INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

BLUE BOOK

VOLUME VII — FASCICLE VII.6

TERMINAL EQUIPMENT AND PROTOCOLS FOR TELEMATIC SERVICES

RECOMMENDATIONS T.400-T.418

IXTH PLENARY ASSEMBLY
MELBOURNE, 14-25 NOVEMBER 1988

INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

BLUE BOOK

VOLUME VII — FASCICLE VII.6

TERMINAL EQUIPMENT AND PROTOCOLS FOR TELEMATIC SERVICES

RECOMMENDATIONS T.400-T.418

IXTH PLENARY ASSEMBLY
MELBOURNE, 14-25 NOVEMBER 1988

CONTENTS OF THE CCITT BOOK
APPLICABLE AFTER THE NINTH PLENARY ASSEMBLY (1988)

BLUE BOOK

Volume X

/

CONTENTS OF FASCICLE VII.6 OF THE BLUE BOOK

Recommendations T.400 to T.418

Terminal equipment and protocols for telematic services

REMARK

The terminology on Document Architecture has evolved during the elaboration of Recommendations during the study period 1985-1988.

In order to prevent the readers from misinterpretations, *Document Architecture Transfer and Manipulation (DATAM)* is the general name for the T.400 series.

Document Transfer and Manipulation (DTAM) is relevant to the T.430 series.

PRELIMINARY NOTE

In this Fascicle, the expression "Administration" is used for shortness to indicate both a telecommunication Administration and a recognized private operating agency.

FASCICLE VII.6

Recommendations T.400 to T.418

TERMINAL EQUIPMENT AND PROTOCOLS FOR TELEMATIC SERVICES

PAGE INTENTIONALLY LEFT BLANK

PAGE LAISSEE EN BLANC INTENTIONNELLEMENT

INTRODUCTION TO DOCUMENT ARCHITECTURE, TRANSFER AND MANIPULATION

CONTENTS

Introduction

Recommendation T.400 is one of a set of the T series of Recommendations to facilitate the interconnection of telematic systems and terminals. It is related to the other CCITT Recommendations in the set as defined by the Reference Model for OSI (X.200) and also makes provision for non-X.200 existing telematic services.

1   Scope

Recommendation T.400 serves as an introduction and guide to Recommendations in the T.400 series, and associated Recommendations in the T.500 series that make use of the T.400 series.

2   The T.400 series and the related T.500 series of Recommendations

Recommendations in the T.400 and T.500 series set the grounds for the architecture, the transfer and the manipulation of documents in current and future telematic services.

2.1   *Document architecture, transfer and manipulation*

The T.400 series of Recommendations specifies how documents are to be described and which are the rules for transferring and manipulating such documents.

2.1.1   *T.410 series of Recommendations: Open document architecture (ODA) and interchange format*

The T.410 series of Recommendations defines the document architecture, i.e. rules for structuring documents, and the interchange format, i.e. rules for representing documents. It also gives rules for defining document application profiles, i.e. the specifications of the features of a document to be used in an application.

2.1.2 *T.430 series of Recommendations: Document transfer and manipulation (DTAM)*

The T.430 series of Recommendations defines document transfer services and protocols. It also gives rules for defining communication application profiles, i.e. the specifications of the communication support functions, service classes and functional units to be used in a communication applications.

These Recommendations make use, in turn, of functions defined by other Recommendations in the OSI environment such as those in the X.400 series and in the X.200 series. Annex B of this Recommendation shows how the T.410 series relates to the X.400 and X.200 series in the OSI environment.

Annex E of Recommendation T.411 describes the use of the X.400 series of Recommendations to transfer documents conforming to the T.410 series.

The interchange format defined in the T.410 series constitutes an application context as defined in Recommendation X.226. The abstract syntax of the application context is defined in Recommendation T.415.

2.1.3 *Recommendation T.441: Document transfer and manipulation (DTAM) - Operational structure*

Recommendation T.441 defines the concept of the operational structure, which enables the manipulation of documents or parts of documents, and gives an outline of the principles and use of the operational structures.

2.2 *Application profiles*

Each Recommendation in the series T.500 through T.559 specifies either a document application profile based on the rules of T.411, or a communication application profile based on the rules of T.431, or an operational application profile based on the rules of T.441.

2.3 *Terminal and gateway characteristics*

Each Recommendation in the series T.560 through T.599 specifies terminal characteristics or gateway characteristics for equipments using application profiles in the T.500 through T.559 series.

3   Reference list of the T.400 and T.500 series of Recommendations

- Rec. T.400 (1988): Introduction to document architecture, transfer and manipulation

- Rec. T.411 (1988): Open document architecture (ODA) and interchange format - Introduction and general principles

- Rec. T.412 (1988): Open document architecture (ODA) and interchange format - Document structures

- Rec. T.414 (1988): Open document architecture (ODA) and interchange format - Document profile

- Rec. T.415 (1988): Open document architecture (ODA) and interchange format - Open document interchange format (ODIF)

- Rec. T.416 (1988): Open document architecture (ODA) and interchange format - Character content architectures

- Rec. T.417 (1988): Open document architecture (ODA) and interchange format - Raster graphics content architectures

- Rec. 418 (1988): Open document architecture (ODA) and interchange format - Geometric graphics content architectures

- Rec. 431 (1988): Document transfer and manipulation (DTAM) - Services and protocols - Introduction and general principles

- Rec. 432 (1988): Document transfer and manipulation (DTAM) - Services and protocols - Service definition

- Rec. 433 (1988): Document transfer and manipulation (DTAM) - Services and protocols - Protocol specification

- Rec. 441 (1988): Document transfer and manipulation (DTAM) - Operational structure

Associated with the T.400 series, the following Recommendations specify application profiles:

- Rec. 501 (1988): Document application profile MM for the interchange of formatted mixed mode documents

- Rec. 502 (1988): Document application profile PM1 for the interchange of Processable form documents

- Rec. 503 (1988): Document application profile for the interchange of group 4 facsimile documents

- Rec. 504 (1988): Document application profile for the videotex interworking

- Rec. 521 (1988): Communication application profile BT0 for document bulk transfer based on the session service (according to the rules defined in T.62 *bis*)

- Rec. 522 (1988): Communication application profile BT1 for document bulk transfer

- Rec. 523 (1988): Communication application profile DM1 for videotex interworking

- Rec. 541 (1988): Operational application profile for videotex interworking

Terminal and gateway characteristics based on the application profile Recommendations are in the following Recommendations:

- Rec. 561 (1988): Terminal characteristics for mixed mode of operation MM

- Rec. 562 (1988): Terminal characteristics for teletex processable mode PM1

- Rec. 563 (1988): Terminal characteristics for group 4 facsimile apparatus

- Rec. 564 (1988): Gateway characteristics of videotex interworking

ANNEX A

(to Recommendation T.400)

(informative)

Overview of the T.400 and T.500 series of Recommendations


```
                          ┌─────────────────────────┐
                          │ T.400                   │
                          │ Introduction to document│
                          │ architecture, transfer  │
                          │ and manipulation        │
                          └─────────────────────────┘
```

T.411 — T.410 Series — Open document architecture (ODA) & interchange format

T.441 — Document architecture, operations

T.431 — T.430 Series — Document transfer and manipulation (DTAM)

T.501 — T.500 Series — Document application profile

T.541 — Operational application profile for videotex interworking

T.521 — T.520 Series — Communication application profile

T.561 — T.560 Series — Terminal & gateway characteristics

T0801111-89

FIGURE A-1/T.400

ANNEX B

(to Recommendation T.400)

(informative)

The T.400 series on the X.200 OSI environment

FIGURE B-1/T.400
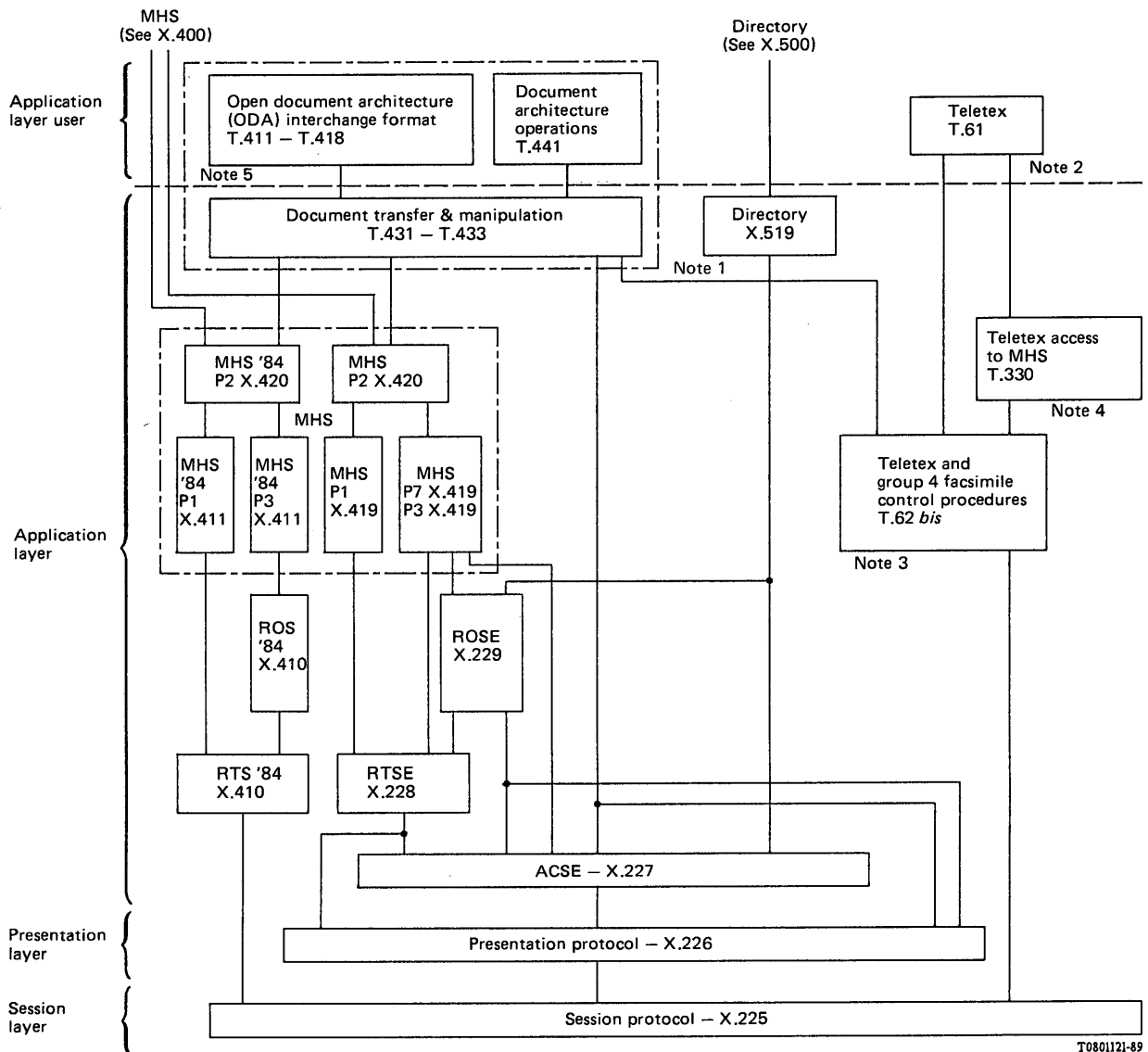
Note 1 - The use of ROSE and RTSE in the T.400 series is for further study.

Note 2 - Character repertoire definition of Recommendation T.61 only.

Note 3 - T.62 *bis*. Together with the relevant service and protocol elements of Recommendations X.215/X.225, are intended to be equal to Recommendation T.62 (*Red Book*), Volume VII - Fascicle VII.3.

Note 4 - Recommendation T.330, telematic access to interpersonal messaging system (IPMS) describes group 4 facsimile and teletex access to IPMS in the T.62 *bis*/X.225 environment.

Note 5 - Recommendation T.411 - Annex E describes the use of MHS to transfer documents conforming to the T.410 series.

OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT

INTRODUCTION AND GENERAL PRINCIPLES

CONTENTS

Introduction

Recommendation T.411 is one of the T.410 series of Recommendations which has been prepared by CCITT COM VIII/WP 2 for the interchange of documents in the context of document architecture transfer, and manipulation.

This series of Recommendations specifies a document architecture (known as the "Open Document Architecture" ODA) and an interchange format (known as the "Open document interchange format" ODIF).

The T.410 series of Recommendations consists of:

- Rec. T.411: Open document architecture (ODA) and interchange format - Introduction and general principles;

- Rec. T.412: Document structures;

- Rec. T.414: Document profile;

- Rec. T.415: Open document interchange format (ODIF);

- Rec. T.416: Character content architectures;

- Rec. T.417: Raster graphics content architectures;

- Rec. T.418: Geometric graphics content architectures.

Note - At present, there is no Recommendation T.413

Other Recommendations may be added to this series.

The T.410 series of Recommendations supersedes Recommendation T.73 (1984)

Development of the T.410 series has been in parallel with:

- ECMA-101: "Office document architecture";

- ISO 8613: "Information processing - Text and office systems - Office document architecture (ODA) and interchange format".

1   Scope

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.

1.2   The T.410 series applies to the interchange of documents by means of data communications.

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.

1.3   This Recommendation:

-   introduces the T.410 series as a whole;

-   gives the references necessary for all the Recommendations pertaining to this series;

-   defines terms used in the context of the T.410 series;

-   presents the concepts of the document architecture;

-   gives an overview of all the Recommendations in the T.410 series;

-   describes the inter-dependencies of these Recommendations;

-   defines conformance to the T.410 series;

-   gives rules for defining document application profiles.

Annex A lists other Recommendations, standards and registers to which references are made, but that are not indispensable for the application of the T.410 series.

Annex B describes the relationships of the T.410 series with other Recommendations and standards.

Annex C describes the correspondence between Recommendation T.73 and the T.410 series of Recommendations.

Annex D establishes the Principles for the assignment of ASN.1 object identifier values in the T.410 series of Recommendations.

Annex E defines the use of MHS to interchange documents conforming to the T.410 series of Recommendations.

2   **Normative references**

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of the T.410 series of Recommendations. 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 edition of the Recommendations and standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

The following references are applicable to the whole of the T.410 series of Recommendations.

