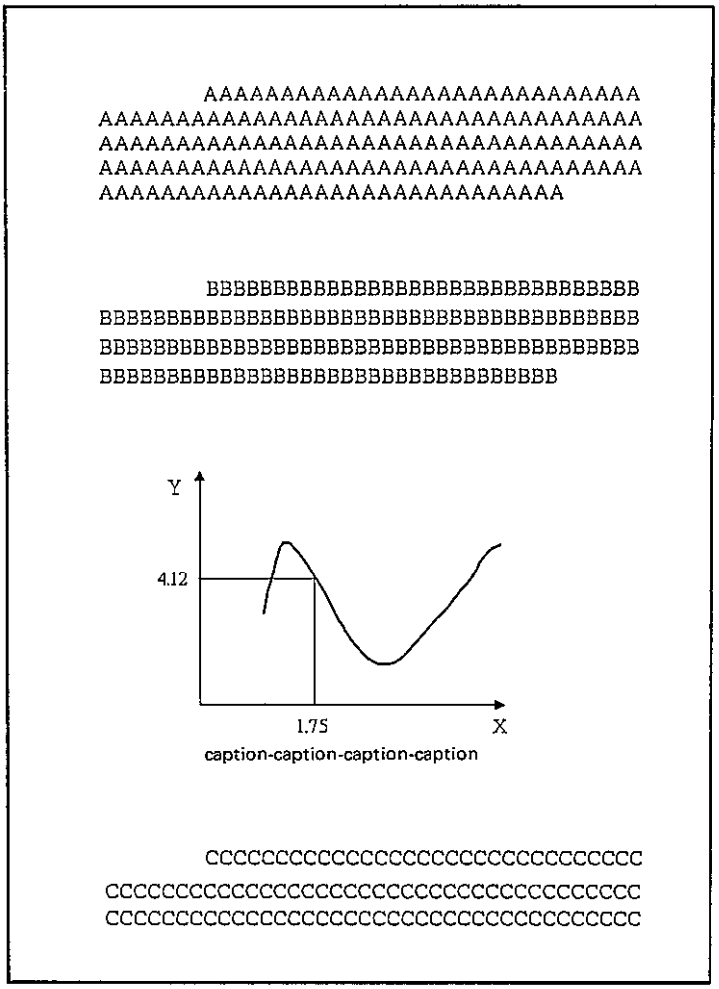
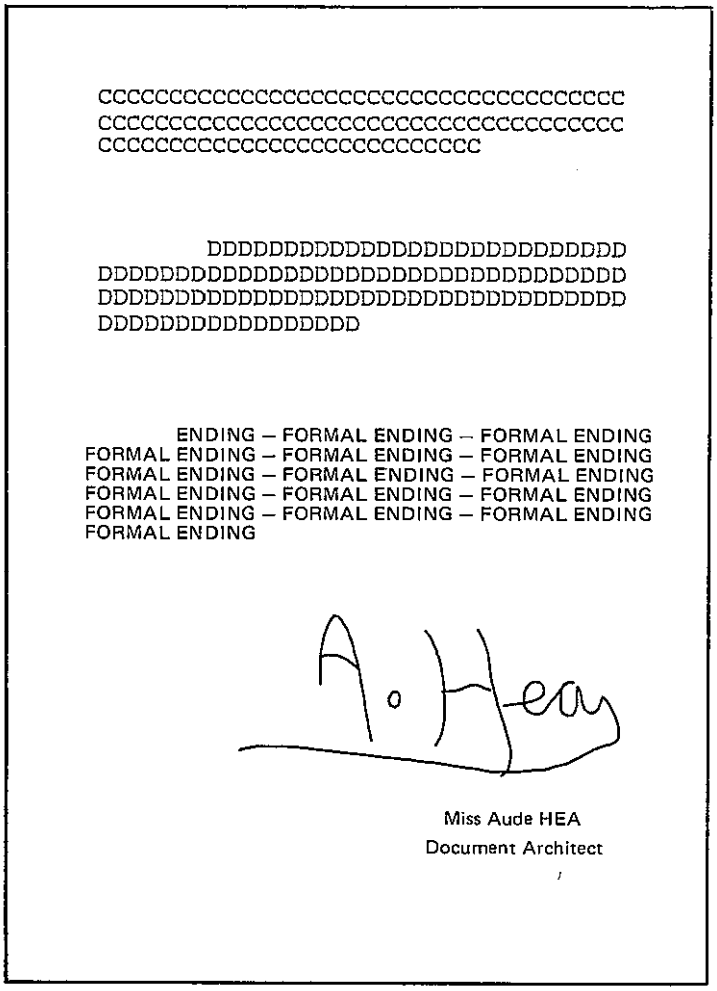


FIGURE B-2/T.412

Specimen document "letter" (2)



T0801770-87

FIGURE B-3/T.412

Specimen document "letter" (3)

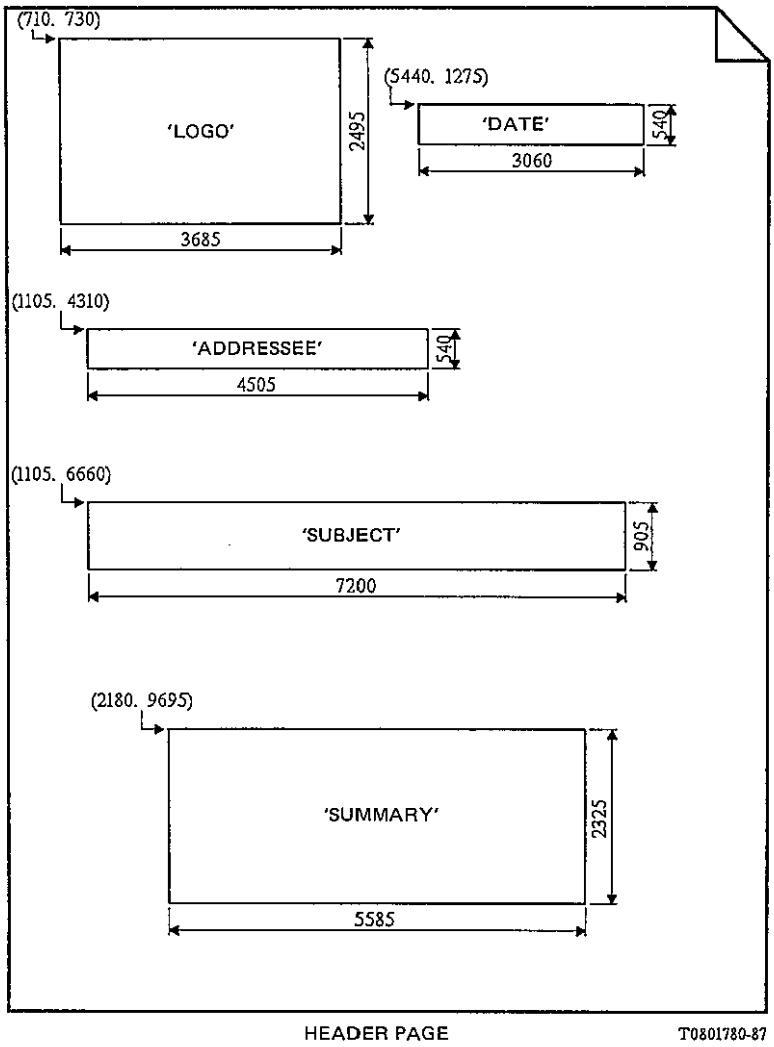


FIGURE B-4/T.412

Layout of "letter" showing pages and blocks (1)

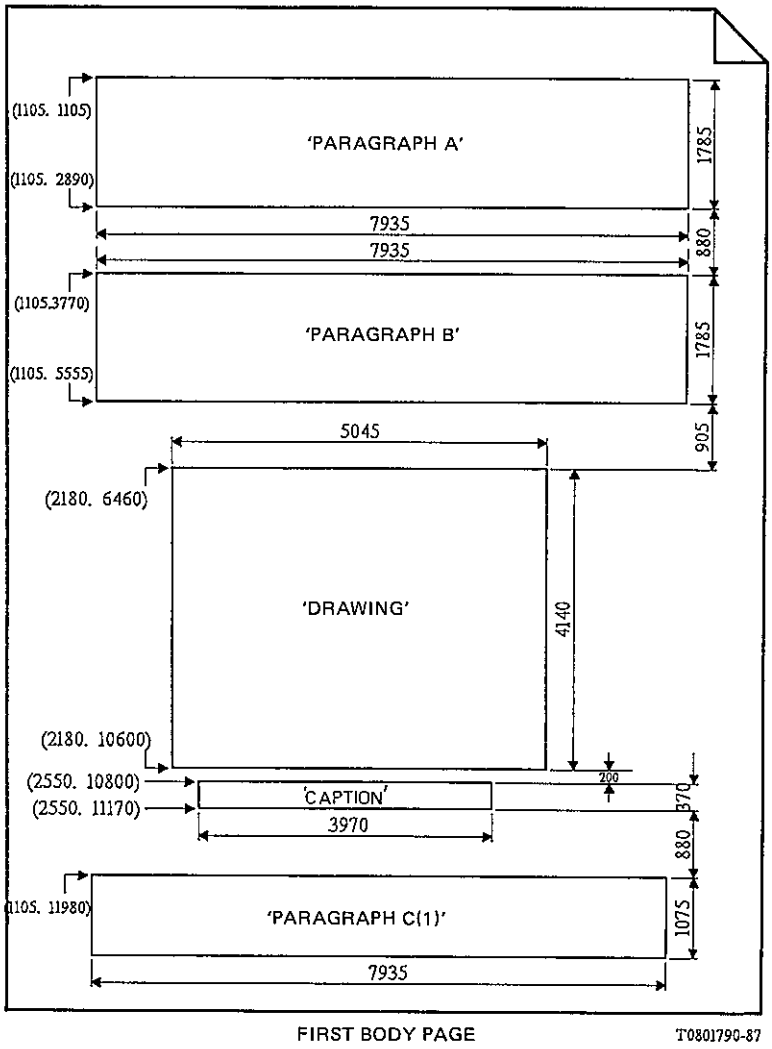


FIGURE B-5/T.412

Layout of "letter" showing pages and blocks (2)

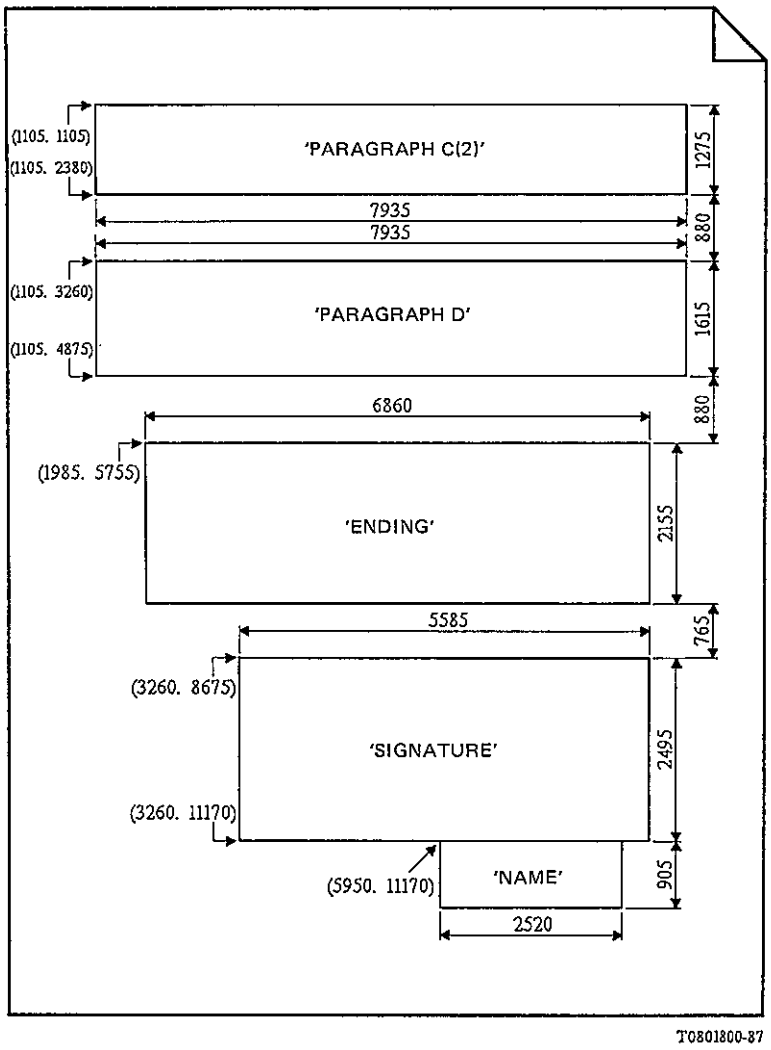


FIGURE B-6/T.412

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 for documents with specific layout structure only*

The specimen document has a specific layout structure that can be illustrated by a diagram as in Figure B-7/T.412.

The diagram contains the same pages and blocks as Figures B-4/T.412 to B-6/T.412 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/T.412 (in double lined boxes).

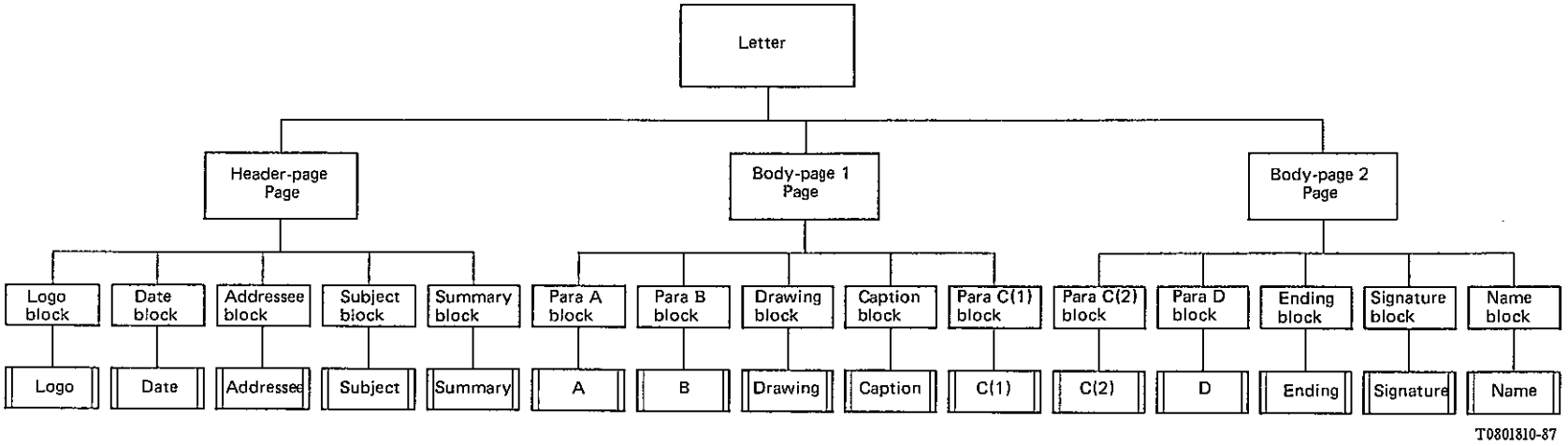


FIGURE B-7/T.412

Specific layout structure (showing pages and blocks)

The document contains one paragraph - paragraph C - which is contained in two blocks; each block on a separate page.