-   Rec. T.4 (1984): Standardization of group 3 facsimile apparatus for document transmission

-   Rec. T.6 (1984): Facsimile coding schemes and coding control functions for group 4 facsimile apparatus

-   Rec. T.50 (1984): International alphabet No. 5

-   Rec. T.60 (1984): Terminal equipment for use in the teletex service

-   Rec. T.101 (1984): International interworking for videotex services

-   Rec. X.208 (1988): Specification of abstract syntax notation one (ASN.1)

-   Rec. X.209 (1988): Specification of basic encoding rules for abstract syntax notation one (ASN.1)

-   Rec. T.412 (1988): Open document architecture (ODA) and interchange format - Document structures

- Rec. T.414 (1988): Open document architecture (ODA) and interchange format - Document profile

- Rec. T.415 (1988): Open document architecture (ODA) and interchange format - Open document interchange format (ODIF)

- Rec. T.416 (1988): Open document architecture (ODA) and interchange format - Character content architectures

- Rec. T.417 (1988): Open document architecture (ODA) and interchange format - Raster graphics content architectures

- Rec. T.418 (1988): Open document architecture (ODA) and interchange format - Geometric graphics content architectures

- ISO 2022 (1986): Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques

- ISO 6429 (1983): Information Processing - ISO 7-bit and 8-bit coded character sets - Additional control functions for character-imaging devices[1]

- ISO 6937-1 (1983): Information processing - Coded character sets for text communication - Part 1: General introduction

- ISO 6937-2 (1983): Information Processing - Coded character sets for text communication - Part 2: Latin alphabetic and non-alphabetic graphic characters

- ISO 6937-3 (1983): Information processing - Coded character sets for text communication - Part 3: Control functions for page-image format[1]

- ISO 8601: Data elements and interchange format - Information interchange - Representation of dates and times[2]

- ISO 8632-1 (1987): Information processing systems - Computer graphics - Metafile for the storage and transfer of picture description information (CGM) - Part 1: Functional specification

- ISO 8632-3 (1987): Information processing systems - Computer graphics - Metafile for the storage and transfer of picture description information (CGM) - Part 3: Binary encoding

- ISO 9541-1: Information processing - Font and character information interchange - Part 1: introduction (2)

- ISO 9541-2: Information processing - Font and character information interchange - Part 2: Registration and naming procedures[2]

- ISO 9541-5: Information processing - Font and character information interchange - Part 5: Font attributes and character model[2]

- ISO 9541-6: Information processing - Font and character information interchange - Part 6: Font and character attribute subsets and applications[2]

3 Definitions

For the purposes of the T.410 series of Recommendations, the following definitions apply:

3.1 active position

The point at which the action specified by the next character is to be effected.

---

[1] Under revision.
[2] To be published.

3.2 aligned around

A tabulation alignment that positions the sequence of character images for a specified character string such that the position point of the character image of the first instance of a specified group of characters within that string is positioned at the tabulation stop.

3.3 aspect ratio

The ratio of the dimension of a pel array in the direction of the pel path to the dimension on the direction of the line progression.

3.4 assured reproduction area

The rectangular area that remains on the nominal page after deducting an agreed allowance for edge losses.

3.5 attribute

An element of a constituent of a document that has a name and a value and that expresses a characteristic of this constituent or a relationship with one or more constituents.

3.6 available area

The area determined by the document layout process into which the content portion is formatted by the content layout process.

3.7 basic component

A basic logical or layout object, or an object class from which basic logical or layout objects may be derived.

3.8 basic layout object

An object in the specific layout structure that has no subordinate.

3.9 basic logical object

An object in the specific logical structure that has no subordinate.

3.10 basic measurement unit (BMU)

A unit of linear measurement equal to 1/1200 of 25.4 mm.

Note - A locally defined scaling factor may be used to map the document to a particular imaging device.

3.11 basic value

An attribute value, a control function parameter value or the value of any other capability that is unconditionally allowed in document interchange in the context of a given document application profile.

3.12 binding

A pair comprising an identifier and a value, where the value may be of any type, may be specified by an expression, and is accessed through use of the binding identifier.

3.13 block

A type of basic layout component that corresponds to a rectangular area within a frame or a page.

3.14 bottom edge

The edge of the positioning area of a basic layout object that is in the direction of the line progression.

3.15 bottom left corner

The corner of a layout object that is least progressed in the horizontal direction and most progressed in the vertical direction of this layout object.

3.16 bottom right corner

The corner of a layout object that is most progressed both in the horizontal and vertical directions of this layout object.

3.17 centred

1) The result of a layout or imaging process that positions the sequence of character images for a line such that the distance from the line home position to the position point of the first character image is approximately equal to the distance from the escapement point of the last character image to the end edge of the positioning area.

2) A tabulation alignment that positions the sequence of character images for a specified character string such that the distance from the position point of the first character image to the tabulation stop is approximately equal to the distance from the tabulation stop to the escapement point of the last character image.

Note – The term "centred" is also used in the parameter "alignment" of the attribute "position" and in the attribute "block alignment".

3.18 character

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

3.19 character base line

A line across a character image, in the horizontal direction when the character image is in its intended viewing orientation.

3.20 character image

The human perceptible rendering of a character on a presentation medium.

3.21 character orientation

The direction of the character base line relative to the character path.

3.22 character path

The direction of progression of successive character images within a line box.

3.23 character sequence

A sequence of characters intended to be presented as one or more lines.

3.24 characters spacing (for constant spacing fonts only)

The distance between the position points of successive character images when the inter-character space equals zero.

3.25 clipped array

The actual pel array to be imaged as determined by taking account of all clipping parameters.

3.26 complete generator set

A constituent of a document consisting of a document root object class description and at least one level of subordinate object class descriptions which are used to control the creation and/or modification of the set of object descriptions representing a corresponding specific structure.

3.27 component

An object or an object class.

3.28 composite component

A composite logical or layout object, or an object class from which composite logical or layout objects may be derived.

3.29 composite layout object

An object in the layout structure that has one or more subordinate objects.

3.30 composite logical object

An object in the logical structure that has one or more subordinate objects.

3.31 constant spacing

The characteristic of a font wherein the distance between the position point and the escapement point is the same for all character images.

3.32 constituent

A set of attributes that is one of the following types: a document profile, an object description, an object class description, a presentation style, a layout style or a content portion description.

3.33 content

The information conveyed by the document, other than the structural information, and that is intended for human perception.

3.34 content architecture

Rules for defining the internal structure and representation of the content of basic components in terms of a set of content elements, attributes and control functions, and guidelines for the presentation of the content.

3.35 content architecture class

The rules for defining the internal structure and representation of the content of basic components in one of a set of forms defined for each type of content element.

Note – Examples of content architecture classes are formatted form, processable form and formatted processable form in the case of character content elements.

3.36 content architecture level

An identified subset of the features pertaining to a content architecture class.

3.37 - content editing process

The process that creates new content or modifies previous content.

3.38 content element

A basic element of the content of a document.

3.39 content layout process

The process that, interacting with the document layout process, consists of the formatting of content portions into available areas and the determination of the sizes of blocks in accordance with information contained in the presentation styles.

3.40 content portion

The result of partitioning the content of a document according to its logical and/or layout structure.

3.41 content portion description

A constituent of a document, representing a content portion that consists of content information and attributes to specify the properties of its content information.

3.42 content type

A category of content elements such as graphic characters, raster graphic elements and geometric graphic elements.

3.43 control function

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

Note – Examples of control functions are Select Graphic Rendition (SGR) in character content architectures and Set Line Type in geometric graphics content architectures.

3.44 current layout position

The identification of a lowest level frame which is maintained during the layout process for each layout stream which occurs.

3.45 data structure

A set of data items and the relationship among them representing the whole or a part of a constituent.

Note – The data items constituting a data structure represent attributes of the document, the document profile, the component, the style or the content portion concerned.

3.46 description

A constituent that corresponds to a structural element.

3.47 descriptor

A data structure representing the document profile, an object class description, a layout style, a presentation style or an object description.

3.48 document

A structured amount of information intended for human perception, that can be interchanged as a unit between users and/or systems.

3.49 document application profile

The specification of a combination of features defined in the T.410 series, intended to form a subset to fulfil the requirements of an application.

3.50 document architecture

1) Rules for defining the structure of documents, in terms of a set of components and content portions, and the representation of documents in terms of constituents and attributes.

2) The structural information of a document consisting of the set of one or more of the following structures: specific logical structure, specific layout structure, generic logical structure and/or generic layout structure.

3.51 document architecture class

The rules for defining the structure and representation of documents in formatted form, processable form or formatted processable form.

3.52 document architecture level

An identified subset of the features pertaining to a document architecture class.

3.53 document body

The part of a document that may include a generic logical and layout structure, specific logical and layout structure, layout and presentation styles but excludes the document profile.

3.54 document class

A set of logical object class descriptions, layout object class descriptions, generic content portion descriptions, styles and a document profile, that specifies a set of documents with common characteristics.

3.55 document class description

The specification of a document class.

3.56 document layout process

The process that creates a specific layout structure in accordance with the generic layout structure and information contained in the specific logical structure, the generic logical structure and the layout styles.

3.57 document layout root

The composite object of the specific layout structure at the highest level of the hierarchy.

3.58 document logical root

The composite object of the specific logical structure at the highest level of the hierarchy.

3.59 document profile

A set of attributes which specifies the characteristics of the document as a whole.

3.60 document profile level

An identified subset of the features pertaining to the document profile.

3.61 editing process

The stage of a document processing that consists of the content editing process and the logical structure editing process.

3.62 end-aligned

1) The result of a layout or imaging process that positions the sequence of character images for a line such that the escapement point of the last character image is positioned at the end edge of the positioning area.

2) A tabulation alignment that positions the sequence of character images for a specified character string such that the escapement point of the last character image is positioned at the tabulation stop.

3.63 end edge

The edge of the positioning area of a basic layout object that is in the direction of the character path.

3.64 escapement point

A reference point associated with a character image that is used for positioning of the next character image.

3.65 external document class

A document class referred to by the document profile of an interchanged document containing no generic structure.

3.66 factor set

One or more object class descriptions which are used to factorise the attributes of object descriptions representing a specific structure.

3.67 filing

The storage of a document according to some defined method in order to facilitate retrieval of the document.

3.68 font

A set of character images normally with a common design and size.

3.69 font size

The height of the character images in a font.

3.70 formatted form

A form of representation of a document that allows the presentation of the document as intended by the originator and that does not support editing and (re)formatting.

3.71 formatted processable form

A form of representation of the document that allows presentation of the document as intended by the originator and also supports editing and (re)formatting.

3.72 formatting

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

3.73 frame

A type of composite layout component that corresponds to a rectangular area within a page or another frame.

3.74 generic content portion

A content portion associated with an object class.

3.75 generic content portion description

A content portion description associated with an object class description.

3.76   generic-document

A structured amount of information intended for the interchange of generic structures, and optionally associated styles and content portions, for use in the processing of interchanged documents.

3.77   generic layout structure

A set of layout object classes and associated generic content portions.

3.78   geometric logical structure

A set of logical object classes and associated generic content portions.

3.79   geometric graphic element

A graphic element used to describe an image by geometric graphical means.

Note - Geometric graphic elements include those describing primitive geometric shapes such as points, arcs, lines.

3.80   graphic character

A member of a set of graphic symbols used for the representation of information.

Note - Graphic characters include simple alphanumeric characters (for example, accented letters) and pictorial characters (for example, mosaics).

3.81   graphic element

A content element that is capable of having a visual representation.

Note - Three types of graphic elements are distinguished in the T.410 series of Recommendations: graphic characters, geometric graphics elements and raster graphics elements.

3.82   hard line terminator

A line terminator that is intended not to be removed in a re-formatting process.

3.83   horizontal direction (of a layout object)

The direction in a layout object relative to which content architectures may define attributes determined using the horizontal axis of the page.

3.84   imaging order

The order of precedence of layout objects for imaging in the layout object to which they are immediately subordinate.

3.85   imaging process

The process of producing a document on a presentation medium in human perceptible form, making use of the document profile, specific and generic layout structures, presentation styles and content portions.

3.86   indentation

The result of a layout or imaging process that causes the sequence of character images for a line to begin at a distance from the line home position in the direction of the character path.

3.87   initial point

1)   The point associated with a basic layout object relative to which all line boxes imaged within that basic layout object are positioned (character content architectures Recommendation T.416).

2)   The point associated with a basic layout object relative to which all pels imaged within that basic layout object are positioned (raster graphics content architectures Recommendation T.417).

3.88   interchange

The process of transferring a document from an originating system to a receiving system.

3.89   interchange data element

A data structure representing a constituent of a document.

3.90 interchange format

The rules for representing a document for the purpose of interchange.

3.91 interchange format class

A form of interchange format suitable to a specific application.

Note - In the T.410 series of Recommendations, the defined classes differ by the ordering of the interchange data elements or by the coding.

3.92 inter-character space

An additional amount of spacing that is included between adjacent character images.

3.93 intersection

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

3.94 item identifier

A string of characters preceding the first line of characters in a content portion that is used to identify the subsequent text.

3.95 justified

The result of a layout or imaging process that varies the width of the space character and/or the inter-character space to produce a simultaneously start-aligned and end-aligned presentation of the text.

3.96 kern

The part of a character which extends beyond its position point or escapement point.

3.97 layout category

The association of basic logical objects with lowest level frames such that the content of these basic logical objects is placed in the appropriate frames.

3.98 layout object

An element of the specific layout structure of a document, for example, page, block.

3.99 layout object class

An element of the generic layout structure from which a set of layout objects with common characteristics may be derived, for example, pages with common headers and footers.

3.100 layout process

The stage of a document processing that consists of the document layout process and the content layout process.

Note - This is also referred to as formatting.

3.101 layout stream

A set of basic logical objects pertaining to the same layout category.

3.102 layout structure

1) The result of dividing and subdividing the content of a document into increasingly smaller parts, on the basis of the presentation, for example, into pages, blocks.

2) All layout objects and associated content portions forming the layout hierarchy of a document.

3.103 layout style

A constituent of the document, referred to/from a logical component, that guides the creation of a specific layout structure.

3.104 leading edge

The edge of a frame or block that is orthogonal to the direction of the layout path and that is met first, from the outside of the frame or the block, in the opposite direction of the layout path.

3.105 left hand edge

The edge of a frame or block that is parallel to the direction of the layout path and that is met first, from the outside of the frame or the block, in the direction at an angle of 270° counterclockwise relative to the direction of the layout path.

3.106 line box

A rectangular area within which a sequence of character images are positioned.

3.107 line home position

The point within a line box that is used for positioning that line box.

3.108 line progression

1) The direction of progression of successive line boxes within a basic layout object (character content architectures Recommendation T.416).

2) the direction of progression of successive lines of pels within a basic layout object (raster graphics content architectures Recommendation T.417).

3.109 lines spacing

1) The distance between two adjacent reference lines within a basic layout object (character content architectures Recommendation T.416).

2) The distance between two adjacent lines of pels within a basic layout object (raster graphics content architectures Recommendation T.417).

3.110 line terminator

A control function or combination of control functions that indicates the end of a line or the end of a character sequence.

3.111 logical object

An element of the generic logical structure of a document which may have a meaning that is significant to the application or user, for example, chapter, section, paragraph.

3.112 logical object class

An element of the generic logical structure from which a set of logical objects with common characteristics may be derived, for example, composite logical objects representing sections with a common internal structure.

3.113 logical structure

1) The result of dividing and subdividing the content of a document into increasingly smaller parts, on the basis of the human-perceptible meaning of the content, for example, into chapters, sections, paragraphs.

2) All logical objects and associated content portions representing the logical hierarchy of a document.

3.114 logical structure editing process

The process that creates a new specific logical structure or modifies a previous specific logical structure and allocates or re-allocates content to basic logical objects.

3.115 mandatory attribute

An attribute which, when applicable to a constituent, must be specified explicitly in that constituent.

3.116 nominal page

A rectangular area which, as assumed by the sender of a document, has the ideal size of the presentation surface.

Note - Examples of ideal sizes are given in ISO 216.

3.117 non-basic

A qualifier for attribute values, control function parameter values and other capabilities that are only allowed in document interchange in the context of a given document application profile if their use is declared in the document profile.

3.118 non-mandatory attribute

An attribute which, when applicable to a constituent, need not be specified explicitly; if the attribute is not specified explicitly in a given constituent, the attribute does not apply.

3.119 object

An element of the specific layout structure or of the specific logical structure.

3.120 object class

An element of a generic structure from which objects with common characteristics may be derived.

3.121 object class description

A set of attributes that specify the properties of an object class including its relationships, if any, with other components.

3.122 object description

A set of attributes that specify the properties of an object including its relationships, if any, with other components.

3.123 object type

A property of every component that specifies which attributes are permitted in the description to which it applies and indicates the role of the component in the document architecture.

3.124 orphan

One or more lines of text that is associated with subsequent text but isolated from it by a page or column boundary.

3.125 overhang

The result of a layout or imaging process that positions the sequence of character images for a line to begin at a distance from the line home position in the direction opposite to the character path.

3.126 page

A layout component that corresponds to a rectangular area used for presenting the content of the document.

3.127 page coordinate system

An orthogonal co-ordinate system whose origin is the top left corner of the page; its horizontal axis and its vertical axis coincide with the top edge and the left edge of the page, respectively.

3.128 page set

A composite layout component that represents a collection of pages or further page sets.

3.129 pairwise kerning

The distance between two adjacent character images depending on the combination of the two characters together rather than separately.

3.130 parallel annotation

Two sequential character strings that are presented in parallel, wherein the second string is used to indicate the pronunciation or interpretation of the first string.

3.131 partial generator set

A collection consisting of hierarchically related object class descriptions which are used to guide the creation of hierarchically related corresponding object descriptions but does not fully specify all specific structures that may be created.

3.132 pel array

A two-dimensional array of pels used to represent a pictorial image.

3.133 pel path

The direction of progression of successive pels along a line within a basic layout object.

3.134 pel spacing

The distance between any two successive pels along a line within a basic layout object.

3.135 picture element (pel)

The smallest graphic element that can be independently addressed within a picture; (an alternative term for raster graphics element).

3.136 positioning area

The rectangular area within a basic layout object within which the position points and the escapement points of all character images are located.

3.137 position point

The point relative to which the character image is placed i.e. the character is imaged with the position point at the active position.

3.138 presentation

The operation of rendering a document in a form perceptible to a human being.

3.139 presentation medium

The carrier of information in a form perceptible to a human being.

3.140 presentation style

A constituent of the document, referred to from a basic logical or layout component, which guides the format and appearance of the document content.

3.141 presentation surface

A two-dimensional presentation medium (such as paper, film, video display screen) on which the formatted form of a document may be displayed for human viewing.

3.142 processable form

A form of representation of a document that allows editing and formatting.

3.143 processing

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

3.144 raster graphics element

An alternative term for a picture element (pel).

3.145 reformatting

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

3.146 reference area

A rectangular area within a basic layout object, with its side equal to the pel spacing and the line spacing, within which the main part of a pel is imaged.

3.147 reference line

A line through the line home position and parallel to the character path.

3.148 reference point

The point at the corner of the reference area situated in the opposite direction of both pel path and line progression and which is used for positioning a pel.

3.149 region of interest

A rectangular area within a virtual device coordinate space, with sides which are parallel to the axes of its coordinate system, surrounding those (parts of) geometric graphics elements that are intended to be imaged.

3.150 resource document

A generic-document containing one or more object class descriptions referred to by one or more object class descriptions of another document.

3.151 retrieval

The recovery of previously filed information.

3.152 right hand edge

The edge of a frame or block that is parallel to the direction of the layout path and that is met first, from the outside of the frame or the block, in the direction at an angle of 90° counterclockwise relative to the direction of the layout path.

3.153 scaled measurement unit (SMU)

A unit of linear measurement used for positioning and dimensioning layout objects and content elements on a presentation surface, its value being equal to the basic measurement unit (BMU) times the unit scaling.

3.154 sequential layout order

The sequential order in which layout objects are to be imaged when an imaging order is not specified.

3.155 sequential logical order

The sequential order in which logical objects are to be processed by the layout process.

3.156 sequential order

A convention for ordering the objects in a structure such that each object is succeeded by all of its immediate subordinates, before any other object with the same immediate superior, i.e. in order tree traversal.

3.157 soft line terminator

A line terminator that is allowed to be removed, relocated or replaced in a subsequent layout process.

3.158 spacing ratio

The ratio of line spacing to pel spacing.

3.159 specific layout structure

A set of layout objects and associated content portions.

3.160 specific logical structure

A set of logical objects and associated content portions.

3.161 start-aligned

1) The result of a layout or imaging process that positions the sequence of character images for a line such that the position point of the first character image of that sequence is positioned on the line home position or at the point specified by the first line indentation or overhang if any.

2) A tabulation alignment that positions the sequence of character images for a specified character string such that the position point of the first character image of that sequence is positioned at the tabulation stop.

3.162 start edge

The edge of the positioning area of a basic layout object that is in the direction opposite to the character path.

3.163 structural element

The structural elements of a document are the content portion, the object and the object class.

3.164 tabulation alignment

A layout or imaging process that causes the sequence of character images for à specified character string to be positioned according to a specified method, (start-aligned, end-aligned, centred or aligned-around) at a specified point (tabulation stop) along a reference line.

3.165 tabulation stop

A position along a reference line that is to be used for a specified method (start-aligned, end-aligned, centred or aligned-around) of tabulation alignment.

3.166 text unit

A data structure representing a content portion description.

3.167 top edge

The edge of the positioning area of a basic layout object that is in the direction opposite to the line progression.

3.168 top left corner

The corner of a layout object that is least progressed both in the horizontal and vertical directions of this layout object.

3.169 top right corner

The corner of a layout object that is most progressed in the horizontal direction and least progressed in the vertical direction of this layout object.

3.170 trailing edge

The edge of a frame or block that is orthogonal to the direction of the layout path and that is met first, from the outside of the frame or the block, in the direction of the layout path.

3.171 unit scaling

A scaling factor (an integer or a fraction) that is applied to the basic measurement unit (BMU) to derive a scaled measurement unit (SMU).

3.172 variable spacing

The characteristic of a font wherein the distance between the position point and the escapement point for different character images may be different.

3.173 widow

One or more lines of text associated with preceding text but isolated from it by a page or column boundary.

4   Conventions

The following conventions are used in the T.410 series of Recommendations:

4.1   Names of attributes

The names of attributes are referenced by giving the name of the attribute in double quotation marks preceded by the word attribute. For example:

"...expressed by the attribute "subordinates". This attribute..."

In some cases terms are used to describe concepts which have the same name as an attribute. In the case of reference to such concepts neither the word attribute nor quotes are used.

4.2   *Names of attribute values*

The names of attribute values are referenced by giving the name of the attribute value in single quotation marks. For example:

"...the second parameter allows a sender to select a recto or a verso presentation of the page by specifying 'recto' or 'verso', respectively."

4.3   *Component and component description*

The term component may be used in conjunction with qualifying terms, including: basic, composite, page set, page, frame, block, logical and layout. For example:

a)   "layout components containing block" means "layout objects containing blocks, or classes of layout objects containing blocks";

b)   "frame component" means "an object of type frame, or a class from which objects of type frame may be derived".

The term component description may be qualified the same way.

4.4   *Introduction of terms*

Underlining is used for the purpose of highlighting the point at which the definition of a term occurs in the text.

These terms are also defined in Section 3.

5   General concept of ODA

5.1   *Purpose of ODA*

The purpose of the document architecture is to facilitate the interchange of documents, in a manner such that:

-    different types of content, including text, image, graphic and sound, can coexist within a document;

-    the intentions of a document originator with respect to editing, formatting and presentation can be communicated most effectively.

This Section uses a number of terms for which definitions are given in Section 3. However, for the purpose of the current Section, different though compatible definitions of the essential terms are given below.

-    Interchange is the process of providing a document to a receiving person or device, by means of data communication or by exchange of storage media.

-    Editing is the carrying out of operations associated with creation and amendment of the structure and/or the content of a document.

-    Formatting is the carrying out of operations to determine the layout of a document, i.e. the appearance of its content on a presentation medium.

-    Presentation is the operation of rendering the content of a document in a form perceptible to a human being. Typical presentation media are paper and video screens.

The document architecture provides for the representation of documents in three forms:

- Formatted form, that allows documents to be presented as intended by the originator;

- Processable form, that allows documents to be edited and formatted;

- Formatted processable form, that allows documents to be presented as well as edited and reformatted;

Alternative terms commonly used are "final form" and "image form" for "formated form", and "revisable form" for "processable form". Each of these forms allows the originator to express intentions regarding the structuring and/or formatting of the interchanged document.

5.2 *Overall concept of ODA*

The concept of ODA is based on:

- the existence of a layout view and a logical view of the document: the view from the physical viewpoint (for example, a collection of pages) and the view in the sense of its abstract components (for example, an assembly of sentences);

- the existence of a specific structure and a generic structure; the specific "document" structure is the one that the user may read; the generic structure is the template that guides the creation of the document and that could be re-used for its amendment;

- the existence of document classes: a document class is the set of generic features that are common to a category of documents (for example, Sales Report Form).

5.2.1 *Logical structure and layout structure*

The key concept in the document architecture is that of structure. Document structure is the division and repeated subdivision of the content of a document into increasingly smaller parts. The parts are called objects. The structure has the form of a tree.

The document architecture permits two structures to be applied to a document: a logical structure and a layout structure. Any one or both structures y be applied to a given document.

In the logical structure, the document is divided and subdivided on the basis of the meaning. Examples of logical objects are chapters, sections, figures and paragraphs.

In the layout structure, the document is divided and subdivided on the basis of the layout. Examples of layout objects are pages and blocks.

An example of the logical view of a document called "report" is shown in Figure 1/T.411.

Report

Report heading | Table of contents | Chapter

Author's name | Title | Date | T. O. C. entry | Chapter heading | Section

Chapter number | Chapter title | Page reference | Section heading | Section element

Paragraph | Figure

Picture | Caption

A — One or more occurrences of A

A / B C — A consists of B and C

A / B C — A consists of B or C

T0801130-87
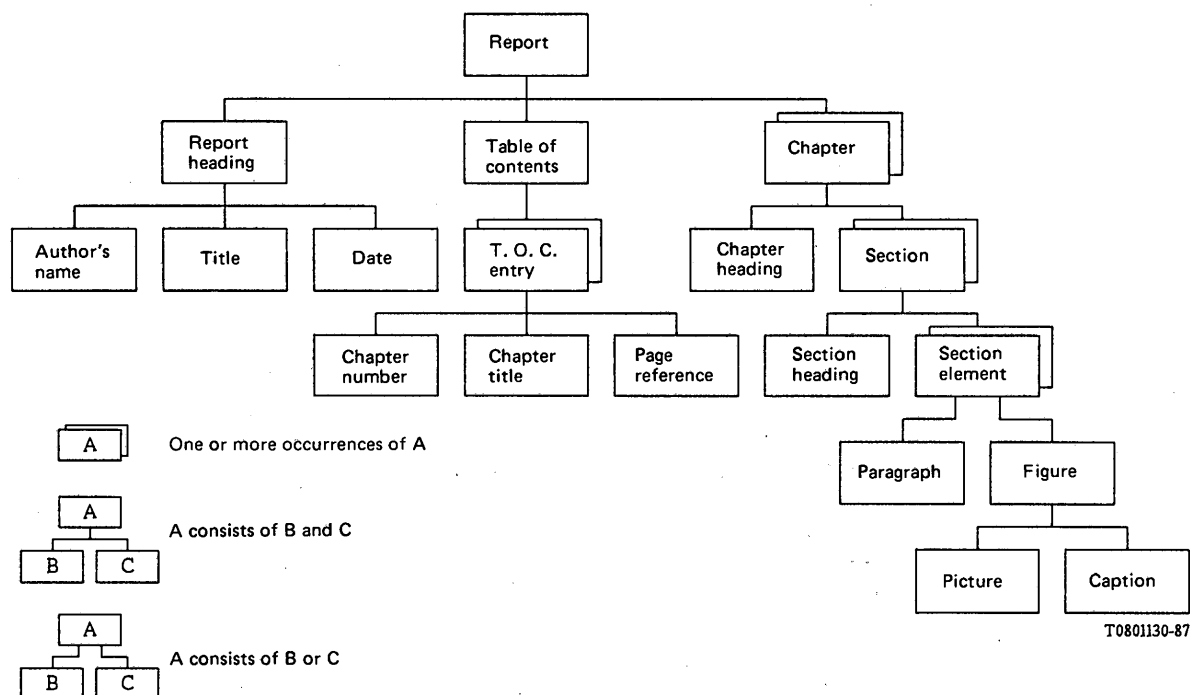
FIGURE 1/T.411

Example of a logical view of a document

The logical structure and the layout structure provide alternative but complementary views of the same document. For example, a book can be regarded as consisting of chapters containing figures and paragraphs, or alternatively, as consisting of pages that contain text blocks and/or graphic blocks.

An object that is not subdivided into smaller objects is called a basic object. All other objects are called composite objects.

The following types of layout objects are defined in the document architecture:

-   block: a basic layout object corresponding to a rectangular area on the presentation medium containing a portion of the document content;

-   frame: a composite layout object corresponding to a rectangular area on the presentation medium and containing either one or more frames or one or more blocks;

-   page: a basic or composite layout object corresponding to a rectangular area of the presentation medium and, if it is a composite object, containing either one or more frames or one or more blocks;

-   page set: a set of one or more page sets and/or pages;

-   document layout root: the highest level object in the hierarchy of the specific layout structure.

For logical objects, no classification other than "basic logical object", "composite logical object" and "document logical root" is defined in the document architecture. Logical object categories such as "chapter", "section" and "paragraph" are application-dependent and can be defined using the "object class" mechanism (see § 5.2.5).

5.2.2 *Content portions*

The basic elements of the content of a document are called content elements. For content consisting of character text, the content elements are characters. In the case of images or graphics, the content elements are Picture elements (also called pels) or geometric graphics elements (lines, arcs, polygons, etc.).

When a document has both logical structure and layout structure, each content element belongs, in general, to exactly one basic logical object and to exactly one basic layout object. A set of related content elements that belong to one basic logical object (if the document has any logical structure) and one basic layout object (if the document has any layout structure) is called a content portion.