Each layout object, i.e. page or block, is characterised by a number of attributes. Table B-1/T.412 contains a list of all attributes that have to be specified in this example.

Some attributes that are "defaultable" are not shown in the table. Their values can be derived from the standard default values defined in this Recommendation (see § 5.1.2.4).

TABLE B-1/T.412

Constituents for Figure B-7/T.412
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 0
Content identifier - Layout Content information	1 0 2 0 "To members of ISO/TC97/SC18/WG3"

TABLE B-1/T.412 *(Cont.)*

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-*/
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

TABLE B-1/T.412 *(Cont.)*

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*/
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*/

TABLE B-1/T.412 *(Cont.)*

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" HPz = 5950, VP = 11170 HD = 2520, VD = 905 300 0
Content identifier - Layout Content information	1 2 4 0 "Miss Aude HEA\nDocument Architect"

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/T.412.

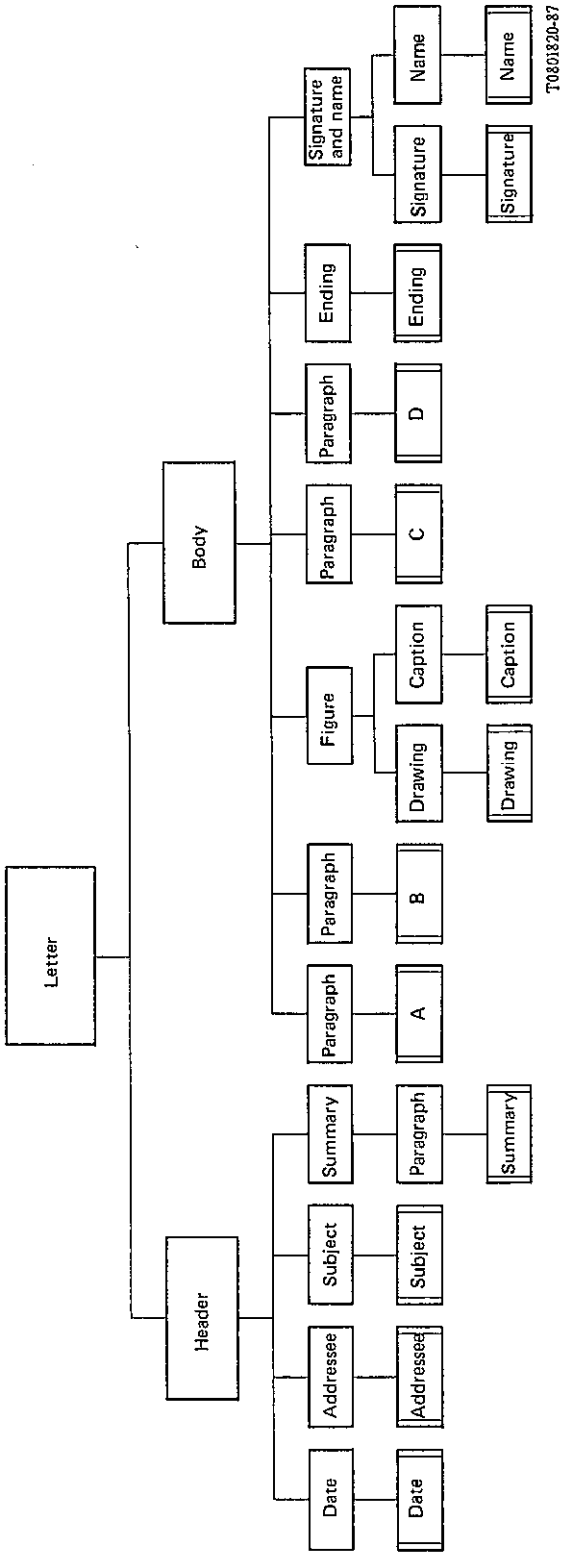


FIGURE B-8/T.412

Specific logical structure

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 can 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/T.412 and B-3/T.412 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.

TABLE B-2/T.412

Constituents for Figure B-8/T.412
Specific logical structure

Object type Object identifier User-visible name Subordinates Default value lists	DOCUMENT LOGICAL ROOT 3 "Letter" 0, 1 list for basic logical objects: attribute: content architecture class value: 'processable character'
Object type Object identifier User-visible name Subordinates	COMPOSITE LOGICAL 3 0 "Header" 0, 1, 2, 3
Object type Object identifier User-visible name Content portions	BASIC LOGICAL 3 0 0 "Date" 0
Content identifier - Logical Content information	3 0 0 0 "CESSON 26 JUNE 1985"
Object type Object identifier User-visible name Content portions	BASIC LOGICAL 3 0 1 "Addressee" 0
Content identifier - Logical Content information	3 0 1 0 "To members of ISO/TC97/SC18/WG3"
Object type Object identifier User-visible name Presentation style Content portions	BASIC LOGICAL 3 0 2 "Subject" 5 0 0
Content identifier - Logical Content information	3 0 2 0 "SUBJECT: PROPOSED EXAMPLE TO CLARIFY THE DOCUMENT ARCHITECTURE MODEL"
Object type Object identifier User-visible name Subordinates	COMPOSITE LOGICAL 3 0 3 "Summary" 0

TABLE B-2/T.412 *(Cont.)*

Object type Object identifier User-visible name Presentation style Content portions	BASIC LOGICAL 3 0 3 0 "Summary paragraph" 5 1 0
Content identifier - Logical Content information	3 0 3 0 0 /*Unformatted string of SUMMARY-*/
Object type Object identifier User-visible name Subordinates	COMPOSITE LOGICAL 3 1 "Body" 0, 1, 2, 3, 4, 5, 6
Object type Object identifier User-visible name Presentation style Content portions	BASIC LOGICAL 3 1 0 "Paragraph A" 5 2 0
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

TABLE B-2/T.412 *(Cont.)*

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*/
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/T.412

Presentation styles for Figure B-8/T.412

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

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/T.412. 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 T.410 Series of Recommendations to define such a layout process or the recipient may specify his own rules for deriving the document layout structure.

Hence it should be noted that 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 should 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.

B.5 *Processable form document with generic logical structure and generic layout structure*

B.5.1 *Introduction*

This sub-section illustrates how the specimen document can 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 can be used during the revision process to ensure that the document remains in conformance with a predefined structure, i.e. a document class.

The generic logical structure illustrated in this example (see Figure B-9/T.412) is one which may have been used to generate the specific logical structure previously shown in Figure B-8/T.412.

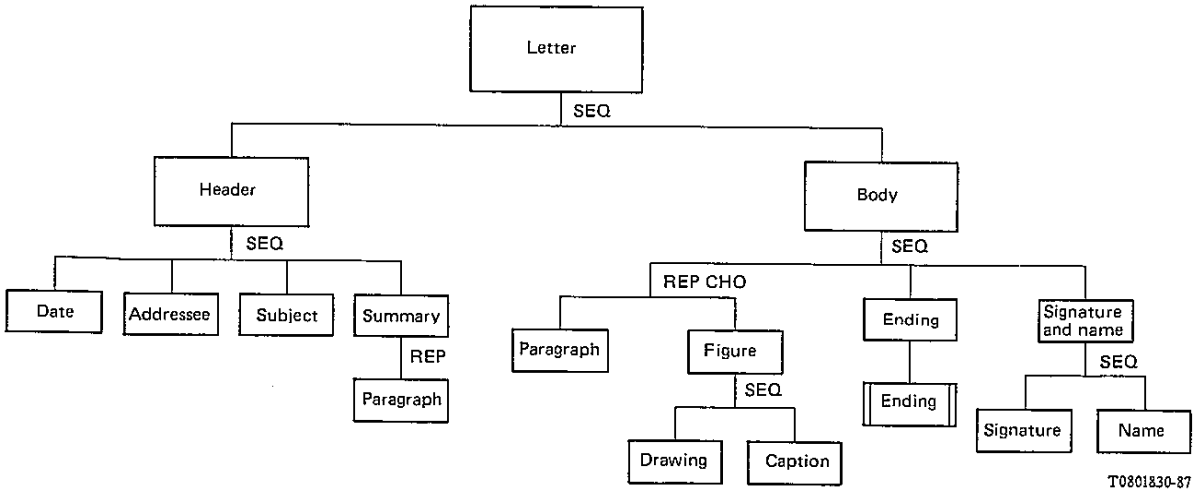


FIGURE B-9/T.412

Generic Logical structure

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 can 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/T.412) 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/T.412).

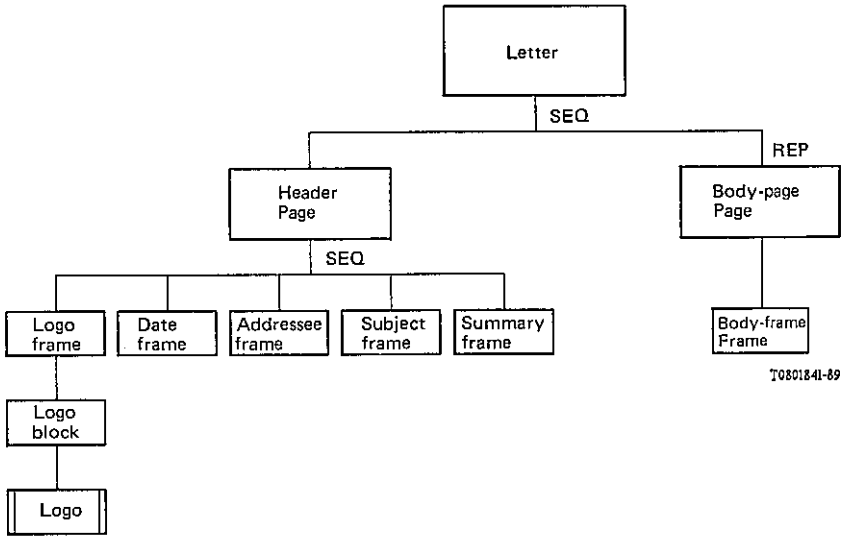


FIGURE B-10/T.412

Generic layout structure

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 directives. Each logical object description contains an attribute which relates a particular layout style with 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 can 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/T.412, 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 § 6 and 7).

Referring to Figure B-9/T.412, 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 must 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 must be "header" followed by "body" (i.e. in the order left to right as indicated in the diagram).

The object descriptions subordinate to "header" consist of "date", "addressee", "subject", and "summary". All of these must occur once in any corresponding specific logical structure and must 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 must occur once only. There is no logical object "logo" because this is assumed to be part of the generic layout structure.

In this example, both "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" to both 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.

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/T.412 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/T.412, the generic layout structure called "letter" indicates that a specific layout structure created by the document layout process must 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. Note that 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 *Objet descriptions and object class descriptions*

The object class descriptions pertaining to the generic logical structure are given in Table B-4/T.412. Table B-5/T.412 describes the presentation styles and the layout styles associated with the document.

TABLE B-4/T.412

Constituents for Figure B-9/T.412
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)

TABLE B-4/T.412 *(Cont.)*

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
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/T.412

Styles for Figure B-9/T.412 - 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
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 0 1 (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/T.412 lists the object descriptions for the objects in the specific logical structure. The object descriptions in this table must be interpreted together with the object class descriptions given in Table B-4/T.412. Note that all content portion descriptions are grouped together since it is assumed that interchange format class A (see Recommendation T.415) will be used to interchange this particular document.

TABLE B-6/T.412

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
Object type Object identifier Object class User-visible name Content portions	BASIC LOGICAL 3 0 1 2 0 1 (Addressee) "Addressee" 0
Object type Object identifier Object class User-visible name Content portions	BASIC LOGICAL 3 0 2 2 0 2 (Subject) "Subject" 0
Object type Object identifier Object class User-visible name Subordinates	COMPOSITE LOGICAL 3 0 3 2 0 3 (Summary) "Summary" 0
Object type Object identifier Object class User-visible name Content portions	BASIC LOGICAL 3 0 3 0 2 0 3 1 (Summary-paragraph) "Summary-paragraph" 0
Object type Object identifier Object class User-visible name Subordinates	COMPOSITE LOGICAL 3 1 2 1 (Body) "Body" 0, 1, 2, 3, 4, 5, 6

TABLE B-6/T.412 *(Cont.)*

Object type Object identifier Object class User-visible name Content portions	BASIC LOGICAL 3 1 0 2 1 1 (Body-paragraph) Paragraph A 0
Object type Object identifier Object class User-visible name Line spacing Content portions	BASIC LOGICAL 3 1 1 2 1 1 (Body-paragraph) Paragraph B 400 0
Object type Object identifier Object class User-visible name Subordinates	COMPOSITE LOGICAL 3 1 2 2 1 0 (Figure) "Figure" 0, 1
Object type Object identifier Object class User-visible name Content portions	BASIC LOGICAL 3 1 2 0 2 1 0 0 (Drawing) "Drawing" 0
Object type Object identifier Object class User-visible name Content portions	BASIC LOGICAL 3 1 2 1 2 1 0 1 (Caption) "Caption" 0
Object type Object identifier Object class User-visible name Content portions	BASIC LOGICAL 3 1 3 2 1 1 (Body-paragraph) Paragraph C 0
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

TABLE B-6/T.412 *(Cont.)*

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/T.412 lists the object class descriptions pertaining to the generic layout structure.

TABLE B-7/T.412

Constituents for Figure B-10/T.412
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 identifier 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

TABLE B-7/T.412 *(Cont.)*

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*/

B.6 *Specific layout structure*

Figure B-11/T.412 illustrates the specific layout structure generated by the document layout process (see § 6) from the specific logical structure, generic logical structure and generic layout structure described in § B-5.