It follows from this description that:

- a basic logical object has associated with it one or more content portions;

- a basic layout object has associated with it one or more content portions;

- any logical or layout object (basic or composite) has associated with it an integral number of content portions;

- there is, in general, no one-to-one correspondence between logical objects and layout objects.

The last point is illustrated by Figure 2/T.411.
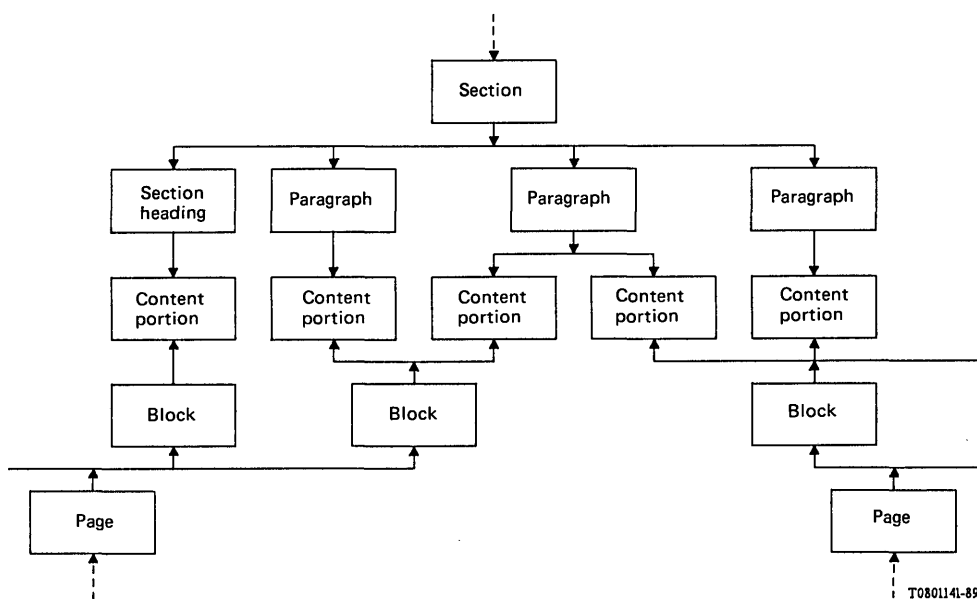
FIGURE 2/T.411

Possible correspondence between logical and layout objects

5.2.3 *Content architectures*

A content portion associated with a basic logical object or a basic layout object may have a more detailed internal structure. The rules governing such an internal structure depend on the type of content and are called a content architecture. The content of a basic logical object or a basic layout object is structured according to only one content architecture.

Recommendations T.416, T.417 and T.418 contain definitions of content architectures for characters, raster graphics and geometric graphics.

5.2.4 *Attributes*

An attribute is a property of a document, or of a document constituent (i.e. a logical object, a layout object, a logical object class, a layout object class, a style or a content portion). It expresses a characteristic of the document or document component concerned, or a relationship with one or more documents or document components.

The set of attributes associated with a document as a whole is called document profile. It represents reference information about the document and may repeat information in the document content, for example the title and the name of the author.

The set of attributes that applies to a logical object or a layout object depends on the type of the object: different sets of attributes are defined for basic logical objects, composite logical objects, document logical root, blocks, frames, pages, page sets and document layout root. These are called document architecture attributes. Document architecture attributes are independent of the type of content of the objects to which they apply.

Examples of document architecture attributes are:

-   the attribute "object identifier" (all objects);

-   the attribute "subordinates" (composite objects);

-   layout directives such as the attribute "indivisibility", the attribute "offset", the attribute "separation" (different sets of attributes for basic and composite logical objects);

-   the attribute "position" (blocks and frames);

-   the attribute "dimensions" (blocks, frames and pages).

In addition to the document architecture attributes, a set of presentation attributes applies to basic logical and basic layout objects. The set of presentation attributes that applies to a given basic object depends on the content architecture governing the content of this object: a different set of presentation attributes is defined for each content architecture.

Examples of presentation attributes are:

-   the attribute "line spacing" (character content architectures);

-   the attribute "clipping" (raster graphics content architectures);

-   the attribute "line rendition" (geometric graphics content architectures).

Presentation attributes may be collected into presentation styles, to which references may be made from both logical and layout objects.

The attributes that apply to a content portion include a content portion identifier and a set of coding attributes, the composition of which depends on the coding method used for the content, for example, the attribute "number of pels per line" for facsimile-coded raster graphics images.

5.2.5 *Relations between logical structure and layout structure*

The logical structure and the layout structure are, in principle, independent of each other. The logical structure of a document is determined by the author and embedded in the document during the editing process. The layout structure is usually determined by a formatting process. The formatting process may be controlled by attributes called layout directives associated with the logical structure.

Examples of layout directives are:

-   the requirement that a chapter starts on a new page;

-   the requirement that the title of a section and the first two lines of its first paragraph are presented on the same page;

-   the amount of indentation for a list of items.

Layout directives may be collected into layout styles each of which may be referred to by one or more logical objects.

5.2.6 *Specified and generic structures*

In a document, the logical objects and/or the layout objects can often be classified into groups of similar objects. Therefore the concept of object class is introduced.

The similarity may be related to logical features such as chapter, section or paragraph hierarchy, to layout features such as size or style, or to content such as page headers and footers. Even an entire document may be a member of a group of similar documents, a letter, a memorandum or a report.

An object class or a document class is a specification of the set of properties that are common to its members. Such a specification consists of a set of rules to determine the values of the attributes that specify the common properties. These rules can be used to control the consistency among the objects or documents making up the class, and to facilitate the creation of additional objects or documents within the class.

The set of logical object classes and layout object classes associated with a document, and their relationships, are called generic logical structure and generic layout structure.

The structures that are particular to a given document are named specific logical structure and specific layout structure.

A document class is described by a generic logical structure and a generic layout structure. The generic logical structure represents the set of all potential specific logical structures, and the generic layout structure represents the set of all potential specific layout structures that are applicable to the document class concerned.

The generic logical structure can be used as a set of rules from which specific logical objects and structures are derived during the editing process. The generic layout structure can be used as a set of rules from which specific layout objects and structures are derived during the formatting process.

An example of generic layout structure is depicted by Figure 3/T.411 which shows a page layout with frames for a header, a footer and two columns of body text.
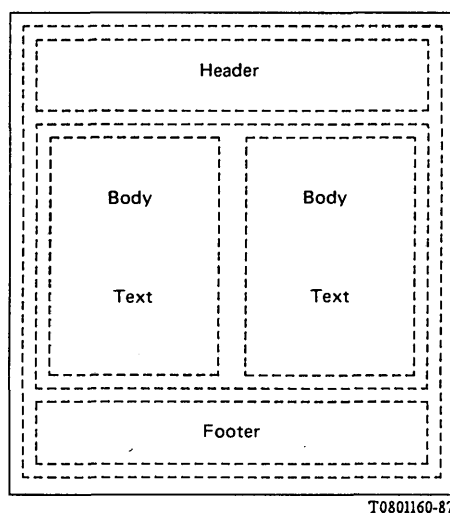
T0801160-87

FIGURE 3/T.411

Example of a page layout

5.2.7 *Document profile*

The document profile consists of a set of attributes associated with a document as a whole. In addition to reference information such as title, date and author's name, which facilitates storage and retrieval of the document, the document profile contains a summary of the document architecture features that are used in the document, in order that a recipient can easily determine which capabilities are required for processing or imaging the document. The attributes representing the latter type of information are called document characteristics and include:

- a specification of the form (formatted, processable or formatted processable) of the document;

- specifications of the content architectures used in the document;

- specifications of the character sets, character fonts, character styles, character orientations and types of emphasis used in the document.

The document profile may be interchanged alone.

5.2.8 *Generic-document*

A generic-document consisting of a document profile end generic structures can be used to assist in the processing of interchanged documents. A generic-document may be interchanged.

5.3 *Document processing model*

5.3.1 *Relationships of ODA to document processing*

The T.410 series of Recommendations is concerned with the definition of a document architecture which permits processing of interchanged documents. A model of document processing is provided as a basis for determining the scope of the processes described by the T.410 series.

A basic model of document processing is summarized in this Section (see Figure 4/T.411). This model is not intended to represent an actual implementation, nor to restrict in any way the processing that may be applied to an interchanged document.

Conceptually, a document is viewed as progressing through three phases of processing as shown in Figure 4/T.411. The order of the processes is not intended to imply that they are performed sequentially in an actual implementation.
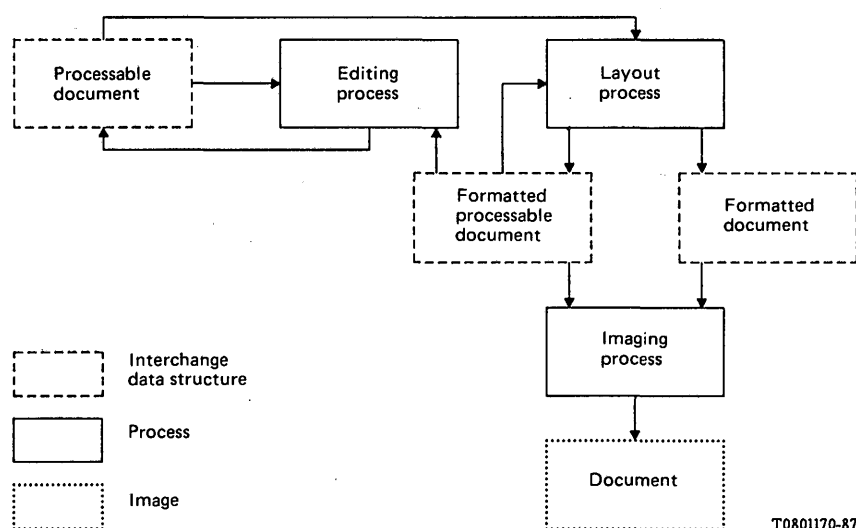
FIGURE 4/T.411

Basic document processing model

5.3.2 *Editing process*

The document editing process is concerned with creating a new document or modifying a previous one. The document architecture provides data structures for representing the document resulting from this process and for representing control information which influences this process.

While document creation and modification may differ in the functions performed and procedures followed, they are considered to be equivalent in the view of this model because the result of both is the same: a new document.

Upon completion of editing, the resulting document can be interchanged. Such a document is said to be interchanged in "processable" form; it is suitable for input to either the editing or layout process.

5.3.3 *Layout process*

The document layout process is concerned with defining a page-oriented organization (i.e. a layout) for the document content. This process can operate in two ways.

The layout process can generate a document which is not intended to be modified; it is suitable only for input to the imaging process. Such a document is said to be in "formatted" form.

This process can also generate a "formatted processable" form document which can be processed further if desired; it is suitable for input to any of the imaging, layout or editing processes.

The document architecture provides data structures for representing both forms of formatted documents and for representing control information which influences the layout process.

5.3.4 *Imaging process*

The document imaging process is concerned with presenting an image of the document in a form perceptible to a human, for example, on a paper or on a screen. A document interchanged in accordance with the T.410 series of Recommendations may contain information relating to the imaging process which allows it to be imaged as required by the originator of the document. However, the imaging process is not defined by the T.410 series and is regarded as a locally defined process that depends on the presentation device used.

Other forms of document processing may be possible; these are not specifically addressed by the document architecture.

6   Overview of Recommendations in the T.410 series

The T.410 series consists of Recommendations T.411, T.412, T.414, T.415, T.416, T.417 and T.418. At present, there is no Recommendation T.413.

6.1   *Recommendation T.411 – Introduction and general principles*

Recommendation T.411 provides information about the T.410 series as a whole by way of an introductory description of the document architecture, an overview of each of the Recommendations and a description of their interdependencies. References necessary for all Recommendations in the Series are given, and terms used throughout all Recommendations in the series are defined. Conformance to the T.410 series is specified and rules for defining document application profiles are given.

6.2   *Recommendation T.412 – Document structures*

Recommendation T.412 defines document architecture concepts which can be applied to the description of representations of documents. The purpose is to permit a common understanding of the structure of a document. The term "document architecture" is used to mean a set of rules by which a document can be produced or interpreted.

Recommendation T.412 describes the architectural concepts and defines the document structures and attributes. It specifies the interface between the document architecture and the content architectures, and defines the document architecture classes. A description of the document processing model is provided.

In addition, examples of document structures based on the T.410 series, and a suggested notation for representing them are included.

6.3   Recommendation T.414 - Document profile

Recommendation T.414 defines the document profile that provides information concerning the handling of the document. This is accomplished by means of attributes (for example, title, author(s)), a few of which apply to the representation of the document profile itself. Some relate to the processing of the document (for example, filing/retrieval, other applications). Other attributes provide a means for a user to specify user-specific information (for example, organization, status). Some of the information given in the document profile could duplicate that in the body of the document.

The document profile may be interchanged alone, that is without the remainder of the document constituents.

Note - Information contained in the profile is intended for a recipient (person) and/or device (for example, keywords). Some attribute values may have been supplied automatically (for example, size).

6.4   Recommendation T.415 - Open document interchange format (ODIF)

Recommendation T.415 defines the format of the data stream used to interchange documents structured in accordance with the T.410 series.

The ODIF data stream is described in terms of a set of data structures, called "interchange data element", which represent the constituents (document profile, object descriptions, object class descriptions, presentation styles, layout styles and content portion descriptions) of a document. The formats of the interchanged data element according to ODIF are defined using the Abstract Syntax Notation One (ASN.1) specified in CCITT Recommendation X.208.

6.5   Recommendation T.416 - Character content architectures

Recommendation T.416 applies to documents that are structured according to the architecture defined in T.412 and that include character content, consisting of a combination of graphic characters, control functions and space characters.

For this type of content architecture it defines those aspects of positioning and imaging that are applicable to the presentation of character content. It also defines specific character content architecture classes in terms of their structure, attributes, character repertoires, control functions and coding.

6.6   Recommendation T.417 - Raster graphics content architectures

Recommendation T.417 applies to documents that are structured according to the architecture defined in T.412 and that include raster graphics content, consisting of a descriptive representation of pictorial information provided by an array of picture elements (pels), encoded according to facsimile or bitmap encoding.

For this type of content architecture, it defines those aspects of positioning and imaging that are applicable to the presentation of raster graphics content. It also defines each class of raster graphics content architecture in terms of its structure, presentation attributes, content layout process, control functions and coding attributes.

6.7   Recommendation T.418 - Geometric graphics content architectures

Recommendation T.418 applies to documents that are structured according to the architecture defined in T.412 and that include geometric graphics content, consisting of a descriptive representation of picture description information as an ordered set of elements such as lines, arcs, polygons, attributes for these drawing elements, elements that structure the content portion, etc., using the Computer Graphic Metafile (CGM) and its binary encoding defined in ISO 8632-1 and ISO 8631-3, respectively.

For this type of content architecture, it defines those aspects of positioning and imaging that are applicable to the presentation of geometric graphics content. It also defines the geometric graphics content architecture class in term of its structure, presentation attributes, the relevant CGM parameters, the content layout process, control functions and coding attributes.

7 Inter-dependencies of the Recommendations

If there is a requirement to interchange documents or generic-documents, it is necessary to use Recommendations T.411, T.412, T.414 and T.415 together:

- Rec. T.411: Introduction and general principles;

- Rec. T.412: Document structures;

- Rec. T.414: Document profile;

- Rec. T.415: Open document interchange format (ODIF).

Should there be a requirement to interchange just the document profile, then only Recommendations T.411, T.414 and T.415 are necessary.

Additionally, it will be necessary to use one or more of the remaining Recommendations, depending on the particular type of content to be interchanged, for example:

- Rec. T.416: Character content architectures.

8 Conformance

The conformance to the T.410 series of Recommendations is defined in terms of conformance of a data stream that represents a document, a generic-document or a document profile. For the definition of conformance it is necessary to distinguish two cases:

- the document profile attribute "document application profile" is the identifier of a document application profile;

- no value is specified for the document profile attribute "document application profile",

A document application profile can only be specified if it is identified by an ASN.1 object identifier. This includes document application profiles defined in International Standards or CCITT Recommendations, or registered by registration authorities (see Annexes B, C and D of Recommendation X.208).

When the attribute "document application profile" is present in the document profile of a given document or generic-document, the data stream representing this document or generic-document is in conformance with the T.410 series of Recommendations if it conforms to the specified document application profile,

In the absence of the specification of a document application profile, the data stream representing the document or the generic-document must be assumed to conform to Recommendations T.411, T.412, T.414, T.415, T.416, T.417 and T.418. This means that the document or generic-document may contain:

a) any of the document architecture classes defined in Recommendation T.412, any attribute and attribute value permitted for that class;

b) any content architecture class which is defined in Recommendations T.416, T.417 and T.418 that defines such classes and any presentation attribute, control function, coding attribute and graphic element permitted for that class;

c) any document profile attribute defined in Recommendation T.414;

d) anyone of the interchange format classes as defined in Recommendation T.415;

e)   no content architecture classes other than those defined in Recommendations T.416, T.417 or T.418;

f)   only graphic character sets specified by other CCITT Recommendations or International Standards;

g)   only geometric graphic elements specified by ISO 8632-1 and parameter values of these elements defined in ISO 8632-1 and ISO 8632-3 for these elements.

9   Document application profile

A document application profile is the specification of a combination of features that are defined in various Recommendations of the T.410 series. It is identified by a unique ASN.1 object identifier obtained in accordance with the rules in Recommendation X.208.

In order to define a valid combination, the features must be selected according to the rules given in § 9.2:

-   Features pertaining to a document architecture class are selected to form a document architecture level.

-   Features of a content architecture class are selected to form a content architecture level.

-   Features of the document profile are selected to constitute a document profile level.

-   An interchange format class is selected.

A document application profile must include:

-   one or more document architecture levels;

-   one or more content architecture levels;

-   a document profile level;

-   an interchange format class.

The document architecture features can be broken down into:

-   three classes:

    -   formatted document architecture (FDA),
    -   processable document architecture (PDA),
    -   formatted processable document architecture (FPDA);

-   for each class its:

    -   constituents;

-   for each constituent its:

    -   attributes;

-   for each attribute its:

    -   classification (mandatory, non-mandatory, defaultable);
    -   permissible values divided into basic, non-basic values;
    -   default value, if the attribute is defaultable.

The content architecture features depend primarily on the type of content. For each type of content, various content architecture classes exist (for example, for character content architectures the classes are: character formatted, character processable, and character formatted processable);

-   for each content architecture class are defined its:

    -   presentation attributes,
    -   coding attributes,
    -   control functions;

- for each presentation attribute, coding attribute and control function parameter its:

    - permissible values divided into basic, non-basic values;
    - default value.

The features of the document profile are its:

- attributes;

- and for each attribute its:

    - classification (mandatory, non-mandatory),
    - permissible values.

The interchange formats that are permitted by Recommendation T.415 are:

- the open document interchange format (ODIF) divided into:

    - class A,
    - class B.

9.1 *General principles for defining a document application profile*

A document application profile can only place constraints on the previously listed features, it cannot extend them.

A document application profile shall not allow the use of attributes for purposes beyond those defined in the T.410 series of Recommendations. That is, a document application profile shall not modify in any way the semantics of the attributes defined in the T.410 series.

9.2 *Rules for defining a document application profile*

The rules for defining a document application profile consist of rules for defining document architecture levels, content architecture levels, a document profile level an for selecting an interchange format class.

9.2.1 *Rules for defining a document architecture level*

Recommendation T.412 specifies the three document architecture classes that may be used in defining document architecture levels. These are formatted document architecture class, processable document architecture class and formatted processable document architecture class.

For each of these classes, Recommendation T.412 defines which document structures may be used in documents that pertain to that class. These structures are classified as mandatory or optional. Each class also specifies which objects and object classes are applicable to these structures and, again, objects are classified as mandatory or optional. The class also defines which attributes are applicable to those objects and object classes and the body of Recommendation T.412 defines all permissible values and a standard default value for each defaultable attribute.

A document architecture level defines restrictions concerning which structures, objects and object classes, attributes and attribute values are allowed to be contained in documents or generic-documents that pertain to that level.

Note - The term "superclass" is sometimes applied to the set of document classes or object classes whose hierarchy of subordinate object classes and associated attributes and attribute values is restricted by a document application profile.

For each document architecture class, only one document architecture level can be specified. For example, a document application profile cannot make use of two different document architecture levels pertaining to the processable document architecture class (PDA).

The rules for defining a document architecture level are given below.

a) The document architecture level must pertain to a particular document architecture class, that is, the level must make use of only those document structures, objects and object classes that pertain to the specified document architecture class.

b) The document architecture level must specify which document structures pertain to that level. Structures pertaining to the corresponding document architecture class that are mandatory must be specified as mandatory in the document architecture level. Structures specified as optional in the document architecture class may be specified as optional or mandatory in the document architecture level.

Note – If a factor set or a partial generator set is used, then the document architecture level should, in general, specify this as optional.

c) When a document application profile allows the interchange of more than one document architecture class (for example, formatted, processable and formatted processable), the document architecture levels must be consistent. For example, the generic logical structure used in the document architecture level of processable form must be identical to that used in the document architecture level of formatted processable form.

d) The document architecture level must specify which objects and object classes pertain to that level. Objects and object classes that are mandatory for a particular structure must be specified as mandatory in the document architecture level. Objects and object classes that are specified as optional may be specified as optional or mandatory in the document architecture level.

e) The document architecture level must specify any restrictions that are applicable to the document structures that belong to the level. For example, the number of hierarchical levels allowed in a particular structure may be restricted or the specific structures allowed may be required to pertain to certain defined document classes.

f) The document architecture level must specify, in the case of formatted document architecture class, whether the pages are to be composite, basic.

g) The document architecture level must specify, in the case of formatted or processable document architecture classes, whether only one content portion or multiple content portions can be associated with basic objects.

h) For each object or object class used, the document architecture level must specify which attributes are applicable. These must include the appropriate minimum set of attributes pertaining to each object type as defined in Recommendation T.412.

i) For each permitted attribute, the document architecture level must specify the basic, default and non-basic (if any) values that are applicable. These values must be taken from the range of permissible values specified in the attribute definitions in Recommendation T.412.

Note – It is recommended that the default value used for defaultable attributes is that specified in the corresponding document architecture class.

j) The document architecture level may classify attributes that are designated as being defaultable or non-mandatory in Recommendation T.412 as being mandatory for that level. The classification of mandatory attributes must not be changed.

k) The document architecture level must specify which attributes may be included in the attribute "default value lists" and must specify the object types for which a default value list can be specified. Recommendation T.412 gives a definition of the use of this attribute.

9.2.2 *Rules for defining a content architecture level*

Each Recommendation in the T.410 series that caters for particular content types defines one or more than one content architecture class that corresponds to that content type. The number of content architecture classes defined depend upon the particular content type.

Each content architecture class definition consists of the specification of the following:

- a set of presentation attributes,

- a set of content elements,

- a set of control functions,

- the type(s) of coding used,

- a set of coding attributes.

For each presentation attribute and coding attribute, the content architecture class definition specifies the permissible values and a recommended default value. Similarily the content architecture class definition specifies the permissible values and a recommended default value for the control function parameters (where applicable).

Each content architecture class definition also specifies the basic component types that the content architecture class can be used in.

A content architecture level defines restrictions concerning which presentation attributes, control functions and coding attributes, and their values, are allowed to be used in association with content pertaining to that level. The content architecture level may also define restrictions concerning the content elements and types of coding that may be used.

For each content architecture class, that is defined for a particular type of content, only one content architecture level can be specified. For example, a document application profile cannot make use of two different content architecture levels pertaining to the formatted character content architecture class (CF).

The rules for defining a content architecture level are given below.

a) The content architecture level must pertain to a particular content architecture class, that is, the presentation attributes, content elements, control functions, types of coding and coding attributes specified by the content architecture must be taken from those specified in the corresponding content architecture class.

b) When a document application profile allows the interchange of more than one content architecture class pertaining to the same content type (for example, formatted, processable and formatted processable for character content architecture classes), the levels must be consistent. For example the features used in the content architecture level of formatted form must be, when applicable, identical to those used in the content architecture level of formatted processable form.

c) Subject to the above restrictions, there is no further restriction on which presentation attributes, content elements, control functions, type of coding and coding attributes can be specified in a content architecture level.

d) The content architecture level must specify, for each permitted presentation attribute, control function and coding attribute, the basic, default and non-basic (if any) values that are applicable. These values must be taken from the range of permissible values specified in the the corresponding content architecture class.

   Note - It is recommended that the default value used is that specified in the content architecture class.

e) The content architecture level must specify which set or sets of content elements are applicable. These must be taken from the permissible sets specified in the corresponding content architecture class. If appropriate, a default set of content elements may be specified. In addition, a distinction may be made between basic and non-basic content elements. The type or types of coding allowable must also be specified.

   Note - There may be mandatory content elements (for example, BEGIN METAFILE or END METAFILE in the case of geometric graphics content type) that are to be present in every set of content elements specified by a content architecture level.

The T.410 series of Recommendations allows the interchange of documents containing content architecture levels pertaining to content architecture classes that are not defined in this series. It does not define how such content architecture levels should be specified except that the interface between the content architecture and the document architecture should be defined as specified in Recommendation T.412. The only restriction imposed on the use of content architecture levels defined outside of the T.410 series is that they are not allowed to be used if no document application profile identifier is indicated in the document profile (see § 8).

9.2.3 *Rules for defining a document profile level*

Recommendation T.414 defines all attributes that may be specified for use in a document profile. The rules for specifying how document profile attributes may be used in a document profile level are given below.

a) The document profile level may specify any document profile attribute defined in Recommendation T.414. It must not specify attributes not defined in Recommendation T.414.

b) The document profile level must specify the minimum set of document profile attributes defined in Recommendation T.414, Annex B.

c) The document profile level may specify any document profile attribute as being mandatory or non-mandatory for that level.

d) The document profile level must specify attribute values taken from the range of permissible values defined in Recommendation T.414.

e) The document profile level may specify additional restriction on the use of certain attributes and limit the values applicable to these attributes.

f) The document profile level shall not modify the semantics of the absence of attributes from those semantics specified in Recommendation T.414.

9.2.4 *Rules for selecting the interchange format class*

Recommendation T.415 defines the valid interchange format classes that can be used for interchanging a document or a generic-document. It also defines the restrictions on the use of these interchange format classes. Only one interchange format class may be specified in a document application profile. No other restriction may be specified concerning the use of an interchange format class in a document application profile.

ANNEX A

(to Recommendation T.411)

(Informative)

Reference to other Recommendations, standards and registers

Reference to the following Recommendations, standards and registers is made for information. It is not required for the application of the T.410 series of Recommendations.

- Rec. T.61 (1984): Character repertoire and coded character sets for the international Teletex service

- Rec. T.73 (1984): Document interchange protocol for the Telematic services

- Rec. T.400 (1988): Introduction to document architecture, transfer and manipulation

- Rec. T.431 (1988): Document transfer and manipulation (DTAM) - Services and protocols - Introduction and general principles

- Rec. T.432 (1988): Document transfer and manipulation (DTAM) - Services and protocols - Service definition

  Rec. T.433 (1988): Document transfer and manipulation (DTAM) - Services and protocols - Protocol specification

- Rec. T.441 (1988) : Document transfer and manipulation (DTAM) - Operational structure

- Rec. T.501 (1988): Document application profile MM for the interchange of formatted mixed mode documents

- Rec. T.502 (1988) Document application profile PM1 for the interchange of processable form documents

- Rec. T.503 (1988): Document application profile for the interchange of Group 4 facsimile documents

- Rec. T.504 (1988): Document application profile for videotex interworking

- Rec. T.521 (1988): Communication application profile BT0 for document bulk transfer based on the session service (according to the rules defined in T.62 *bis*)

- Rec. T.522 (1988): Communication application profile BT1 for document bulk transfer

- Rec. T.523 (1988): Communication application profile DM1 for videotex interworking

- Rec. T.541 (1988): Operational application profile for videotex interworking

- Rec. T.561 (1988): Terminal characteristics for mixed mode of operation MM

- Rec. T.562 (1988): Terminal characteristics for Teletex processable mode PM1

- Rec. T.563 (1988): Terminal characteristics for Group 4 facsimile apparatus

- Rec. T.564 (1988): Gateway characteristics of videotex interworking

- Rec. X.200 (1984): Reference model of open systems interconnection for CCITT applications

- Rec. X.210 (1984): Open systems interconnection (OSI) layer service definition conventions

- Recommendation X.215 (1984): Session service definition for open systems interconnection in CCITT applications

- Recommendation X.216 (1984): Presentation service definition for open systems interconnection for CCITT applications

- ISO 216 (1975): Writing paper and certain classes of printed matter trimmed sizes - A and B series

- ISO 2375 (1985): Data Processing - Procedure for registration of escape sequences

- ISO 7350 (1984): Text Communication - Registration of graphic character subrepertoires

- ISO 8613-1 (1988): Information processing - Text and office systems - Office document architecture (ODA) and interchange format - Part 1: Introduction and general principles

- ISO 8613-2 (1988): Information processing - Text and office systems - Office document architecture (ODA) and interchange format - Part 2: Document structures

- ISO 8613-4 (1988): Information processing - Text and office systems - Office document architecture (ODA) and interchange format - Part 4: Document profile

- ISO 8613-5 (1988): Information processing - Text and office systems - Office document architecture (ODA) and interchange format - Part 5: Office document interchange format

- ISO 8613-6 (1988): Information processing - Text and office systems - Office document architecture (ODA) and interchange format - Part 6: Character content architectures

- ISO 8613-7 (1988): Information processing - Text and office systems - Office document architecture (ODA) and interchange format - Part 7: Raster graphics content architectures

- ISO 8613-8 (1988): Information processing - Text and office systems - Office document architecture (ODA) and interchange format - Part 8: Geometric graphics content architectures

- ISO International register of character sets to be used with escape sequences

- ISO International register of graphic character subrepertoires

- ECMA - 101 (1985): Office document architecture

- ANSI X3.151 (1987): Bond papers and index bristols - Basic sheet sizes and standard stock sizes

- JIS P 0138 (1961): Trimmed sizes of paper

ANNEX B

(to Recommendation T.411)

(Informative)

Relationships with other Recommendations and standards

B.1   The T.410 series of Recommendations has been developed in parallel with ISO 8613 (1988): Information processing - Text and office systems - Office document architecture (ODA) and interchange format.