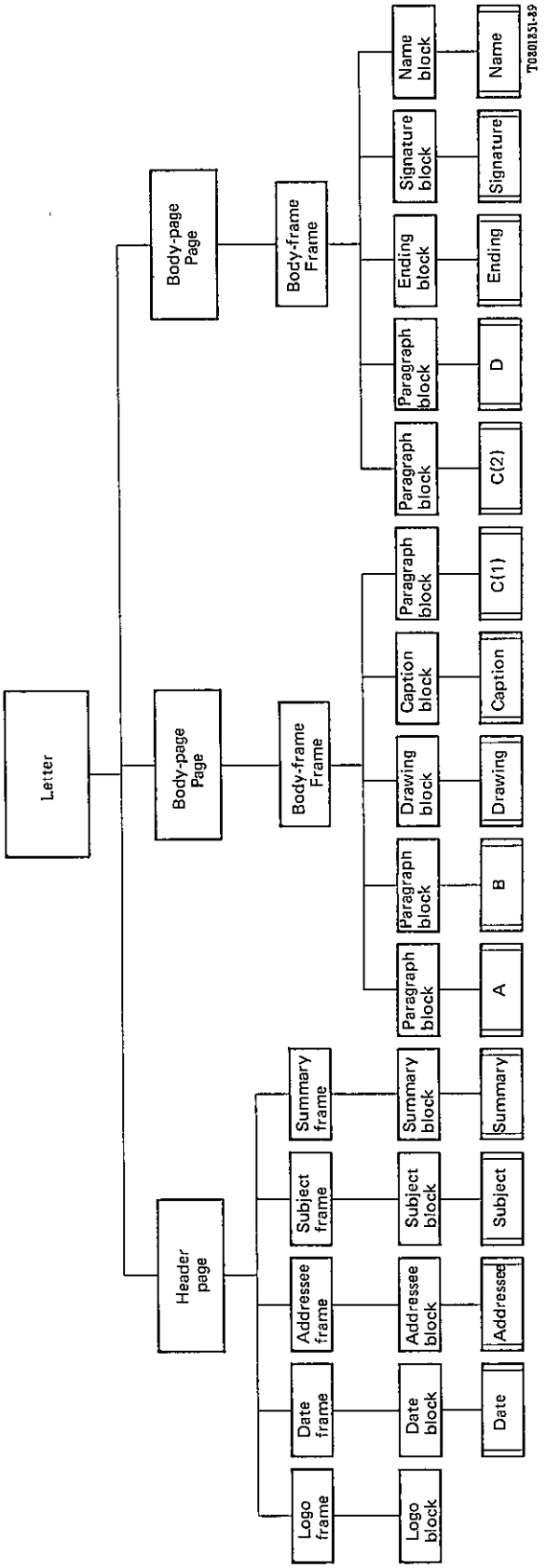


FIGURE B-11/T.412

Specific layout structure

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 sub-section 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/T.412 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/T.412 are listed in Table B-8/T.412. These are to be interpreted in conjunction with the layout object class descriptions listed in Table B-7/T.412. Also, note that in Table B-8/T.412, 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/T.412. In a formatted-processable form document, the content portions in Table B-8/T.412 replace the ones described in Table B-6/T.412.

TABLE B-8/T.412

Constituents for Figure B-11/T.412
Specific layout structure

Object type Identifier Object class User-visible name Subordinates	DOCUMENT LAYOUT ROOT Object 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

TABLE B-8/T.412 *(Cont.)*

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
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

TABLE B-8/T.412 *(Cont.)*

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
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

TABLE B-8/T.412 *(Cont.)*

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"
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*/.

TABLE B-8/T.412 *(end)*

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"

Figures B-12/T.412, B-13/T.412 and B-14/T.412 illustrate the layout structure of the specimen document corresponding to that described in Table b-8/T.412, by outlining the frames and blocks within each page.

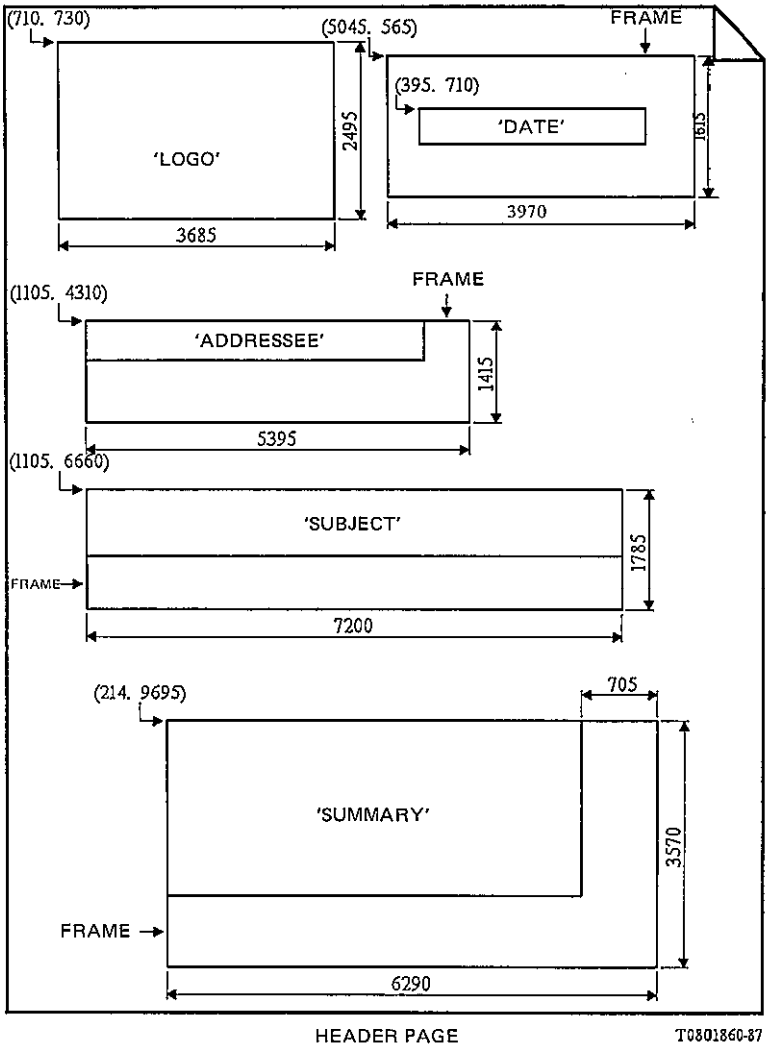


FIGURE B-12/T.412

Layout structure of "letter" showing "pages", "frames" and "blocks" (1)

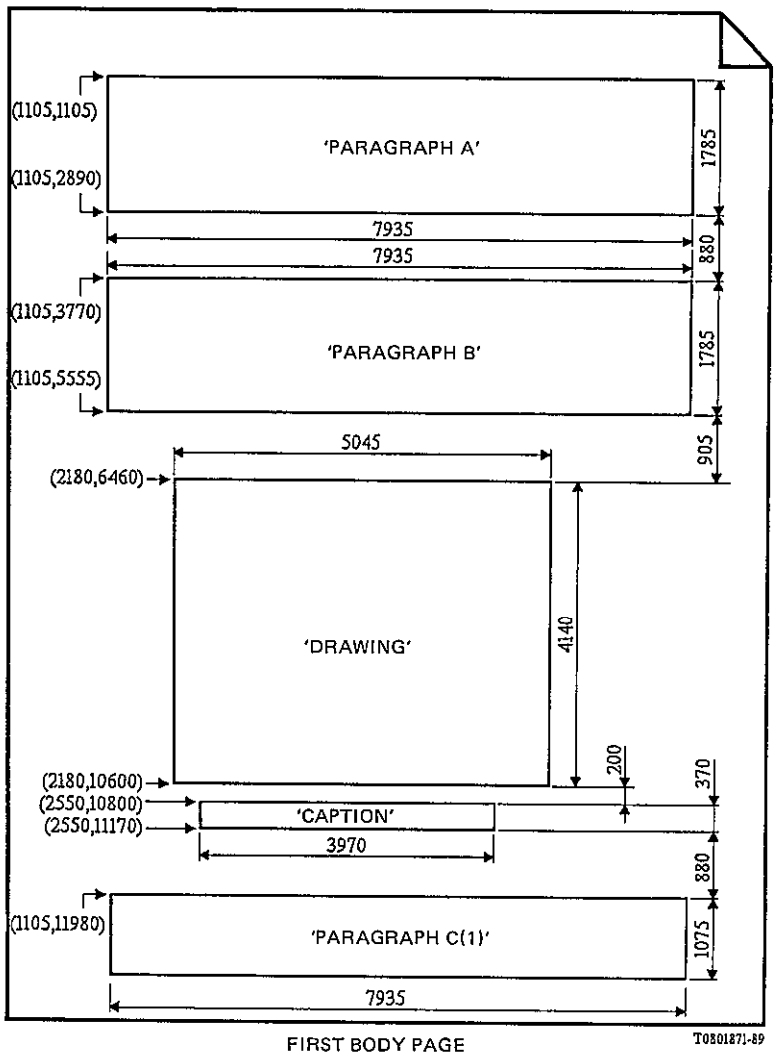


FIGURE B-13/T.412

Layout structure of "letter" showing "pages", "frames" and "blocks" (2)

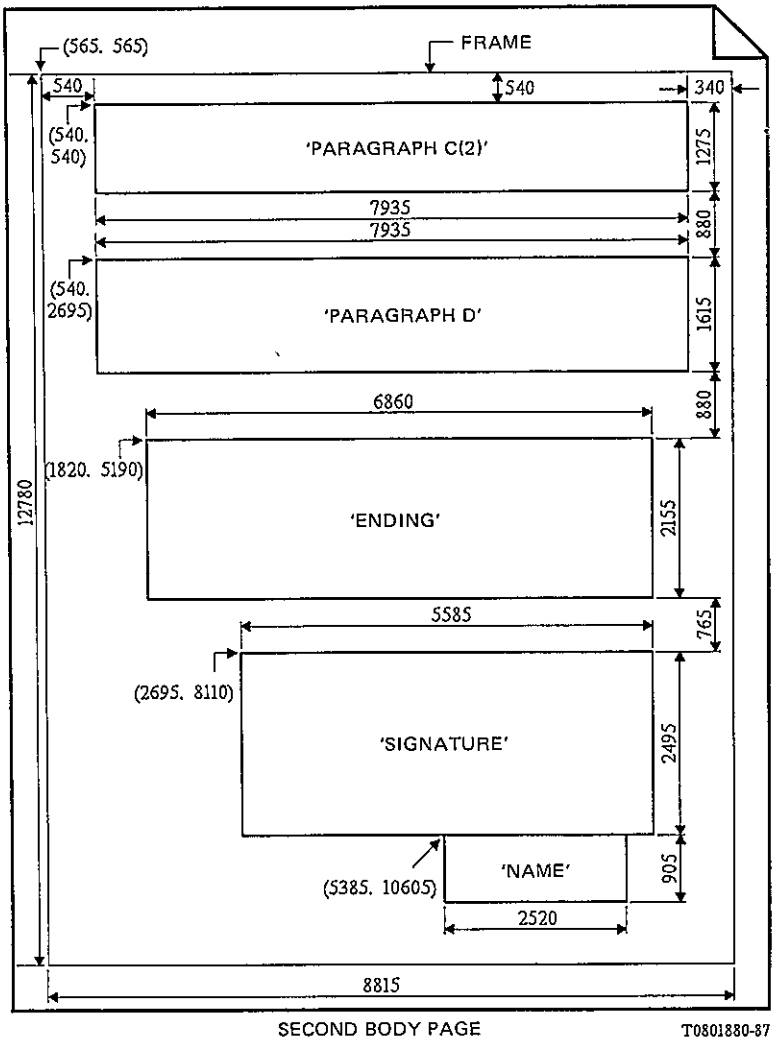


FIGURE B-14/T.412

Layout structure of "letter" showing "pages", "frames" and "blocks" (3)

ANNEX C

(to Recommendation T.412)

(Informative)

Examples of document architecture levels

C.1 *Document architecture levels*

TABLE C-1/T.412

Document architecture levels

Document architecture level	Generic logical structure	Specific logical structure, permitted object types	Generic layout structure	Specific layout structure, permitted object types
FDA 0	none	none	none	document layout root page
FDA 1	none	none	partial	document layout root, page, block
FDA 2	none	none	partial	all object types
PDA 0	none	document logical root, basic logical objects	none	none
PDA 1	none	document logical root, basic logical objects	complete	none
PDA 2	partial	all object types	none	none
PDA 3	partial	all object types	complete	none
PDA 4	partial	all object types	none	none
PDA 5	partial	all object types	complete	none
FPDA 0	none	document logical root, basic logical objects	complete	all object types
FPDA 1	partial	all object types	complete	all object types
FPDA 2	partial	all object types	complete	all object types

The abbreviations used in Table C-1/T.412 have the following meaning:

- *None*

  The corresponding structure is not present.

- *Partial*

  A partial generator set of either logical object class descriptions or layout object class descriptions is present. Such a set may include object class descriptions corresponding to each object and no object class descriptions as special cases.

- *Complete*

  A complete generator set of either logical object class descriptions or layout object class descriptions is present.

- *All object types*

  The columns defining specific logical structures and specific layout structures specify the object types that may be specified in the corresponding structure. "All object types" indicates that any combination of object types is permissible. In the case of specific logical structure, a structure consisting of just document logical root and basic logical objects is a special case. In the case of specific layout structure, a structure containing just document layout root, pages and blocks is a special case.

Further restrictions of the use of these document architecture levels:

a) Pages as basic objects are only permitted in levels FDA 0 and FDA 1.

b) If a specific logical structure and a generic layout structure are both present in a document, then the generic layout structure is always required to be complete, in order that the layout process can be controlled.

c) The layout process can be controlled, so as to meet the specification in Section 6, in levels PDA 1, PDA 3, PDA 5 and in the FPDA levels.

d) The editing process can be controlled in levels PDA 4, PDA 5 and FPDA 2.

TABLE C-2/T.412

FDA levels: layout structures

Attributes	FDA 0		FDA 1		FDA 2	
	G	S	G	S	G	S
Shared attributes						
Object type		M	M	D	M	D
Object identifier		NM		M		M
Object class identifier			M		M	
Object class				NM		NM
Subordinates		NM		M		M
Content portions		NM	NM	NM	NM	NM
Resource					NM	
Presentation style					NM	D
Content architecture class					NM	D
Content type		D	NM	D		
User-readable comments			NM	D	NM	D
Applications comments					NM	D
User-visible name					NM	D
Bindings					NM	D
Default value lists		NM	NM	NM	NM	NM
Document layout root attributes						
Page set attributes						
No additional attributes						
Page attributes						
Dimensions		D	NM	D	NM	D
Presentation attributes		D	NM	D	NM	D
Colour					NM	D
Transparency					NM	D
Page position					NM	D
Balance					NM	D
Medium type					NM	D
Frame attributes						
Position					NM	D
Dimensions					NM	D
Colour					NM	D
Transparency					NM	D
Border					NM	D
Block attributes						
Position			NM	D	NM	D
Dimensions			NM	D	NM	D
Presentation attributes		D	NM	D	NM	D
Colour			NM	D	NM	D
Transparency			NM	D	NM	D
Border			NM	D	NM	D

G For object class descriptions
S For object descriptions
M Mandatory at this levelNM Attribute non-mandatory at this level
D Attribute defaultable at this level
<blank entry> Not applicable

C.3 *Attributes applicable to PDA document architecture levels*

TABLE C-3/T.412

PDA levels: logical structures

Attributes	PDA 0		PDA 1		PDA 2		PDA 3		PDA 4		PDA 5	
	G	S	G	S	G	S	G	S	G	S	G	S
Shared Attributes												
Object type		M		M	M	D	M	D	M	D	M	D
Object identifier		NM		NM		M		M		M		M
Object class identifier					M		M		M		M	
Generator for subordinates					NM		NM		M		M	
Content generator					NM	NM	NM	NM	NM	NM	NM	NM
Object class						NM		NM		M		M
Subordinates		NM		M		M		M		M		M
Content portions		NM		NM	NM	NM	NM	NM	NM	NM	NM	NM
Resource					NM		NM		NM		NM	
Presentation style				NM	NM	D	NM	D	NM	D	NM	D
Content architecture class					NM	D	NM	D	NM	D	NM	D
Content type		D		D								
User readable comments					NM	D	NM	D	NM	D	NM	D
Applications comments					NM	D	NM	D	NM	D	NM	D
User visible name					NM	D	NM	D	NM	D	NM	D
Bindings									NM	D	NM	D
Default value lists		NM		NM	NM	NM	NM	NM	NM	NM	NM	NM
Document logical root attributes												
Protection					NM	D	NM	D	NM	D	NM	D
Layout style					NM	D	NM	D	NM	D	NM	D
Composite logical object attributes												
Protection					NM	D	NM	D	NM	D	NM	D
Layout style					NM	D	NM	D	NM	D	NM	D
Basic logical object attributes												
Protection					NM	D	NM	D	NM	D	NM	D
Layout style					NM	D	NM	D	NM	D	NM	D