Except for references (Recommendations make reference to other Recommendations rather than to their equivalent ISO Standards), mandated stylistic differences and provisions of ISO 8613 that are outside the scope of the T.410 series of Recommendations, the texts in the T.410 series are identical to the texts in correspondingly numbered parts of ISO 8613.

B.2   Provision has been made in the T.410 series of Recommendations for compatibility with Recommendation T.73 (1984): Document transfer protocol for the telematic services, by providing for a specific document interchange format class B, and by the provision of a number of structures and attributes primarily intended for use in document interchange format class B. These structures and attributes are identified and cross-referenced to the appropriate Recommendations of the T.410 series in Annex C.

Whenever interchange format class B is used, with the appropriate document structures and attributes, documents may be exchanged with application contexts conforming to Recommendation T.73 (1984).

Both the Recommendations T.73 (1984) and the T.410 series of Recommendations application contexts will need to define the interchange by use of identical document application profiles.

B.3　The T.410 series of Recommendations has been developed in parallel with ECMA-101, Office Document Architecture (1985). The text in the T.410 series is identical to the second edition of ECMA-101 (to be published) except for mandatory stylistic changes and provisions of ECMA-101 that are outside the scope of these CCITT Recommendations.

ANNEX C

(to Recommendation T.411)

(Informative)

**Correspondence between Recommendation T.73 (1984)
and Recommendations in the T.410 series**

CCITT Recommendation T.73 (1984) is replaced by Recommendations in the T.410 series.

This Annex describes the relationships between Recommendation T.73 (1984) and Recommendations in the T.410 series.

C.1　*Data stream format*

In Recommendation T.73 (1984) only one interchange data stream is specified, which corresponds to the interchange format class B in Recommendation T.415.

C.2　*Presentation capabilities descriptor*

The "presentation capabilities descriptor" specified in Recommendation T.73 (1984) corresponds to the "document characteristics" in Recommendation T.433.

C.3　*Attributes*

Several attributes and attribute values have different names in Recommendation T.73 (1984) and Recommendations in the T.410 series.

Table C-1/T.411 lists all attributes of Recommendation T.73 (1984) along with their locations, together with the corresponding names and locations in Recommendations in the T.410 series.

TABLE C-1/T.411

Correspondence of attributes and values

T.73 (1984)		T.410 Series	
Attribute/Value	Location	Attribute/Value	Location
"object type" 'document' 'page set' 'page' 'frame' 'block'	2.5.3.1	"object type" 'document layout root' 'page set' 'page' 'frame' 'block'	T.412 5.3.1.1
"object identifier"	2.5.3.2	"object identifier" or "object class identifier"	T.412 5.3.1.2 T.412 5.3.1.3
"reference to corresponding generic object"	2.5.3.3	"object class"	T.412 5.3.3.1
"reference to subordinate objects"	2.5.3.4	"subordinates"	T.412 5.3.3.2
"reference to content portions"	2.5.3.5	"content portions"	T.412 5.3.3.3
"user-readable comments"	2.5.3.6	"user-readable comments"	T.412 5.3.5.1
"default value list"	2.5.3.7	"default value list"	T.412 5.3.5.5
"position"	2.5.3.8	"position"	T.412 5.4.1.1
"dimensions"	2.5.3.9	"dimensions"	T.412 5.4.1.2
"claiming"	2.5.3.10	DELETED	
"transparent" 'transparent'	2.5.3.10	"transparency" 'transparent'	T.412 5.4.3.2
"content type" 'character box element' 'photographic element'	2.5.4.1	"content type" 'formatted character content architecture' 'formatted raster graphics content architectures'	T.412 5.3.4.2

Conventions: - names of attributes in double quotation marks
            - names of attribute values in single quotation marks

TABLE C-1/T.411 *(cont.)*

Correspondence of attributes and values

T.73 (1984)		T.410 Series	
Attribute/Value	Location	Attribute/Value	Location
Presentation attributes for character box elements		Presentation attributes for character content architectures	
"character path" '0', '90', '180', '270'	2.5.4.2.1	"character path" '0', '90', '180', '270'	T.416 7.1.4
"line progression" '90', '270'	2.2.4.2.1	"line progression" '90', '270'	T.416 7.1.14
"character box orientation" '0', '90', '180', '270'	2.5.4.2.1	"character orientation" '0', '90', '180', '270'	T.416 7.1.3
"character box size"	2.5.4.2.2	DELETED	
"character base line offset"	2.5.4.2.2	DELETED	
"character spacing"	2.5.4.2.2	"character spacing"	T.416 7.1.5
"line spacing"	2.5.4.2.2	"line spacing"	T.416 7.1.15
"alignment" 'left aligned' 'right aligned' 'centered' 'justified'	2.5.4.2.3	"alignment" 'start-aligned' 'end-aligned' 'centred' 'justified'	T.416 7.1.1
"line layout"	2.5.4.2.3	"line layout table"	T.416 7.1.13
"initial offset"	2.5.4.2.3	"initial offset"	T.416 7.2.2
"graphic rendition"	2.5.4.2.4	"graphic rendition"	T.416 7.1.10
Conventions: - names of attributes in double quotation marks - names of attribute values in single quotation marks			

TABLE C-1/T.411 *(cont.)*

Correspondence of attributes and values

T.73 (1984)		T.410 Series	
Attribute/Value	Location	Attribute/Value	Location
Presentation attributes for photographic elements		Presentation attributes for raster graphics content architectures	
"pel path" '0', '90', '180', '270'	2.5.4.3.1	"pel path" '0', '90', '180', '270'	T.417 6.1.3
"line progression" '90', '270'	2.5.4.3.1	"line progression" '90', '270'	T.417 6.1.2
"pel transmission density" '180', '200', '240 , '300', '400', '600', '1200' (pels per 25.4 mm)	2.5.4.3.2	"pel transmission density" n/a '6', '5' '4', '3', '2', '1', '0' (BMU per pel spacing)	T.417 6.2.2
"initial offset"	2.5.4.3.3	"initial offset"	T.417 6.2.1
Attributes of content portions		Content portion attributes	
"content portion identifier"	2.5.5.1	"content identifier layout"	T.412 5.9.1
"type of coding" 'T.61', 'T.6'	2.5.5.2	"type of coding" 'ISO 2022', 'T.6'	T.412 5.9.2
Conventions: - names of attributes in double quotation marks - names of attribute values in single quotation marks			

TABLE C-1/T.411 *(cont.)*

Correspondence of attributes and values

T.73 (1984)		T.410 Series	
Attribute/Value	Location	Attribute/Value	Location
Coding attributes for photographic elements		Coding attributes for raster graphics content architectures	
"number of pels per line"	2.5.5.3	"number of pels per line"	T.417 7.2.3
"number of discarded pels"	2.5.5.3	"number of discarded pels"	T.417 7.2.4
"number of lines"	2.5.5.3	"number of lines"	T.417 7.2.2
"compression"	2.5.5.3	"compression"	T.417 7.2.1
"alternative graphic representation"	2.5.5.4	"alternative representation"	T.412 5.9.3.2
Document profile attributes		Document profile attributes	
reference to generic layout structure value=obj.id. "0"	2.3.3	"generic layout structure" 'partial'	T.414 5.2.1
reference to specific layout structure value=obj.id. "1"	2.3.3	"specific layout structure" 'present'	T.414 5.2.2
presentation capabilities	2.3.3	document characteristics	T.414 5.3
other document profile attributes	2.3.3	document management attributes	T.414 5.4
basic terminal characteristics 'Teletex' 'Group 4 Facsimile' 'Mixed Mode'	4.4	"document application profile" (no value for Teletex) 'Group 4 Facsimile' 'Mixed Mode'	T.414 5.3.1

Conventions: - names of attributes in double quotation marks
          - names of attribute values in single quotation marks

TABLE C-1/T.411 *(end)*

Correspondence of attributes and values

T.73 (1984)		T.410 Series	
Attribute/Value	Location	Attribute/Value	Location
Interchange format	4.4	"document architecture class" (no value for TIF.0)	T.414 5.3.3
'TIF.0'		'Formatted'	
'TIF.1'			
Non-basic terminal capabilities	4.4	Non-basic document characteristics	T.414 5.3.7
"graphic character sets"		DELETED	
"control·character sets"		DELETED	
"page dimensions"		"page dimensions"	T.414 5.3.7.4.1
"coding attributes"		"coding attributes" 'raster graphics coding attributes'	T.414 5.3.7.5
"presentation attributes"		"character presentation features" "raster graphics presentation features"	T.414 5.3.7.6
Non-basic structural capabilities	4.4	Non-basic structure characteristics	T.414 5.3.8
"number of objects per page"		"number of objects per page"	T.414 5.3.8.1

Conventions: - names of attributes in double quotation marks
            - names of attribute values in single quotation marks

ANNEX D

(to Recommendation T.411)

(Informative)

Principles for the assignment of ASN.1 object identifier values

Values of ASN.1 object identifiers are assigned in various Recommendations in the T.410 series. The assignment of these values is based on the following principles:

a) the value of the first component is 2, representing "joint-iso-ccitt";

b) the value of the second component is 8, designating the area of joint ISO-CCITT work "document architecture";

c) the value of the third component is 0, 1, 2 or 3, identifying one of the following categories of object identifier values assigned within this area of work:

0 - object identifier value to be used as a part of an ASN.1 external data type;

1 - object identifier value to be used as a part of an ASN.1 module identifier;

2 - object identifier value for the identification of a content architecture class;

3 - object identifier value for the identification of a type of coding;

d) the meaning of the fourth component and that of the fifth component, if any, depends on the value of the third component as follows:

- if the value of the third component is 0, the fourth component identifies a particular external data type; values of the fourth component are assigned in Recommendation T.415; in this case, there is no fifth component;

- if the third component is 1, 2 or 3, the fourth component identifies the Recommendation in the T.410 series in which the value of the fifth component is assigned:

5 - T.415,

6 - T.416,

7 - T.417,

8 - T.418.

ANNEX E

(to Recommendation T.411)

(Normative)

Use of MHS to interchange documents conforming to the T.410 series of Recommendations

E.1 *ODA identification in the P1 protocol of MHS*

Documents shall be identified by a set of ASN.1 object identifiers as externally-defined encoded-information-types. One member shall always be the ASN.1 object identifier for ODA, the other members shall be one or more ASN.1 object identifiers for the document application profiles to which the message body parts conform.

ODA document	{2 8 0 0}
Document Application Profile	{2 8 . . }
... ...	{2 8 . . }
... ...	{. . . .}
... ...	{. . . .}

Note 1 - Documents conforming to ODA shall not be converted.

E.2   *ODA identification in the P2 protocol of MHS*

Documents conforming to ODA shall be identified as ODA extended body parts. Each extended body part shall contain parameter information about the applicable document application profile and the document architecture class.

Note - ODA body parts can be mixed with non-ODA body parts in a P2 body.

The module for specifying the ODA body parts is described below:

IPMSExtendedBodyPartTypeOda   {joint-iso-ccitt(2) oda(8) modules(1) part(0) extended-body-part-type-oda(0)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- Prologue

EXPORTS
        oda-body-part
        OdaBodyPartParameters
        OdaData

IMPORTS

        Interchange-Data-Element
        FROM Interchange-Data-Elements {2 8 1 5 5}
        EXTENDED-BODY-PART-TYPE
        FROM IPMSInformationsObjects {joint-iso-ccitt(2)
                mhs-motis(6) ipms(1) modules(0)
                information-objects(2)};

oda-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS   OdaBodyPartParameters
        DATA   OdaData
        ::= id-et-oda

AbstractSyntax ·   ::= CHOICE {
        OdaBodyPartParameters
                -- shall appear in the parameter element of an
                -- IPM ExtenallyDefinedBodyPart --,

        OdaData
                -- shall appear in the data element of an
                -- IPM ExternallyDefinedBodyPart --}

OdaBodyPartParameters   ::= SET {
        document-application-profile   [0]   OBJECT IDENTIFIER
                -- This object identifier value shall also be used in the
                -- MTS ExternalEncodedInformationType
                -- in addition to the id-et-oda object identifier
        document-architecture-class   [1]   INTEGER {
                                                formatted (0)
                                                processable (1)
                                                formatted processable (2)}}

OdaData .   ::= SEQUENCE OF Interchange-Data-Element

id-et-oda OBJECT identifier   ::= { 2 8 1 0 1 }
        -- Identifies the AbstractSyntax using the ASN.1 basic encoding rules
        -- This object identifier value shall also be used in the
        -- MTS ExternalEncodedInformationType

End -- of IPMSExtendedBodyPartTypeOda

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

CONTENTS

---

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

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.
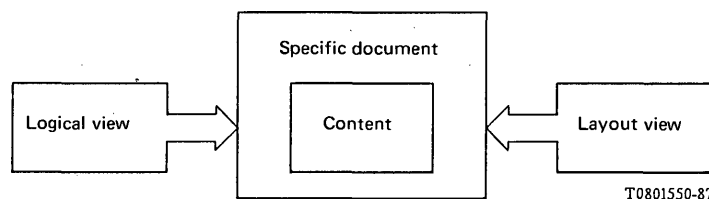
FIGURE 1/T.412

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 their coded representation, and of the rules for the application of the attributes and control functions to the content elements.