M Mandatory at this level
NM Attribute non-mandatory at this level
D Attribute defaultable at this level
G For object class descriptions
S For object descriptions
<blank entry> Not applicable

TABLE C-4/T.412

PDA levels: applicability of layout directive attributes

Layout directive attributes	PDA 0		PDA 1		PDA 2		PDA 3		PDA 4		PDA 5	
	G	S	G	S	G	S	G	S	G	S	G	S
Document logical root attributes												
Layout object class					NM	D	NM	D	NM	D	NM	D
Composite logical object attributes												
Indivisibility					NM	D	NM	D	NM	D	NM	D
Layout object class					NM	D	NM	D	NM	D	NM	D
New layout object					NM	D	NM	D	NM	D	NM	D
Same layout object					NM	D	NM	D	NM	D	NM	D
Synchronization					NM	D	NM	D	NM	D	NM	D
Basic logical object attributes												
Block alignment		D		D	NM	D	NM	D	NM	D	NM	D
Concatenation		D		D	NM	D	NM	D	NM	D	NM	D
Fill order		D		D	NM	D	NM	D	NM	D	NM	D
Indivisibility		D		D	NM	D	NM	D	NM	D	NM	D
Layout category		D		D	NM	D	NM	D	NM	D	NM	D
Layout object class		D		D	NM	D	NM	D	NM	D	NM	D
New layout object		D		D	NM	D	NM	D	NM	D	NM	D
Offset		D		D	NM	D	NM	D	NM	D	NM	D
Same layout object		D		D	NM	D	NM	D	NM	D	NM	D
Separation		D		D	NM	D	NM	D	NM	D	NM	D
Synchronization		D		D	NM	D	NM	D	NM	D	NM	D

NM Attribute non-mandatory at this level
D Attribute defaultable at this level
G For object class descriptions
S For object descriptions
<blank entry> Not applicable

TABLE C-5/T.412

PDA levels: layout structures

Attributes	PDA 0		PDA 1		PDA 2		PDA 3		PDA 4		PDA 5	
	G	S	G	S	G	S	G	S	G	S	G	S
Shared attributes												
Object type			M				M				M	
Object class identifier			M				M				M	
Generator for subordinates			M				M				M	
Content generator			NM				NM				NM	
Content portions			NM				NM				NM	
Resource			NM				NM				NM	
Presentation style			NM				NM				NM	
Content architecture class			NM				NM				NM	
User readable comments			NM				NM				NM	
Applications comments							NM				NM	
User visible name							NM				NM	
Binding							NM				NM	
Default value lists			NM				NM				NM	
Page attributes												
Dimensions			NM				NM				NM	
Colour			NM				NM				NM	
Transparency			NM				NM				NM	
Border			NM				NM				NM	
Balance			NM				NM				NM	
Imaging order			NM				NM				NM	
Page position			NM				NM				NM	
Medium type			NM				NM				NM	
Frame attributes												
Position			NM				NM				NM	
Dimensions			NM				NM				NM	
Colour			NM				NM				NM	
Transparency			NM				NM				NM	
Border			NM				NM				NM	
Balance			NM				NM				NM	
Layout path			NM				NM				NM	
Logical source							NM				NM	
Permitted categories			NM				NM				NM	
Imaging order			NM				NM				NM	
Block attributes												
Position			NM				NM				NM	
Dimensions			NM				NM				NM	
Colour			NM				NM				NM	
Transparency			NM				NM				NM	
Border			NM				NM				NM	

M Mandatory at this level
NM Attribute non-mandatory at this level
D Attribute defaultable at this level
G For object class descriptions
S For object descriptions
<blank entry> Not applicable

C.4 *Attributes applicable to FPDA document architecture levels*

TABLE C-6/T/412

FPDA levels: logical structures

Attributes	FPDA 0		FPDA 1		FPDA 2	
	G	S	G	S	G	S
Shared attributes						
Object type		M	M	D	M	D
Object identifier		M		M		M
Object class identifier			M		M	
Generator for subordinates			NM		M	
Content generator			NM	NM	NM	NM
Object class				NM		M
Subordinates		M		M		M
Content portions		NM	NM	NM	NM	NM
Resource			NM		NM	
Presentation style		NM	NM	D	NM	D
Content architecture class			NM	D	NM	D
Content type		D				
User readable comments			NM	D	NM	D
Applications comments			NM	D	NM	D
User visible name			NM	D	NM	D
Bindings					NM	D
Default value lists		NM	NM	NM	NM	NM
Document logical root attributes						
Protection			NM	D	NM	D
Layout style			NM	D	NM	D
Composite logical object attribute						
Protection			NM	D	NM	D
Layout style			NM	D	NM	D
Basic logical object attributes						
Protection			NM	D	NM	D
Layout style			NM	D	NM	D

M .Mandatory at this level
NM Attribute non-mandatory at this level
D Attribute defaultable at this level
G For object class descriptions
S For object descriptions
<blank entry> Not applicable

TABLE C-7/T.412

FPDA level: applicability of layout directive attributes

Layout directive attributes	FPDA 0		FPDA 1		FPDA 2	
	G	S	G	S	G	S
Document logical root attributes						
Layout object class	NM	D	NM	D	NM	D
Composite logical object attributes						
Indivisibility			NM	D	NM	D
Layout object class			NM	D	NM	D
New layout object			NM	D	NM	D
Same layout object			NM	D	NM	D
Synchronization			NM	D	NM	D
Basic logical object						
Attributes						
Block alignment		D	NM	D	NM	D
Concatenation		D	NM	D	NM	D
Fill order		D	NM	D	NM	D
Indivisibility		D	NM	D	NM	D
Layout category		D	NM	D	NM	D
Layout object class		D	NM	D	NM	D
New layout object		D	NM	D	NM	D
Offset		D	NM	D	NM	D
Same layout object		D	NM	D	NM	D
Separation		D	NM	D	NM	D
Synchronization		D	NM	D	NM	D

NM Attribute non-mandatory at this level
D Attribute defaultable at this level
G For object class descriptions
S For object descriptions
<blank entry> Not applicable

TABLE C-8/T.412

FPDA levels: layout structures

Attributes	FPDA 0		FPDA 1		FPDA 2	
	G	S	G	S	G	S
Shared attributes						
Object type	M	D	M	D	M	D
Object identifier		M		M		M
Object class identifier	M		M		M	
Generator for subordinates	M		M		M	
Content generator	NM		NM		NM	
Object class		M		M		M
Subordinates		M		M		M
Content portions	NM	NM	NM	NM	NM	NM
Resource	NM		NM		NM	
Presentation style	NM	D	NM	D	NM	D
Content architecture class	NM	D	NM	D	NM	D
User readable comments	NM	D	NM	D	NM	D
Applications comments	NM	D	NM	D	NM	D
User-visible name	NM	D	NM	D	NM	D
Bindings	NM	D	NM	D	NM	D
Default value lists	NM	NM	NM	D	NM	NM
Document layout root attributes						
Page set attributes						
No additional attributes						
Page attributes						
Dimensions	NM	D	NM	D	NM	D
Balance	NM	D	NM	D	NM	D
Imaging order		NM		NM		NM
Colour	NM	D	NM	D	NM	D
Transparency	NM	D	NM	D	NM	D
Page position	NM	D	NM	D	NM	D
Medium type	NM	D	NM	D	NM	D
Frame attributes						
Position	NM	D	NM	D	NM	D
Dimensions	NM	D	NM	D	NM	D
Colour	NM	D	NM	D	NM	D
Transparency	NM	D	NM	D	NM	D
Border	NM	D	NM	D	NM	D
Balance	NM	D	NM	D	NM	D
Layout path	NM	D	NM	D	NM	D
Logical source	NM		NM		NM	
Permitted categories	NM	D	NM	D	NM	D
Imaging order		NM		NM		NM
Block attributes						
Position	NM	D	NM	D	NM	D
Dimensions	NM	D	NM	D	NM	D
Presentation attributes	NM	D	NM	D	NM	D
Colour	NM	D	NM	D	NM	D
Transparency	NM	D	NM	D	NM	D
Border	NM	D	NM	D	NM	D

G For object class descriptions
S For object descriptions
M Mandatory at this level
NM Attribute non-mandatory at this level
D Attribute defaultable at this level
<blank entry> Not applicable

ANNEX D

(to Recommendation T.412)

(Informative)

Examples of particular document architectures features

D.1 *Layout of frames*

This paragraph presents examples of the use of the attributes "position", "dimensions" and "layout path" as applicable to frames.

D.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 subordinate 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 value 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 the Recommendation 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 § 5.4.1.2;
- VD – vertical dimension: the value is either the default value defined in the Recommendation 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 § 5.4.1.2;

Content of blocks is indicated by annotation of form <...>.

The notation adopted for the specification of construction expressions is that specified in § A.2.2.

D.1.2 *Position attribute for a frame in a fixed position (see Figure D-1/T.412)*

Illustration

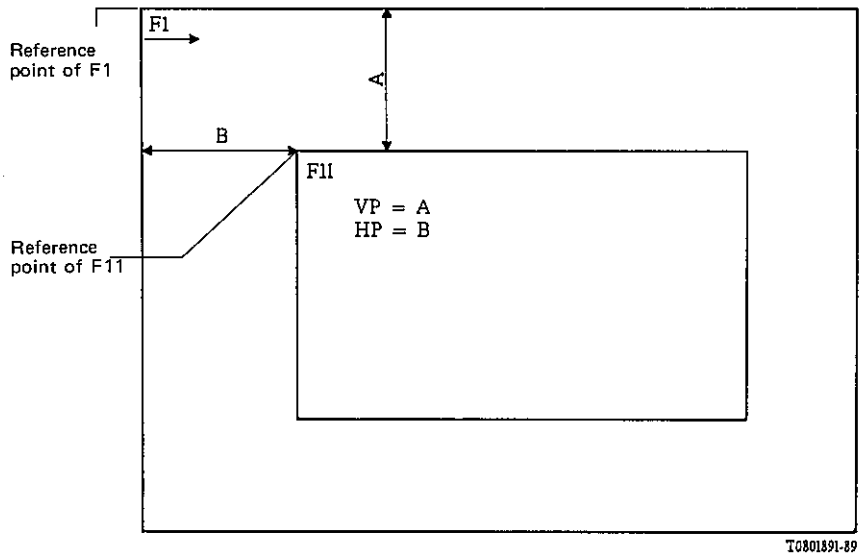


FIGURE D-1/T.412

Fixed position frame

D.1.3 *Single column text of variable height (see Figure D-2/T.412)*

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 to possibly include other forms of layout on the same page. This may be used to hold a paragraph of text or a figure.

Generator for subordinates

P: SEQ (CHO (F0 ...))
F0: none

Illustration

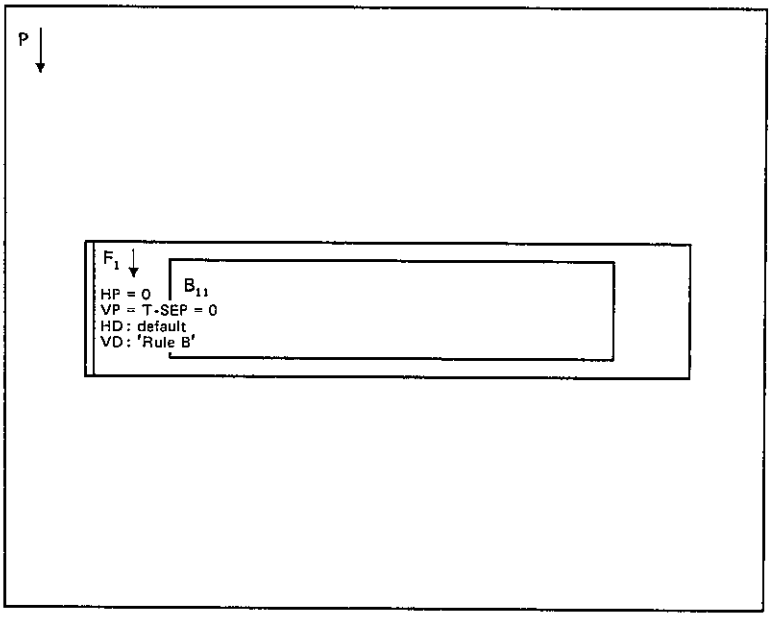


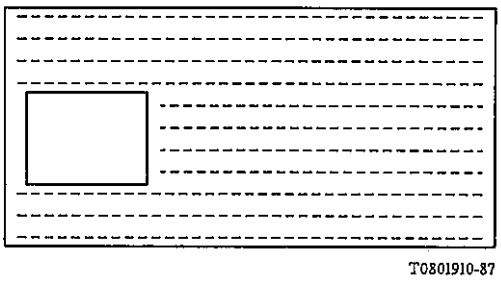
FIGURE D-2/T.412

Single column text of variable height

D.1.4 *"Wrapping" of text around a left aligned picture (see Figure D-3/T.412)*

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 on the left of the column.



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.

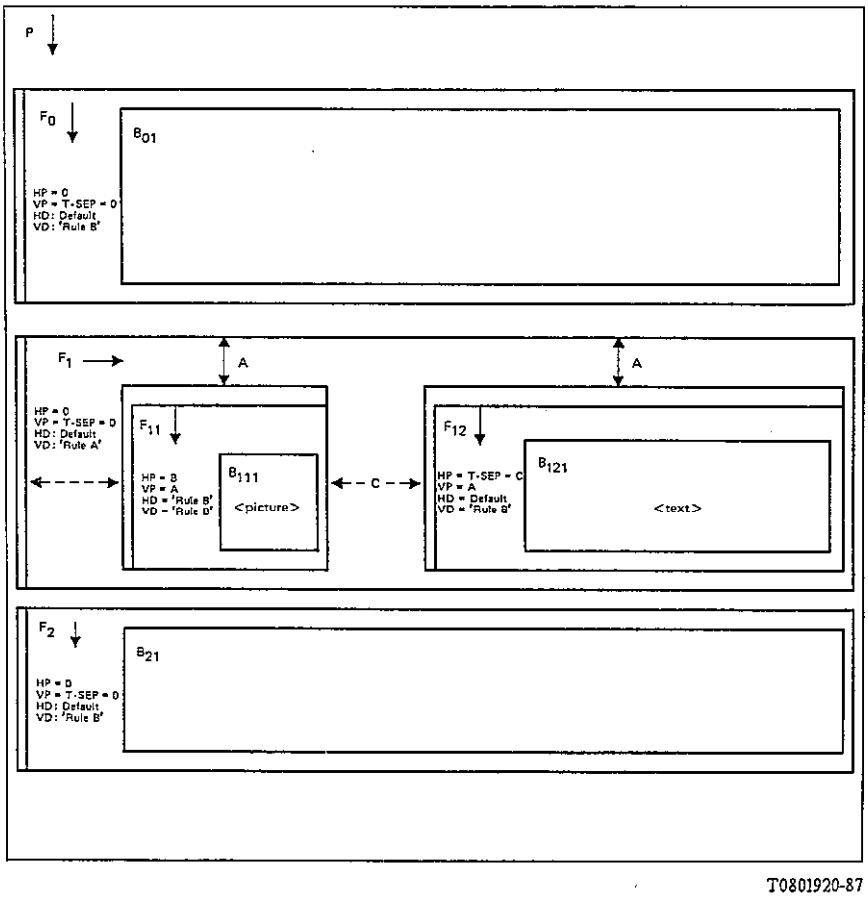
The picture specifies layout object class = F11 in order that each picture that occurs causes one instance of F1.

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 § D.1.3
F1: SEQ (F11 F12)
F11, F12: none

Illustration



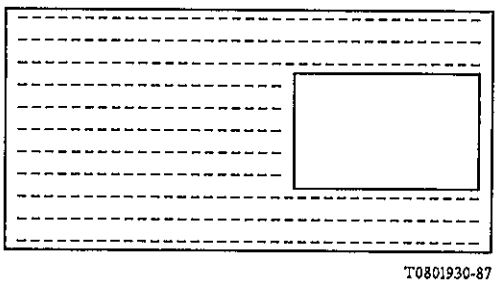
T0801920-87

FIGURE D-3/T.412

"Wrapping" of text around a left aligned picture

Variations

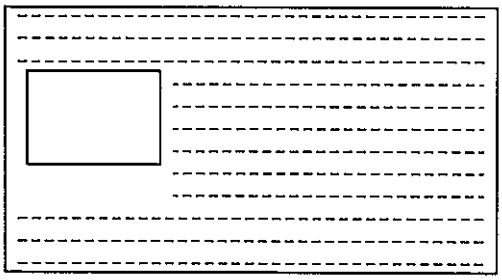
1) If the picture is required to be right aligned:



T0801930-87

then the layout path for frame F1 would be reversed (180^{o}), 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.

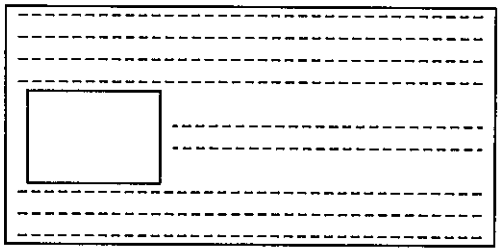
2) If the required layout were that the material to be placed beside the picture is to be associated with it and that the description of the figure should be completed in the indented column, before resuming full width lines:



T0801940-87

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:



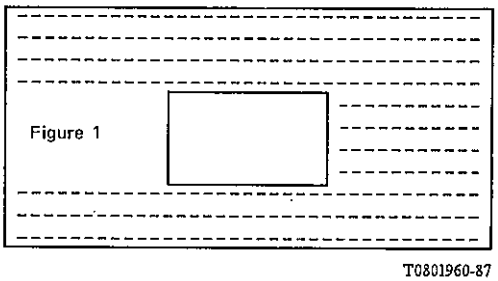
T0801950-87

then the vertical position of F12 is specified by the sub-parameter "alignment" with the value 'centred', other parameters would be unchanged.

D.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.



T0801960-87

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 is preceding 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 should 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 § D.1.3
F1: SEQ (F11 F12 F13)
F11, F12, F13: none.

Illustration

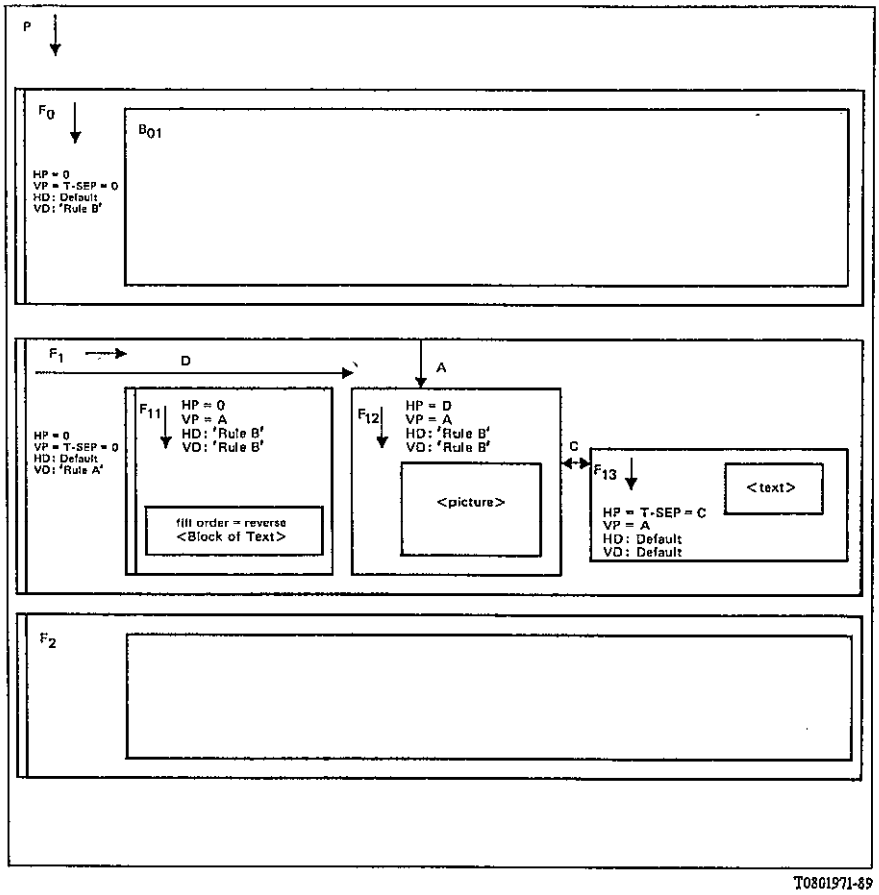
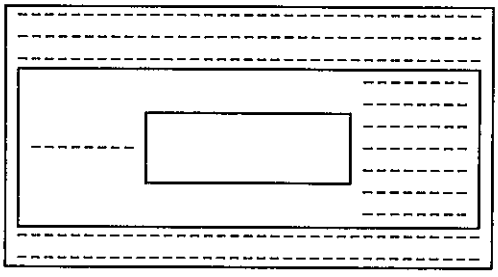


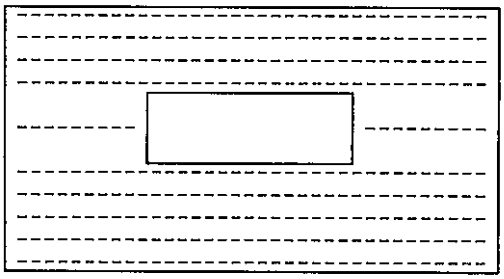
FIGURE D-4/T.412

Horizontally centred picture with text of both sides

Variations

1) The descriptive text on the right associated with the text might be longer or shorter than the figure.





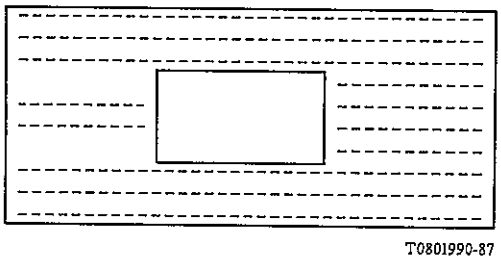
T0801980-87

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.

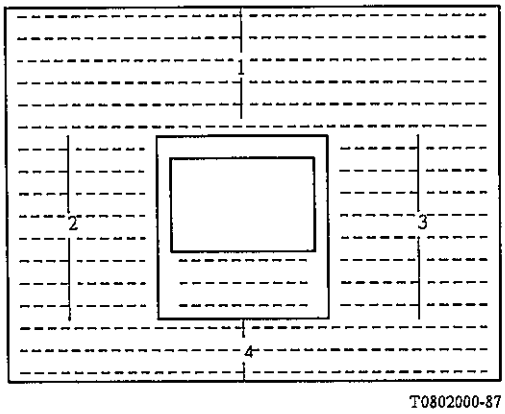
2) The picture is centred and has all its associated text on the left side, and the main text is running down on the right side.



T0801990-87

The only modifications are that frame F13 should 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 is running down on both sides of the figure, and the figure heading below it. In addition, the heading has a border.



T0802000-87

The reading order of the main text is 1, 2, 3, 4, as indicated.

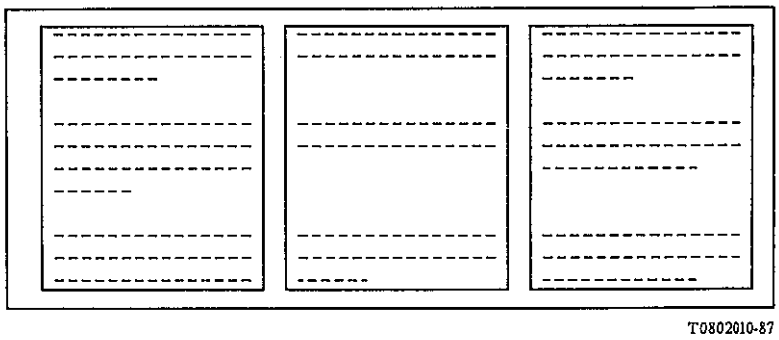
The figure and its heading is represented by a composite logical object with the picture and the figure text as the only two subordinate logical objects, the picture is the first of these.

The composite logical object specifies the attribute "layout object class" F12. There is an additional block in frame F12, not illustrated in Figure D-4/T.412, 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.

D.1.6 *Multi-lingual synchronised text (see Figure D-5/T.412)*

Informal description of feature illustrated

Three columns of text on a page, each column containing text in different languages. There is required to be horizontal synchronisation at various points, for example, paragraphs, sections.



T0802010-87

Assumptions

Positions and widths of columns are fixed values.

The sets of information to be synchronised 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

1) 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^{o} 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^{o}. Therefore the Japanese text would take up as little space in the vertical dimension as possible.

Illustration

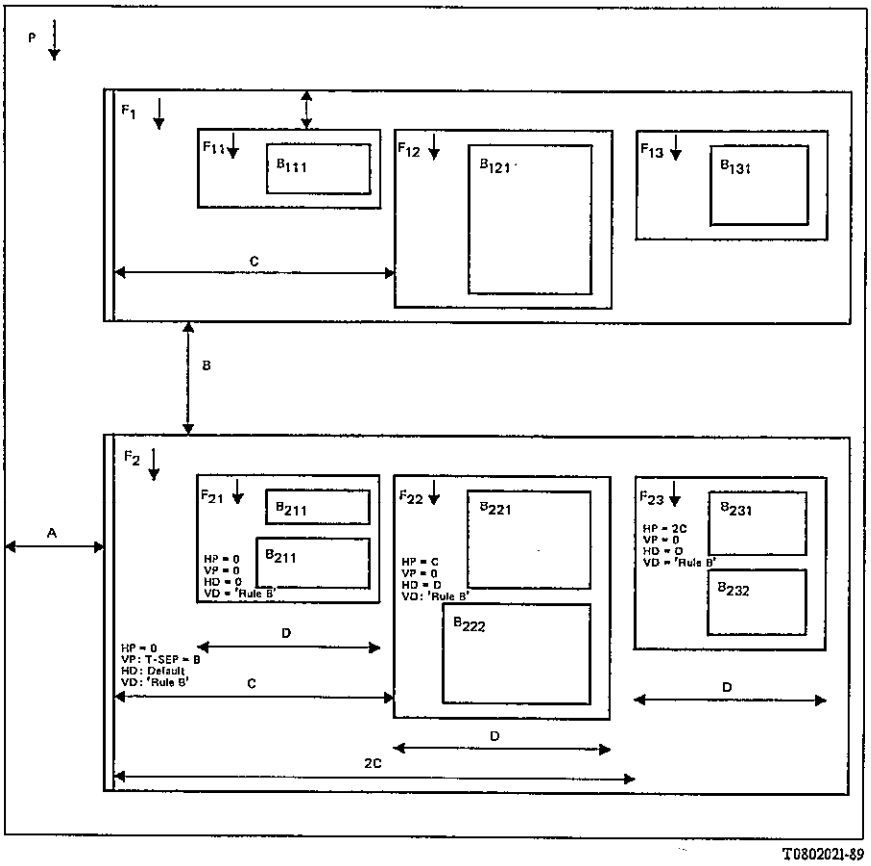


FIGURE D-5/T.412

Multi-lingual synchronised text

D.1.7 *Footnote placement (see Figure D-6/T.412)*

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

Illustration

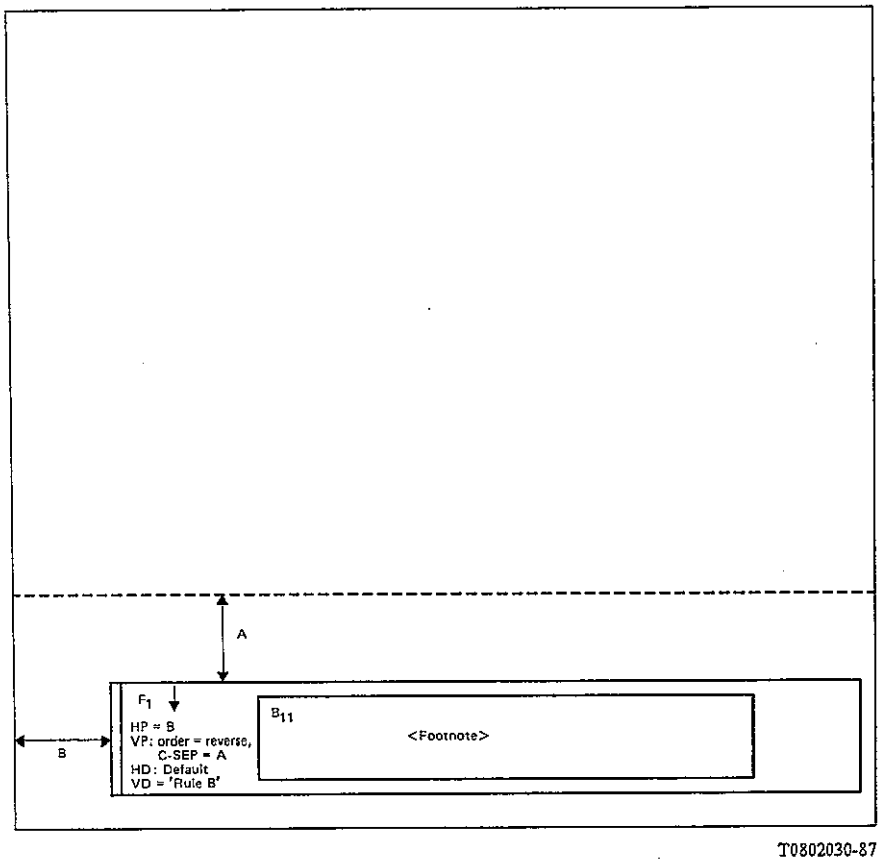


FIGURE D-6/T.412

Footnote placement

D.1.8 *Tabular layout (see Figure D-7/T.412)*

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 and 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 (... F1 ...))
F1: SEQ (F11 ... F1i ...)
F1i: SEQ (FF1i1 ... FF1ij ...)