Selection of content architecture depends on the type or types of content elements to be represented. A single document may contain 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

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

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

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

Documents may reference one externally specified document class description (see § 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.

```
Document
description                      ┌──────────┐
                                │ Document │
                                │ profile  │
                                └──────────┘

                          Document body
                           Generic part
    Generic logical description          Generic layout description
    ┌──────────────┐                     ┌──────────────┐
    │ Logical      │                     │ Layout       │
    │ root class . │                     │ root class   │
    │ description  │                     │ description  │
    │      ┌───────────┐  ┌───────────┐  │     ┌───────────┐  ┌───────────┐
    └──────│ Logical   │  │ Content   │  └─────│ Layout    │  │ Content   │
           │ object    │  │ portion   │        │ object    │  │ portion   │
           │ class     │  │ description│       │ class     │  │ description│
           │ description│ │           │        │ description│ │           │
           └───────────┘  └───────────┘        └───────────┘  └───────────┘

                           Style part
    Layout style set                     Presentation style set
        ┌───────────┐                        ┌───────────┐
        │ Layout    │                        │ Presentation│
        │ style     │                        │ style     │
        └───────────┘                        └───────────┘

                          Specific part
    Specific logical description
    ┌──────────────┐                     ┌──────────────┐
    │ Layout       │                     │ Logical      │
    │ root         │                     │ root         │
    │ description  │                     │ description  │
    │      ┌───────────┐  ┌───────────┐  │     ┌───────────┐  ┌───────────┐
    └──────│ Logical   │  │ Content   │  └─────│ Layout    │  │ Content   │
           │ object    │  │ portion   │        │ object    │  │ portion   │
           │ description│ │ description│       │ description│ │ description│
           └───────────┘  └───────────┘        └───────────┘  └───────────┘
                                               Specific layout description
```

T0802510-89

┌──────┐
│   │   Constituent   ⌜ ⌝   Set of
└──────┘   ⌞ ⌟   constituents

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.
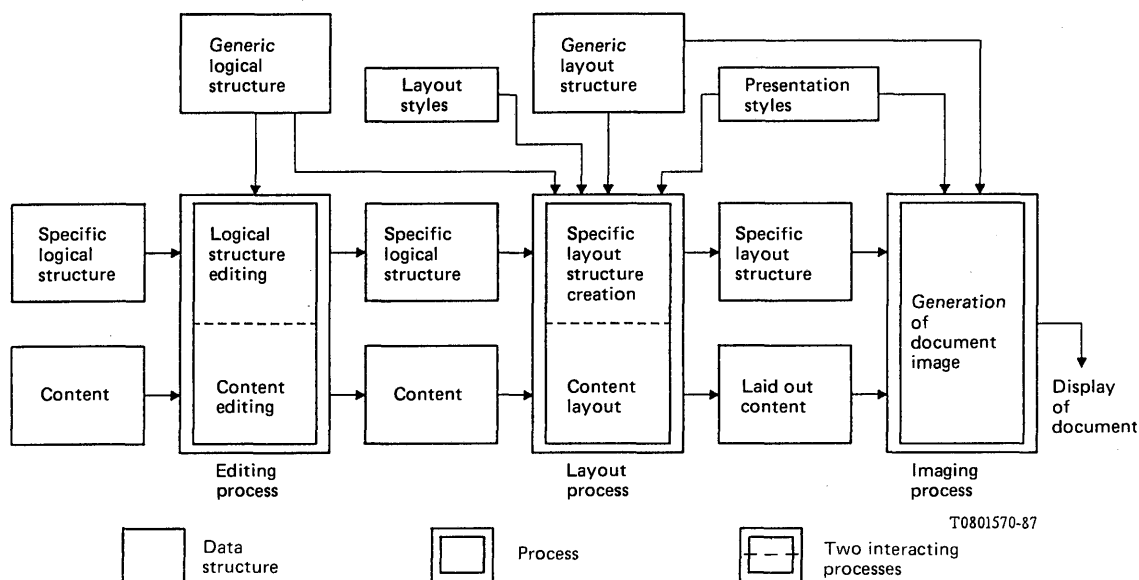
FIGURE 3/T.412

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.

T0801580-87
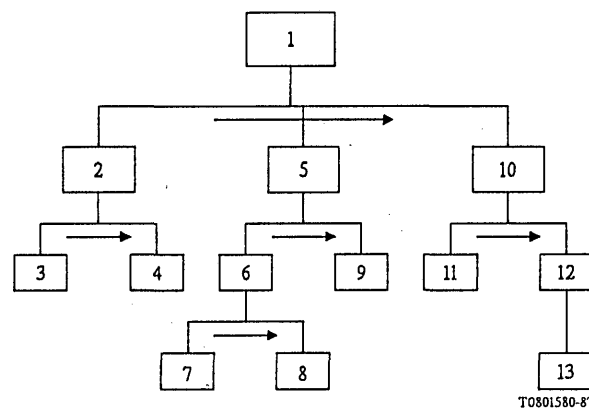
Note – Arrows define sequential order.

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.

T0801590-87
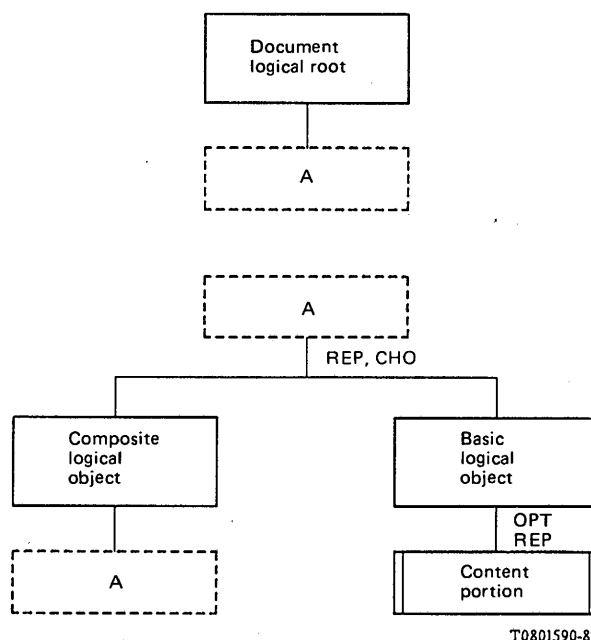
Note - The notation used in this Figure is that defined in Annex A.

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.

T0801600-87
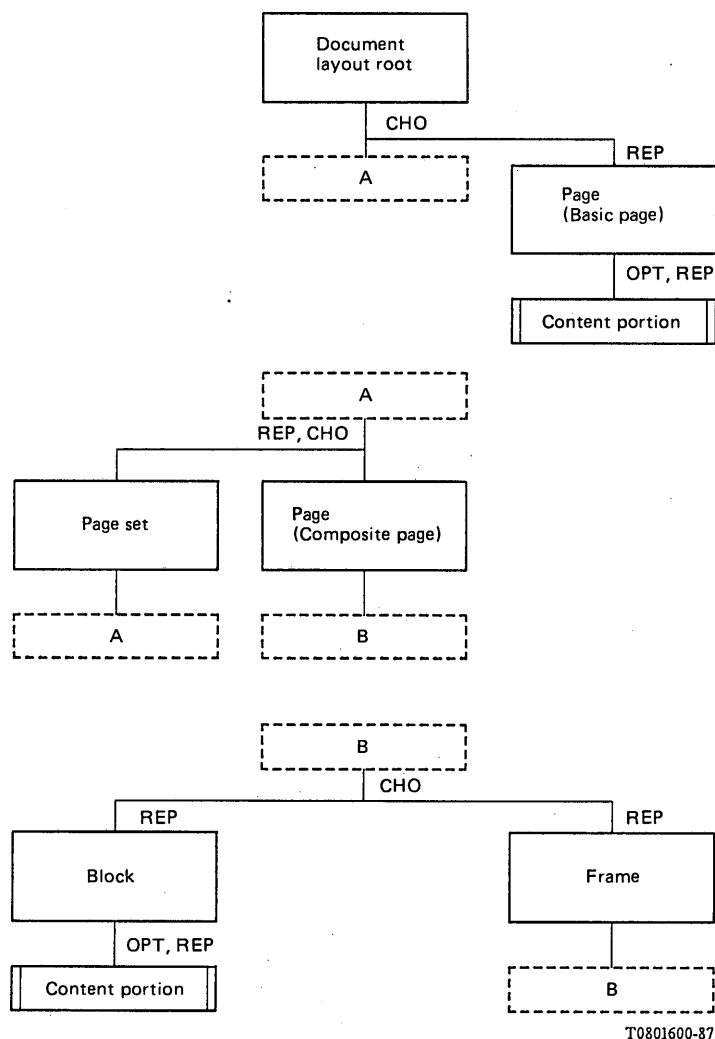
Note - The notation used in this figure is that defined in Annex A.

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.

Horizontal axis of page

Top left corner

Top right corner

Vertical
axis of
page

Horizontal direction
of frame or block

Page

Frame or block

Vertical direction
of frame or block

Bottom left corner

Bottom right corner

T0802520-89
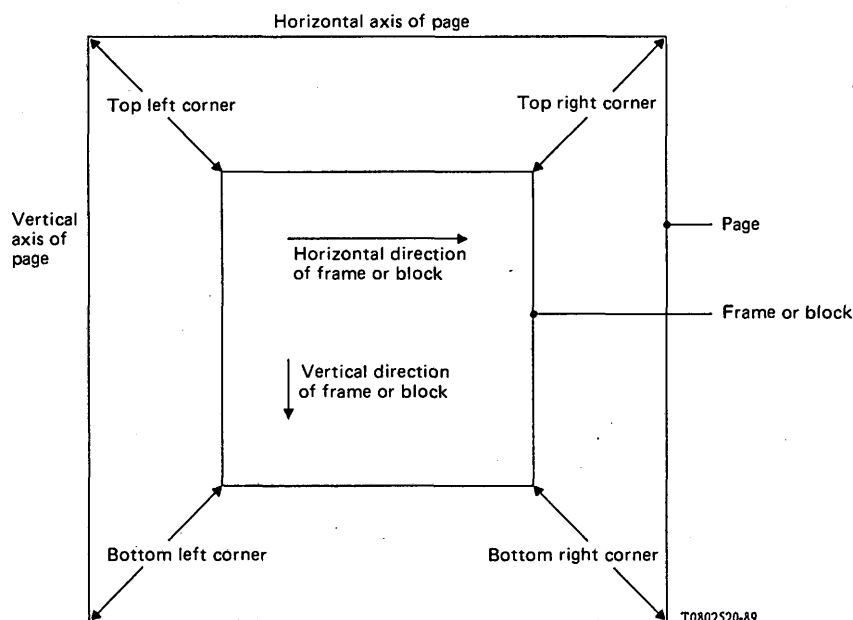
FIGURE 7/T.412

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° 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.
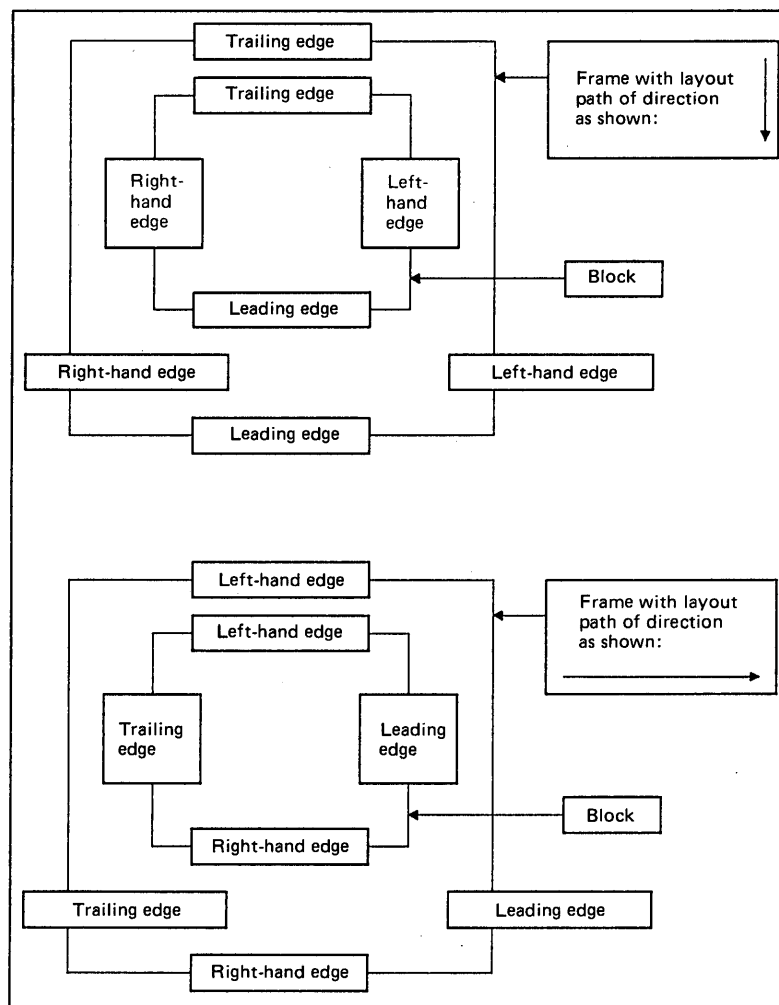
T0801610-87

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.

Border line

Border freespace

Border freespace width

Border line width

T0801630-87
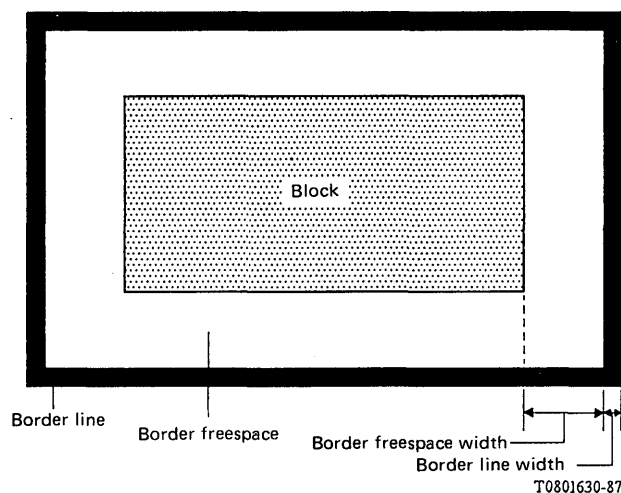
FIGURE 9/T.412

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.