Illustration

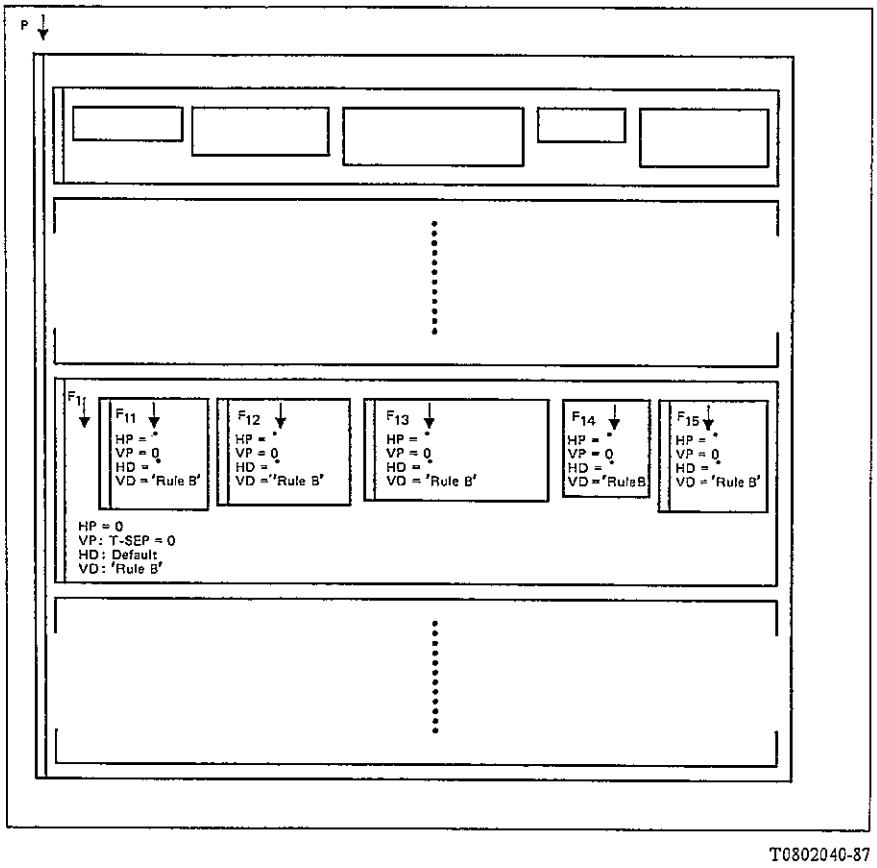


FIGURE D-7/T.412

Tabular layout

D.2 *Layout of blocks*

This paragraph contains illustrations of the use of the attributes:

- concatenation;
- offset;
- separation;
- layout path;
- fill order;
- block alignment.

D.2.1 *Concatenation*

Concatenation can be used to lay out an automatically generated chapter number with a chapter heading, see Figure D-8/T.412. 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 D-9/T.412.

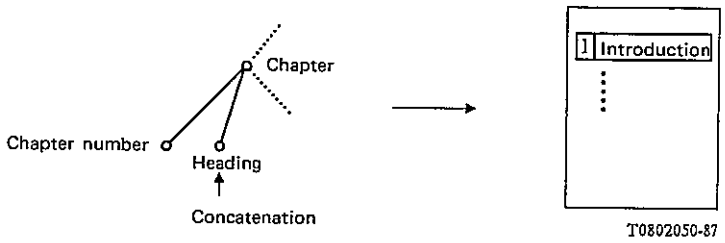


FIGURE D-8/T.412

Concatenation of chapter number and title

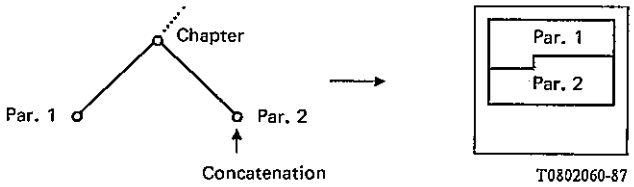


FIGURE D-9/T.412

Concatenation of parts of a paragraph

D.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 D-10/T.412, the attribute "layout path" is assumed to have its default value of 270^{o} 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.

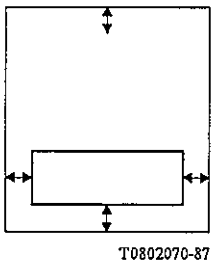


FIGURE D-10/T.412

Illustration of attribute "offset"

D.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 D-11/T.412.

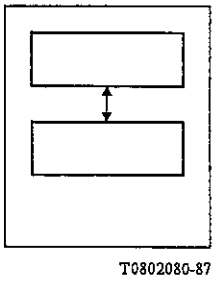


FIGURE D-11/T.412

Illustration of attribute "separation"

D.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 D-12/T.412.

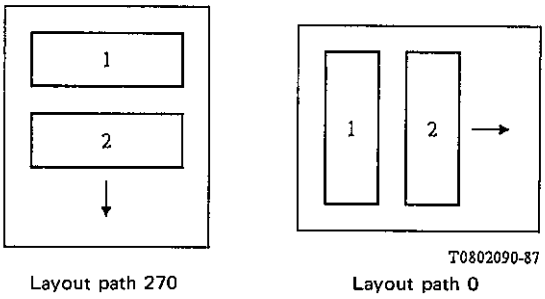


FIGURE D-12/T.412

Layout path 0° and 270°

D.2.5 *Fill order*

Fill order can be used to lay out text or figures from the top or 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^{O}), 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 D-13/T.412 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.

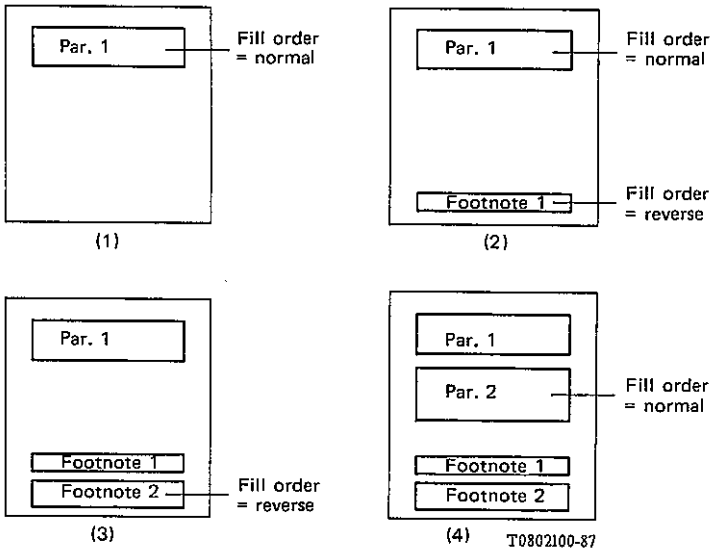


FIGURE D-13/T.412

Illustration of fill order

D.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 D-14/T.412 illustrates the use for aligning figures to the left or right. The figure illustrates that, with the default value of the attribute "layout path" (270^{o}):

- '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.

This figure also illustrates that the alignment is constrained by the specification of the attribute "offset".

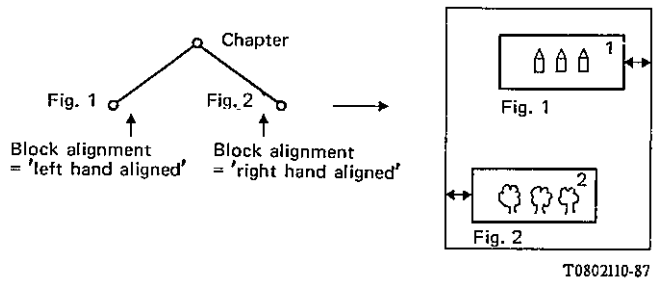


FIGURE D-14/T.412

Illustration of block alignment

D.2.7 *Illustration of attributes used in combination*

This paragraph illustrates the combined use of the attributes "offset", "separation" and "fill order".

In this paragraph 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 D-15/T.412 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'.

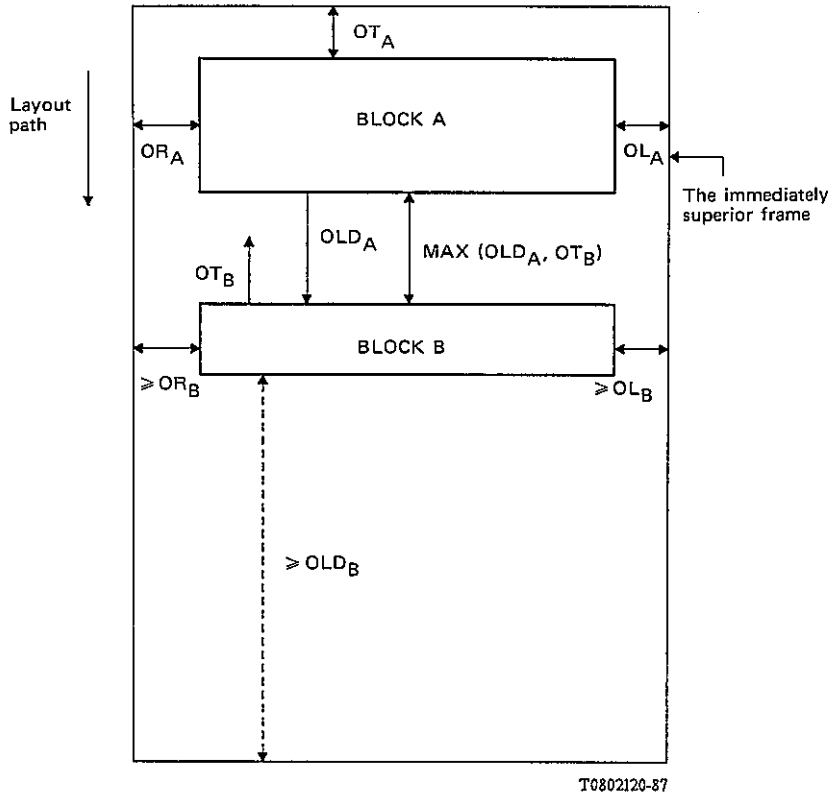


FIGURE D-15/T.412

Illustration of the attribute "separation"

Figure D-16/T.412 illustrates the combined use of the attributes "offset" and "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 D-16/T.412 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.

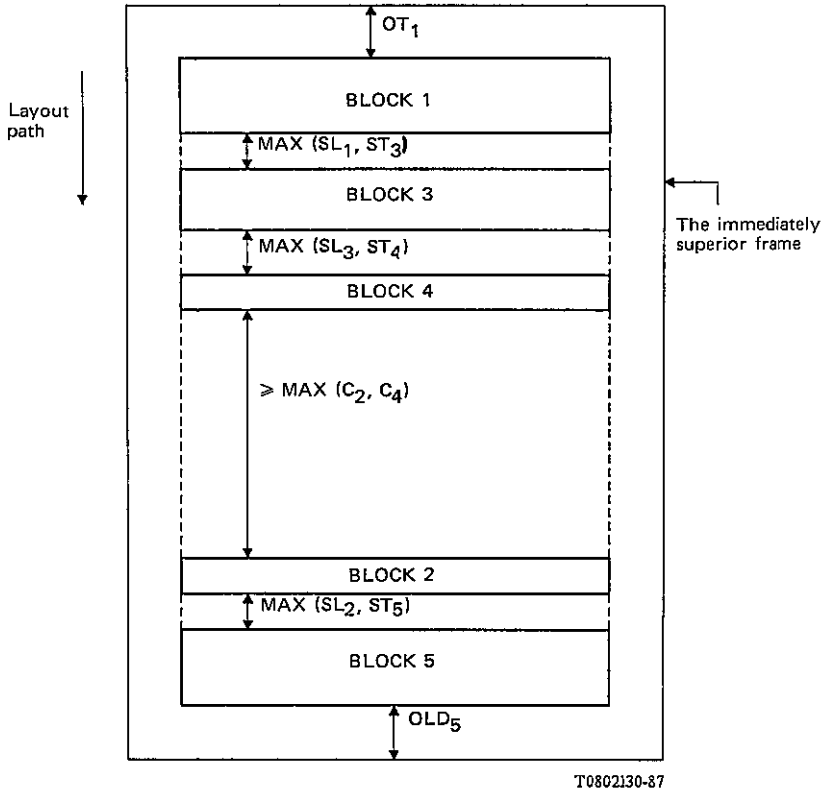


FIGURE D-16/T.412

Illustration of layout of blocks in normal and reverse order

Figure D-17/T.412 illustrates the combined use of the attributes "offset" and "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 D-17/T.412 illustrates the determination of the available area during the part of the layout process concerned with positioning of blocks.

In Figure D-17/T.412, 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.

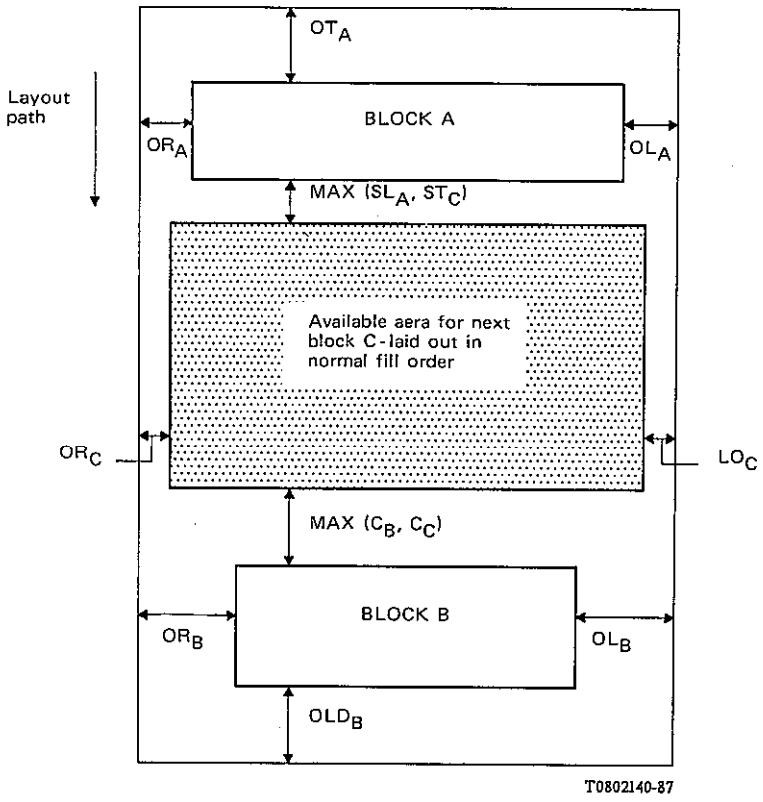


FIGURE D-17/T.412

Illustration of layout of logical objects in normal order

Figure D-18/T.412 illustrates the determination of the available area during the part of the layout process concerned with positioning of blocks.

In Figure D-18/T.412, 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 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 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.

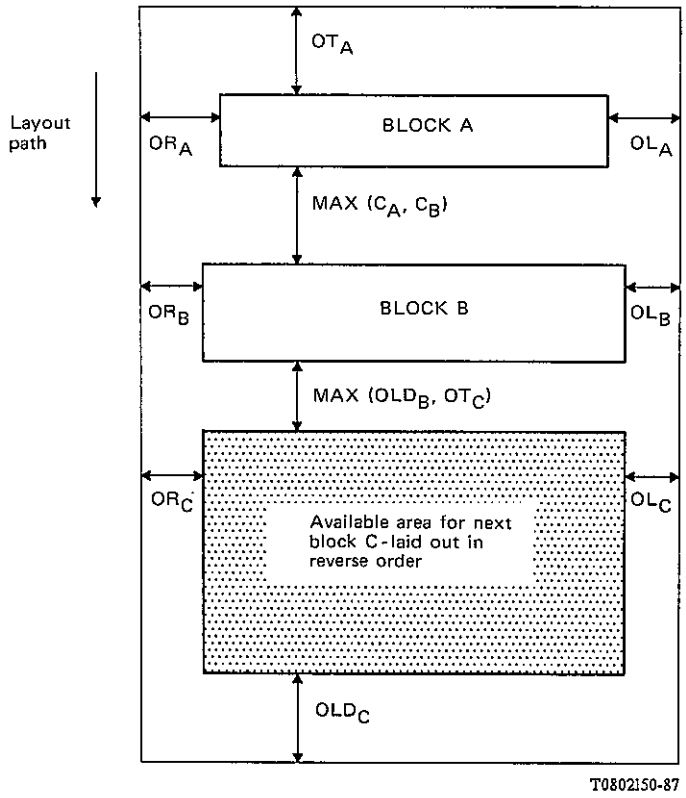


FIGURE D-18/T.412

Illustration of layout of logical objects in reverse order

D.3 *Further constraints on layout*

This paragraph contains illustrations of the use of the attributes:

- layout object class;
- synchronization;
- new layout object;
- same layout object;
- indivisibility;
- balance.

D.3.1 *Layout object class*

The attribute "layout object class" can 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 § D.1.4 and Figure D-3/T.412, 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 § D.1.5 and Figure D-4/T.412, the picture and the associated pieces of text are placed into frames F11, F12 and F13 by specifying layout object class.

D.3.2 *Synchronization*

The attribute "synchronization" can be used to lay out margin notes beside the associated text, this is illustrated in Figure D-19/T.412.

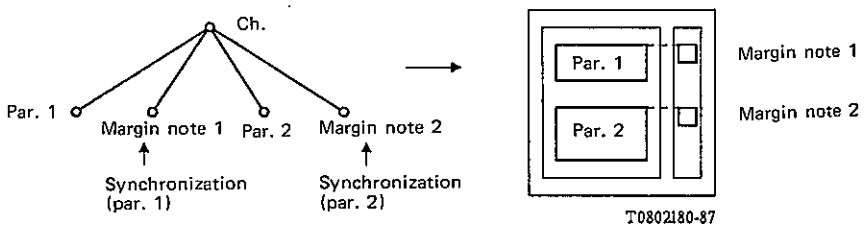


FIGURE D-19/T.412

Illustration of synchronization

D.3.3 *New layout object*

The attribute "new layout object" can be used to constrain a chapter to start on a new page, see Figure D-20/T.412.

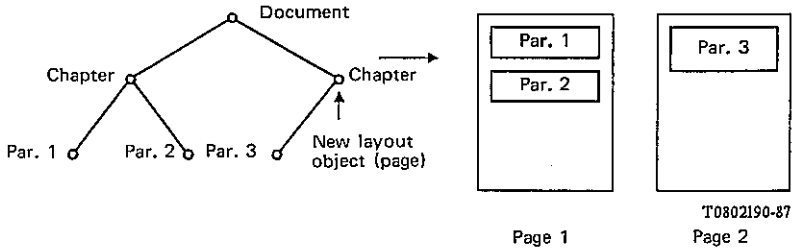


FIGURE D-20/T.412

Illustration of new layout object

D.3.4 *Same layout object*

The attribute "same layout object" can 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 D-21/T.412.

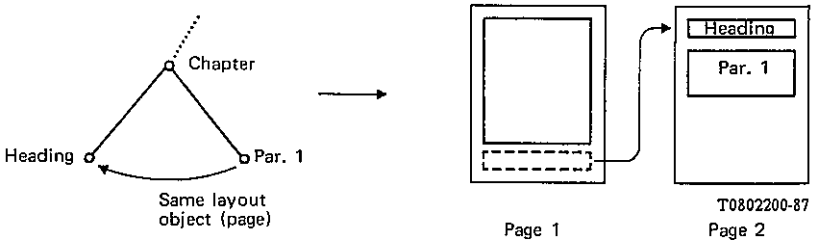


FIGURE D-21/T.412

Illustration of same layout object

D.3.5 *Indivisibility*

The attribute "indivisibility" can 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 D-22/T.412.

In conjunction with the attribute "same layout object" it could be used to specify that a footnote should be placed entirely on the same page as the footnote reference. The use of "same layout object" by itself would only specify that the text of the footnote is to start on the same page.

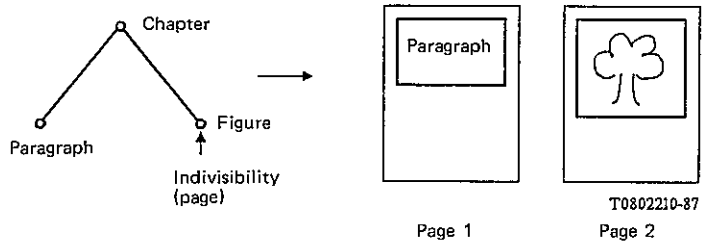


FIGURE D-22/T.412

Illustration of indivisibility

D.3.6 *Balance*

The attribute "balance" can 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 D-23/T.412.

A further example was included in § D.1.6, variation 1.

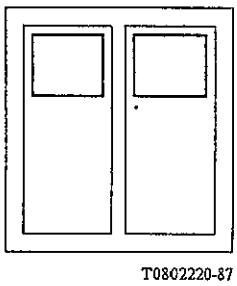


FIGURE D-23/T.412

Illustration of balance

D.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.

D.4.1 *Example 4.1*

Figure D-24/T.412 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 'colourless, opaque';
- block '2' has the texture 'colourless, transparent';

- block '3' has the texture 'colourless, 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'.

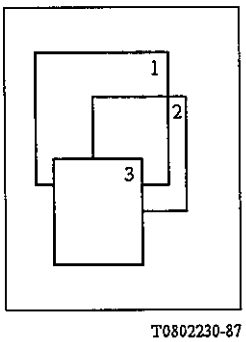


FIGURE D-24/T.412

Illustration of overlapping blocks

D.4.2 *Example 4.2*

Figure D-25/T.412 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 D-25/T.412. The final image is shown in Figure D-26/T.412. In this example, all blocks have the attribute "layout texture" with the value 'colourless, opaque'.

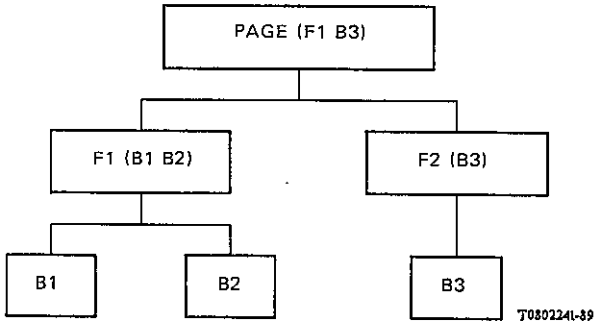


FIGURE D-25/T.412

Sequential layout order

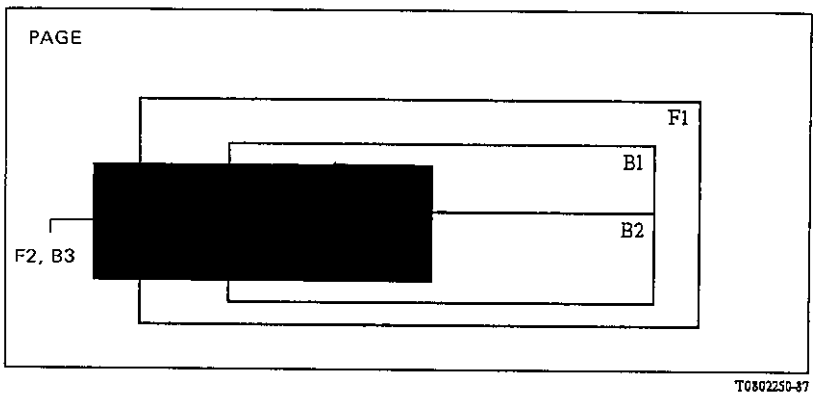


FIGURE D-26/T.412

Final image

D.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 D-27/T.412.

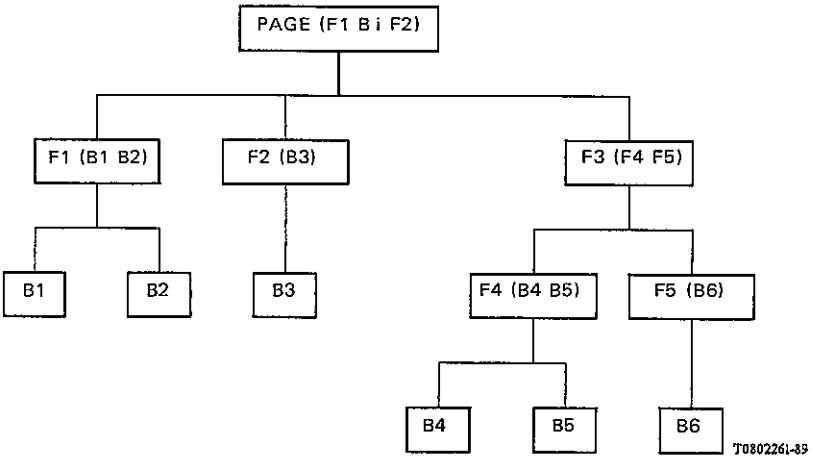


FIGURE D-27/T.412

Sequential layout order

The imaging order either specified explicitly or defined by the sequential layout order is:

- page P: imaging order = F1, 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: 'colourless, transparent';
- blocks B2, B3 and B5: 'colourless, 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 D-28/T.412; to simplify this and subsequent figures, frames are not shown in all cases.

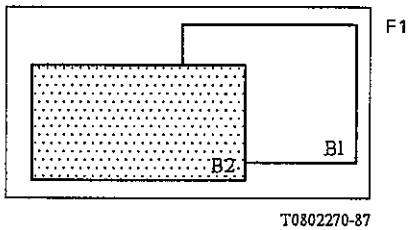


FIGURE D-28/T.412

Image within frame F1

The imaging of the blocks subordinate to F1 is now complete. The next step of the imaging process will consider 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 'colourless, opaque', the content of block B2 is suppressed within the area of intersection with block B3 (see Figure D-29/T.412).

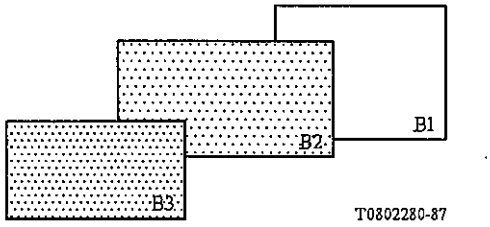


FIGURE D-29/T.412

Imaging of blocks B1, B2, B3

The last branch of the layout structure, the frame F3 must now 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 D-30/T.412. This block does not intersect with other blocks that have been previously imaged.

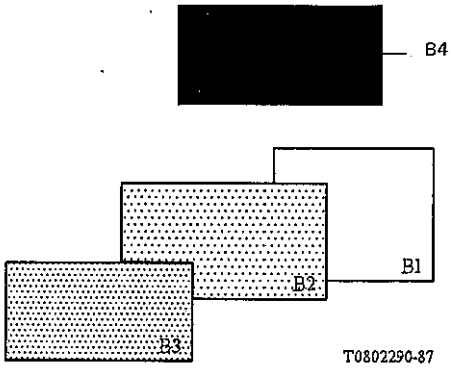


FIGURE D-30/T.412

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 D-31/T.412).

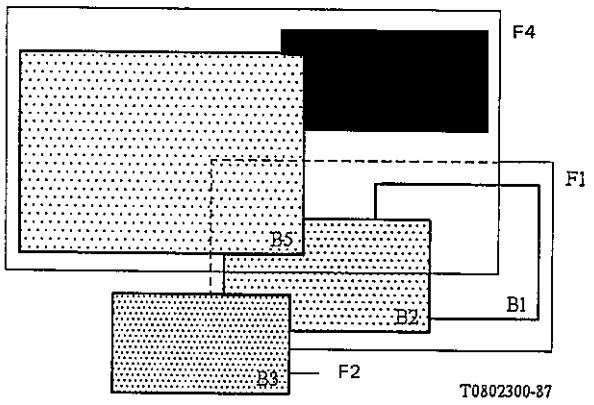


FIGURE D-31/T.412

Image of block B5 added

The final block of the page which must 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 D-32/T.412.

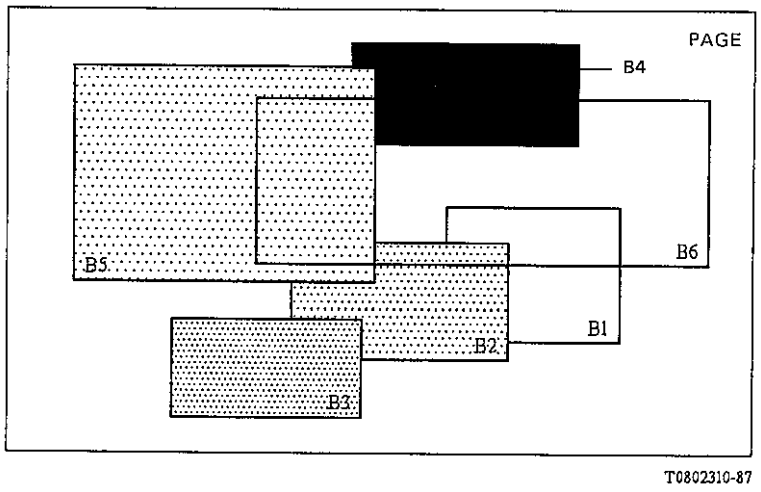


FIGURE D-32/T.412

Final image

D.4.4 *Example 4.4: example of an application*

An example of an application of this feature could be 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.

D.5 *Bindings, content generator*

The attribute "bindings" together with the attribute "content generator" can be used to automatically create numbering schemes for layout and logical objects, for example pages, sections, chapters, footnotes, or other application specific logical structure, for example, "part numbers".

Figure D-33/T.412 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.

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 can 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 such that the first chapter number will be one. The binding name, binding value pair specified at the document logical root is (0, Nr), 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.

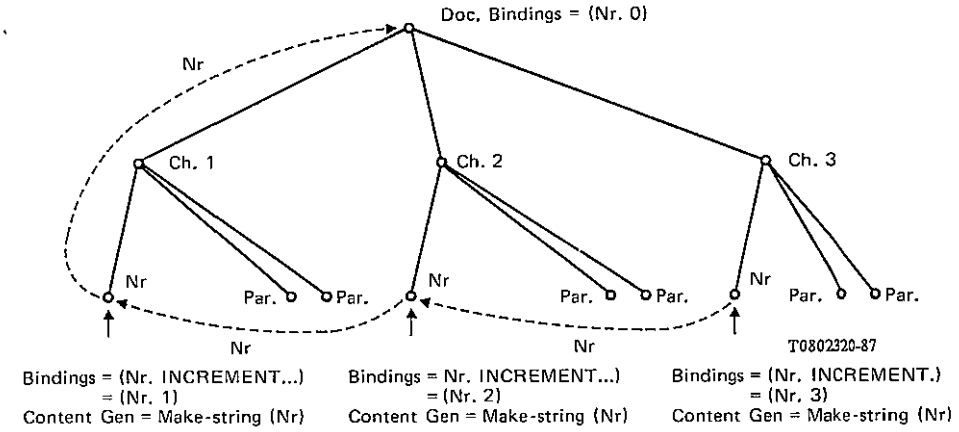


FIGURE D-33/T.412

Use of bindings and content generator for numbering chapters

ANNEX E

(to Recommendation T.412)

(Informative)

The defaulting mechanism

This annex summarises some aspects of the defaulting mechanism.

E.1 *The defaulting mechanism as applicable to each defaultable attribute*

Table E-1/T.412 summarises the steps of the defaulting mechanism described in § 5.1.2.4 as these apply to the various attributes.

TABLE E-1/T.412

Applicable steps for defaulting of attributes

Attribute name	Steps from § 5.1.2.4								
Object type	a		c						
Content architecture class	a		c		e		g	h	j
Content type	a		c		e		g	h	j
User-readable comments	a		c		e				j
Application comments	a		c		e				j
User-visible name	a		c		e				j
Bindings	a		c		e				j
Position	a		c		e		g		j
Dimensions	a		c		e		g	h[a]	j
Border	a	b	c	d	e	f	g	h	j
Balance	a		c		e				j
Layout path	a		c		e		g	h	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
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[b]	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
Indivisibility		b		d		f	g		j
Layout category		b		d		f	g		j
Layout object class		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	
Coding attributes	a		c					h[b]	

a) Only for the case that the attribute applies to a component of object type page. Otherwise the step does not apply.

b) The presentation attributes and coding attributes for which a document application profile may specify new-standard default values are defined in these Recommendations of the T.410 Series that deal with individual content architectures.

E.2 *Determination of content portions and their associated attributes*

E.2.1 *Determination of content associated with basic logical objects*

During the layout process the content associated to 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 to 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, 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 to 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, 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 to 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 to 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 § 5.1.2.5).

5) to 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 the rules defined in steps 1) to 4), using the content portions and/or attribute "content generator" specified by the object class description, as steps 5) to 8), respectively.

9) to 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 the rules defined in steps 1) to 4), using the content portions and/or attribute "content generator" specified by the object class description in the resource-document, as steps 9) to 12), respectively.

13) If no content is determined by the preceding steps 1) to 12), then the content associated with the basic logical object is a "null" string.

   The content portion attributes do not apply.

E.2.2 *Determination of content associated with layout objects*

This is performed partly during the layout process and partly during the imaging process.

E.2.2.1 *Determination during the layout process*

During the layout process the content to be associated to a basic layout object is determined according to the first of the following rules which is applicable:

1) One or more content portions are specified for the object class description used to create the basic layout object, in at least one of these the attribute "content information" is specified.

   In this case, the final determination of the content can be deferred until the imaging process (see § E.1.1.2.2).

2) One or more content portions are specified for the object class description used to create the basic layout object, none of the content portions specifies the attribute "content information". The attribute "content generator" is specified for the object class description.

In this case, the content associated to the basic layout 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 object class description used to create the basic layout object, none of the content portions specifies the attribute "content information". The attribute "content generator" is not specified for the object class description.

In this case, the content associated to the basic layout object is a "null" string.

The content portion attributes do not apply.

4) No content portions are specified for the object class description used to create the basic layout object. The attribute "content generator" is specified for the object class description.

In this case, the content associated to the basic layout 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 § 5.1.2.5).

5) to 8) If the object class description used to create the basic layout object refers to an object class description in the resource-document which either specifies content portions or specifies the attribute "content generator", then apply the rules defined in steps 1) to 4), using the content portions and/or attribute "content generator" specified by the object class description in the resource-document, as steps 5) to 8), respectively.

9) If no content is determined by the preceding steps 1) to 8), then the content associated with the basic layout object is a "null" string.

The content portion attributes do not apply.

During the layout process the content associated with a frame is determined according to the first of the following rules which is applicable:

1) If the attribute "logical source" is specified in the object class description of object type frame then the corresponding logical objects are created and the associated content is evaluated as described in § E.2.1 and laid out within a frame object of this class.

2) If item 1) does not apply but the attribute "logical source" is specified in an object class description in a resource-document referenced by an object class description of object type frame then the corresponding logical objects are created and the associated content is evaluated as described in § E.2.1, and laid out within a frame object of this class.

3) If items 1) and 2) do not apply, layout objects derived from an object class description of object type frame may be used by the layout process to layout content associated with the specific logical structure (and evaluated as described in § E.2.1).

E.2.2.2 *Determination during the imaging process*

During the imaging process the content associated to a basic layout 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 layout object.

In this case, the content associated to the basic layout 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) The basic layout object description refers to an object class description which specifies content portions.

In this case, the content associated to the basic layout object is formed by the concatenation of the strings specified for the attribute "content information" for each of the content portions associated with the object class description, in the sequential order of these content portions.

The content portion attributes are derived from the corresponding content portions.

3) The basic layout object description refers to an object class description which refers to an object class description in the resource-document which specifies content portions.

In this case, the content associated to the basic layout object is formed by the concatenation of the strings specified for the attribute "content information" for each of the content portions associated with the object class description in the resource-document, in the sequential order of these content portions.

The content portion attributes are derived from the corresponding content portions.

4) If no content is determined by the preceding steps 1) to 3), then the content associated with the basic layout object is a "null" string.

ANNEX F

(to Recommendation T.412)

(Informative)

Attribute summary tables

This annex contains summary material describing all attributes defined in the document architecture.

These attributes are summarised in Tables F-1/T.412 and F-2/T.412.

Table F-1/T.412 lists the attributes, references the definition of the attribute and also references other locations containing specification material pertaining to the attribute.

Table F-2/T.412 1ists which attributes can be specified for each of the different types of constituent, and whether the attribute is classified as mandatory, non-mandatory or defaultable. (This table is presented in six parts.)

Table F-3/T.412 lists which layout directive attributes can be applied for each of the types of logical component.

TABLE F-1/T.412

List of attributes of constituents

Attributes of constituents: Attribute name	Specified in §	Reference material in §
Shared attributes		
Identification attributes		
Object type	5.3.1.1	3.1.1, 3.2, 3.3.1
Object identifier	5.3.1.2	
Object class identifier	5.3.1.3	
Construction attributes		
Generator for subordinates	5.3.2.1	3.1.2, 3.5, 6.2.2, A.2.2
Content generator	5.3.2.2	3.5.5, 5.1.3, A.2.4
Relationship attributes		
Object class	5.3.3.1	2.2.3, 2.3.4
Subordinates	5.3.3.2	3.1.1, 3.5.2
Content portions	5.3.3.3	3.1.3, 3.5.5, 6.6
Resource	5.3.3.4	2.3.1, 3.5.8
Presentation style	5.3.3.5	2.3.5, 2.3.9, 5.1.1.4
Content architecture class attributes		
Content architecture class	5.3.4.1	2.2.2
Content type	5.3.4.2	
Miscellaneous attributes		
User-readable comments	5.3.5.1	
Application comments	5.3.5.2	
User-visible name	5.3.5.3	
Bindings	5.3.5.4	5.1.3, A.2.3
Default value lists	5.3.5.5	5.1.2
Layout attributes		
Property attributes		
Positions	5.4.1.1	3.3.2, 6.5
Dimensions	5.4.1.2	6.5.2.2
Border	5.4.1.3	3.3.3, 3.3.5, 6.5, 6.6
Formatting attributes		
Balance	5.4.2.1	6.4.5
Layout path	5.4.2.2	6.5, 6.6
Logical source	5.4.2.3	6.2.1.2, 6.4.4
Permitted categories	5.4.2.4	6.3.1
Imaging attributes		
Imaging order	5.4.3.1	7.1, 7.2
Transparency	5.4.3.2	7.2
Colour	5.4.3.3	7.2
Page position	5.4.3.4	7.3
Medium type	5.4.3.5	7.3

TABLE F-1/T.412 *(cont.)*

List of attributes of constituents

Attributes of constituents: Attribute name	Specified in §	Reference material in §
Presentation attributes	5.4.4	
Logical attributes		
Protection	5.5.1	
Layout style	5.5.2	2.3.5, 2.3.9, 5.1.1.5
Layout style attributes		
Layout style identifier	5.6.1	5.1.1.5
Layout directive attributes		
Block alignment	5.7.1	3.3.3, 6.6
Concatenation	5.7.2	
Fill order	5.7.3	6.6
Indivisibility	5.7.4	6.4.3
Layout category	5.7.5	6.3.1
Layout object class	5.7.6	6.3.2
New layout object	5.7.7	6.4.1
Offset	5.7.8	3.3.3, 6.6
Same layout object	5.7.9	5.1.3, 6.4.2
Separation	5.7.10	3.3.3, 6.6
Synchronization	5.7.11	5.1.3, 6.4.4
Presentations style attributes		
Presentation style identifier	5.8.1	5.1.1.4
Content portion attributes		
Identification attributes		
Content identifier logical	5.9.1	3.4
Content identifier layout	5.9.1	3.4
Common coding attributes		
Type of coding	5.9.2	4
Content information attributes		
Content information	5.9.3.1	2.2.2, 3.1.3, 3.5.5, 6.2.1
Alternative representation	5.9.3.2	6.7
Coding attributes	5.9.4	4

TABLE F-2a/T.412

Attributes which may be specified for constituents shared attributes

Shared attributes Attribute name	Document layout root	Page set	Page (basic)	Page (composite)	Frame	Block	Document logical root	Composite logical object	Basic logical object	Presentation style	Layout style
Object type	M/D	M/D	M/D	M/D	M/D	M/D	M/D	M/D	M/D	--	--
Object identifier	--/M*	--/M	--/M*	--/M*	--/M	--/M*	--/M	--/M	--/M	--	--
Object class identifier	M/--	M/--	M/--	M/--	M/--	M/--	M/--	M/--	M/--	--	--
Generator for subordinates	NM/--	NM/--	--	NM/--	NM/--	--	NM/--	NM/--	--	--	--
Content generator	--	--	NM/--	--	--	NM/--	--	--	NM/NM	--	--
Object class	--/NM	--/NM	--/NM	--/NM	--/NM	--/NM	--/NM	--/NM	--/NM	--	--
Subordinates	--/M*	--/M*	--	--/M*	--/M*	--	--/M	--/M	--	--	--
Content portions	--	--	NM/NM	--	--	NM/NM	--	--	NM/NM	--	--
Resource	NM/--	NM/--	NM/--	NM/--	NM/--	NM/--	NM/--	NM/--	NM/--	--	--
Presentation style	--	--	NM/NM	--	--	NM/NM	--	--	NM/NM	--	--
Content architecture class	--	--	NM/D	--	--	NM/D	--	--	NM/D	--	--
Content type	--	--	NM/D	--	--	NM/D	--	--	--	--	--
User-readable comments	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM	NM
Application comments	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	--	--
User-visible name	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM	NM
Bindings	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	--	--
Default value lists	NM/NM	NM/NM	--	NM/NM	NM/NM	--	NM/NM	NM/NM	--	--	--

M Mandatory
NM Non-mandatory
D Defaultable
-- Not applicable
.../... Object class description/object description
* Exceptional case is specified

TABLE F-2b/T.412

Attributes which may be specified for constituents, layout attributes

Layout attributes Attribute name	Document layout root	Page set	Page (basic)	Page (composite)	Frame	Block	Presentation style
Position	--	--	--	--	NM/D	NM/D	--
Dimensions	--	--	NM/D	NM/D	NM/D	NM/D	--
Border	--	--	--	--	NM/D	NM/D	NM
Balance	NM/D	NM/D	--	NM/D	NM/D	--	--
Layout path	--	--	--	--	NM/D	--	--
Logical source	--	--	--	--	NM/--	--	--
Permitted categories	--	--	--	--	NM/D	--	--
Imaging order	--	--	--	--/NM	--/NM	--	--
Transparency	--	--	NM/D	NM/D	NM/D	NM/D	NM
Colour	--	--	NM/D	NM/D	NM/D	NM/D	NM
Page position	--	--	NM/D	NM/D	--	--	--
Medium type	--	--	NM/D	NM/D	--	--	--
Presentation attributes	--	--	NM/D	--	--	NM/D	NM

M Mandatory
NM Non-mandatory
D Defaultable
-- Not applicable
.../... Object class description/object description

TABLE F-2c/T.412

Attributes which may be specified for constituents, logical attributes

Logical attributes Attribute name	Document logical root	Composite logical object	Basic logical object
Protection	NM/D	NM/D	NM/D
Layout style	NM/NM	NM/NM	NM/NM

NM Non-mandatory
D Defaultable
.../... Object class description/object description

TABLE F-2d/T.412

Attributes which may be specified for constituents, layout style attributes

Layout style attributes Attribute name	Layout style
Layout style identifier	M
User-readable comments	NM
User-visible name	NM
Layout directive attributes	
Block alignment	NM
Concatenation	NM
Fill order	NM
Indivisibility	NM
Layout category	NM
Layout object class	NM
New layout object	NM
Offset	NM
Same layout object	NM
Separation	NM
Synchronization	NM

M Mandatory
NM Non-mandatory

TABLE F-2e/T.412

Attributes which may be specified for constituents, presentation style attributes

Presentation style attributes Attributes name	Presentation style
Presentation style identifier	M
User-readable comments	NM
User-visible name	NM
Presentation attributes	NM
Transparency	NM
Colour	NM
Border	NM

M Mandatory
NM Non-mandatory

TABLE F- 2f/T.412

Attributes which may be specified for constituents, content portion attributes

Content portion attributes Attribute name	Content portion
Content identifier - logical	NM
Content identifier - layout	NM
Type of coding	D
Content information	NM
Alternative representation	NM
Coding attributes	*

NM Non-mandatory
D Defaultable
* Classification defined in each content architecture

TABLE F-3/T.412

Layout directive attributes which nay be applied to logical components

Layout directives Attribute name	Document logical root	Composite logical object	Basic logical object
Layout directive attributes			
Block alignment	--	--	NM/D
Concatenation	--	--	NM/D
Fill order	--	--	NM/D
Indivisibility	--	NM/D	NM/D
Layout category	--	--	NM/D
Layout object class	NM/D	NM/D	NM/D
New layout object	--	NM/D	NM/D
Offset	--	--	NM/D
Same layout object	--	NM/D	NM/D
Separation	--	--	NM/D
Synchronization	--	NM/D	NM/D

NM Non-mandatory
D Defaultable
.../... Object class description/object description
-- Not applicable

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	**Terminals for telematic services**
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland

Geneva, 2008