Frame boundary

Border freespace

Border line

Maximum available area for
layout in frame

Border line width

Border freespace width

T0801620-87
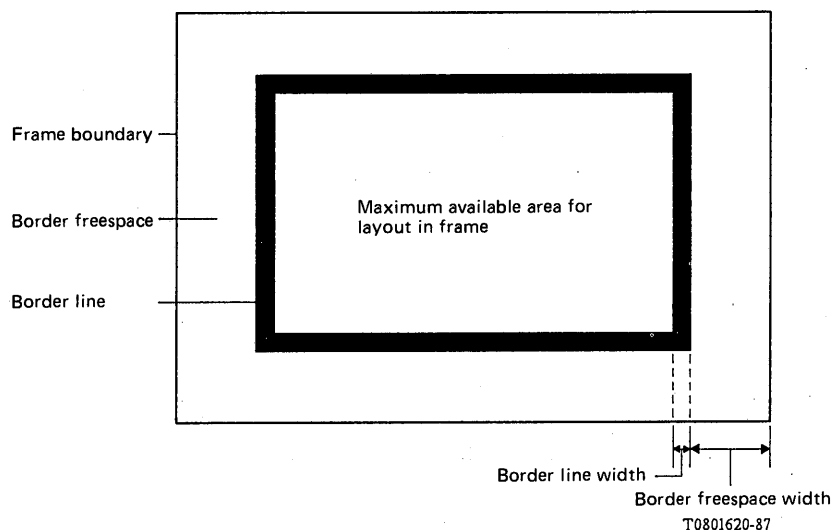
FIGURE 10/T.412

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.

```
                          Composite
                          logical
                          object


      Basic                 Basic                  Basic
      logical               logical                logical
      object                object                 object


    Content       Content          Content         Content
    portion       portion          portion         portion


                For example: page boundary


     Basic         Basic            Basic           Basic
     layout        layout           layout          layout
     object        object           object          object


        Composite                      Composite
        layout                         layout
        object                         object

                                              T0801640-87
```
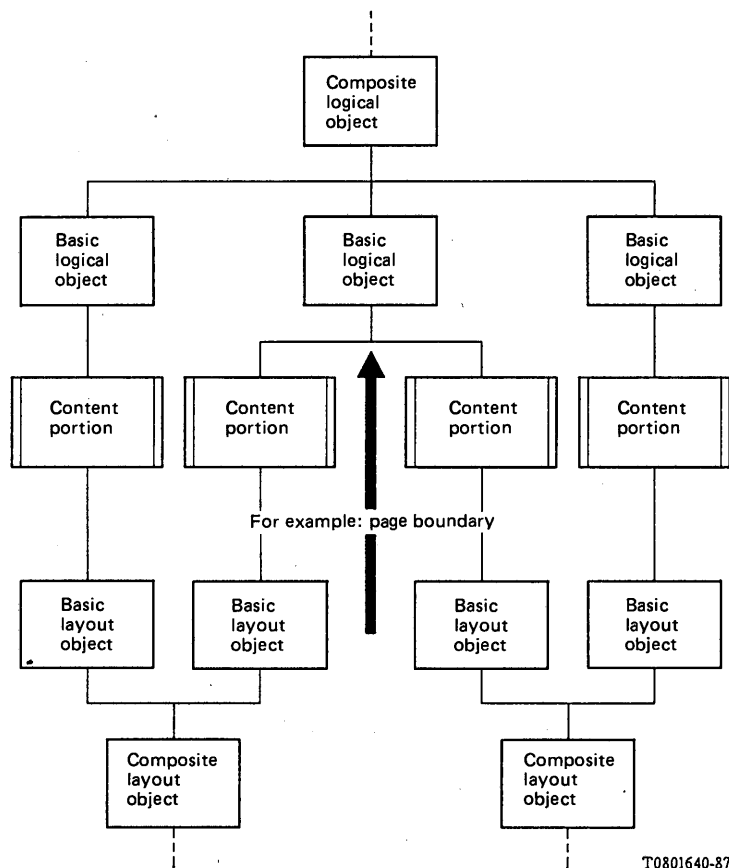
FIGURE 11/T.412

**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.

In the imaging process, the logical structure and referenced styles 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).

T0801650-87
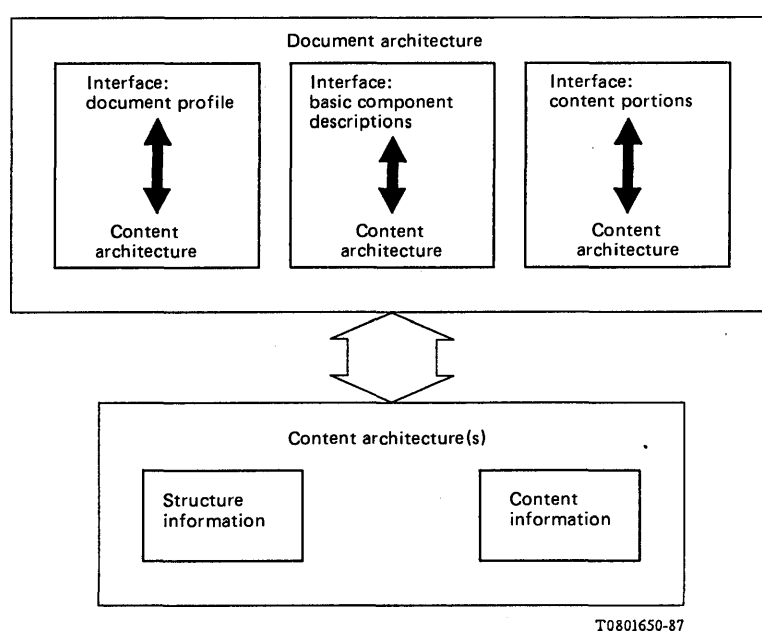
FIGURE 12/T.412

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   *Object identifier*

Constituents

Object descriptions.

Classification

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

Permissible values

A sequence of non-negative integers. The 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;

-    defaultable 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 sub-parameter. For each the permissible value is a non-negative integer.

- For the sub-parameter "separation", zero or more of "leading edge", "trailing edge", "centre separation" can be specified in any instance of this sub-parameter. For each the permissible value is a non-negative integer.

- For the sub-parameter "alignment", the permissible values are 'right-hand aligned', 'centred', 'left-hand aligned'.

- For the sub-parameter "fill order", the permissible values are 'normal order', 'reverse order'.

Default values

If 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 values 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', 'centred', '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 name 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°, '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 alwayss '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 sens 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.
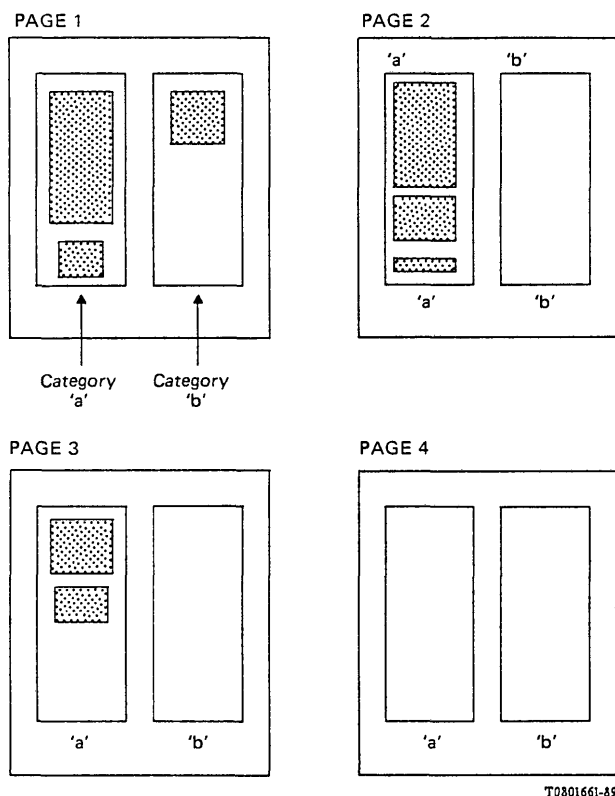
FIGURE 13/T.412

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 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, 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.
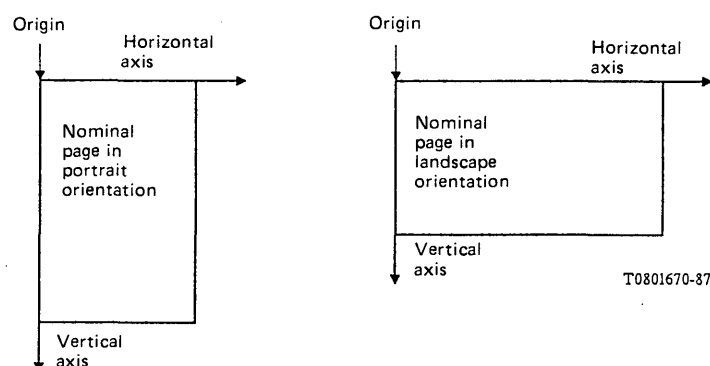
FIGURE 14/T.412

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).
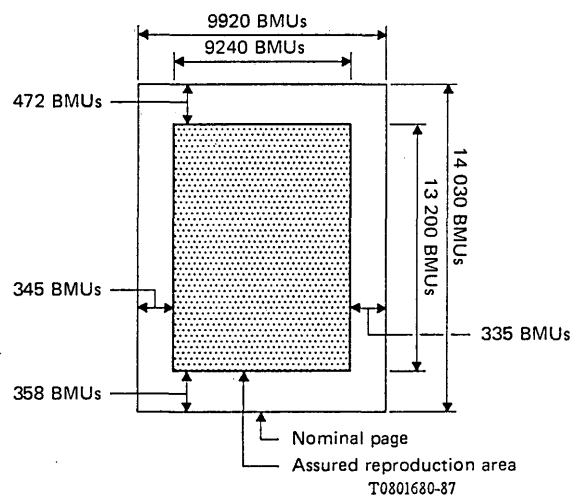
FIGURE 15/T.412

Dimensions and assured reproduction areas for ISO A4 page size
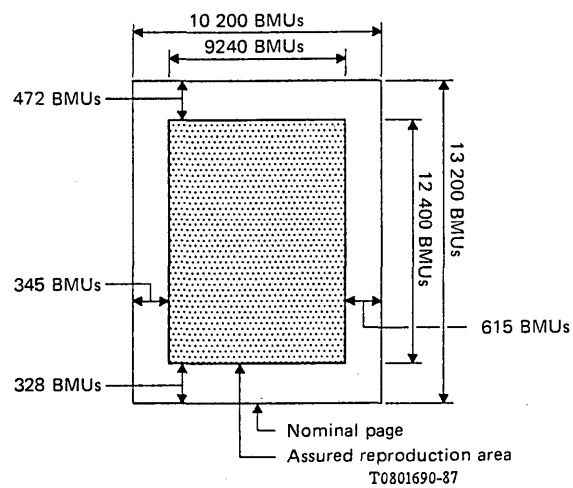
FIGURE 16/T.412

**Dimensions and assured reproduction areas
for North American Letter page size**
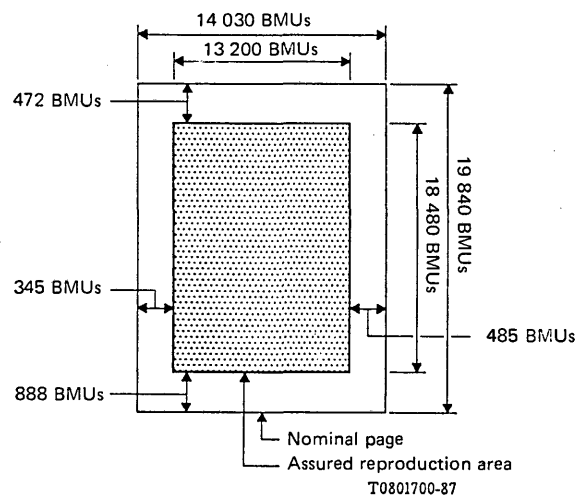
FIGURE 17/T.412

**Dimensions and assured reproduction areas for
ISO A3 page size**
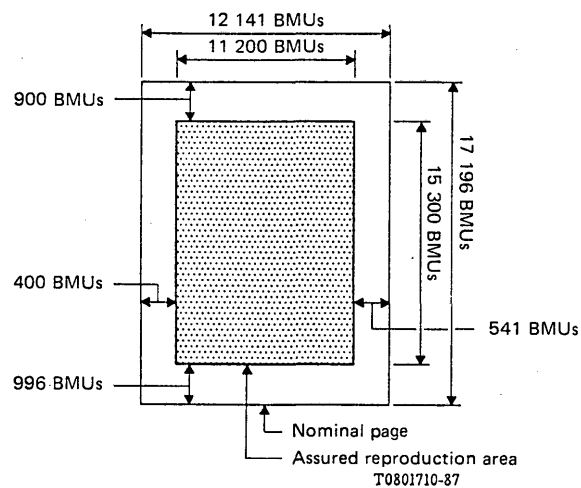
FIGURE 18/T.412

**Dimensions and assured reproduction areas for
Japanese legal page size**

8598 BMUs
7600 BMUs
900 BMUs
400 BMUs
1041 BMUs
12 141 BMUs
10 200 BMUs
Nominal page
Assured reproduction area
T0801720-87
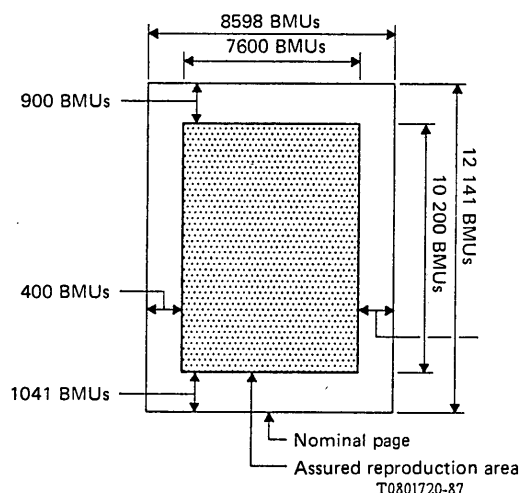
FIGURE 19/T.412

**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.

T0801730-87
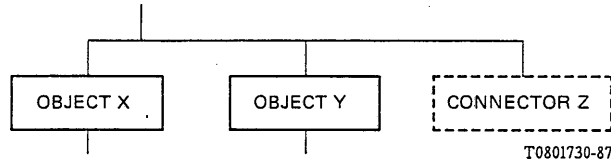
FIGURE A-1/T.412

Example of document structure notation: main structure

T0801740-87
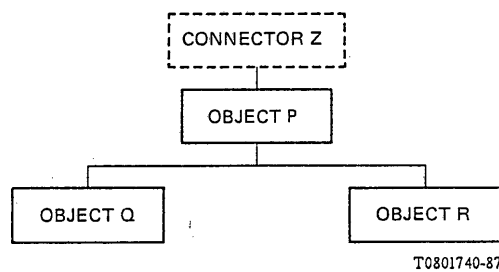
FIGURE A-2/T.412

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:

$$\backslash x$$

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.

CESSON, 26 JUNE 1985

To members of ISO/TC97/SC18/WG3

SUBJECT: PROPOSED EXAMPLE TO CLARIFY THE
DOCUMENT ARCHITECTURE MODEL

SUMMARY — SUMMARY — SUMMARY
SUMMARY — SUMMARY — SUMMARY — SUMMARY
SUMMARY — SUMMARY — SUMMARY — SUMMARY
SUMMARY — SUMMARY — SUMMARY — SUMMARY
SUMMARY — SUMMARY — SUMMARY — SUMMARY
SUMMARY — SUMMARY — SUMMARY — SUMMARY
SUMMARY — SUMMARY — SUMMARY — SUMMARY
SUMMARY — SUMMARY

T0801750-87

FIGURE B-1/T.412

Specimen document "letter" (1)

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA


BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

Y

4.12

1.75   X

caption-caption-caption-caption

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

FIGURE B-2/T.412

Specimen document "letter" (2)

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCC


        DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD·
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDD
```

```
        ENDING – FORMAL ENDING – FORMAL ENDING
FORMAL ENDING – FORMAL ENDING – FORMAL ENDING
FORMAL ENDING – FORMAL ENDING – FORMAL ENDING
FORMAL ENDING – FORMAL ENDING – FORMAL ENDING
FORMAL ENDING – FORMAL ENDING – FORMAL ENDING
FORMAL ENDING
```

Miss Aude HEA

Document Architect

T0801770-87

FIGURE B-3/T.412

Specimen document "letter" (3)

HEADER PAGE   T0801780-87
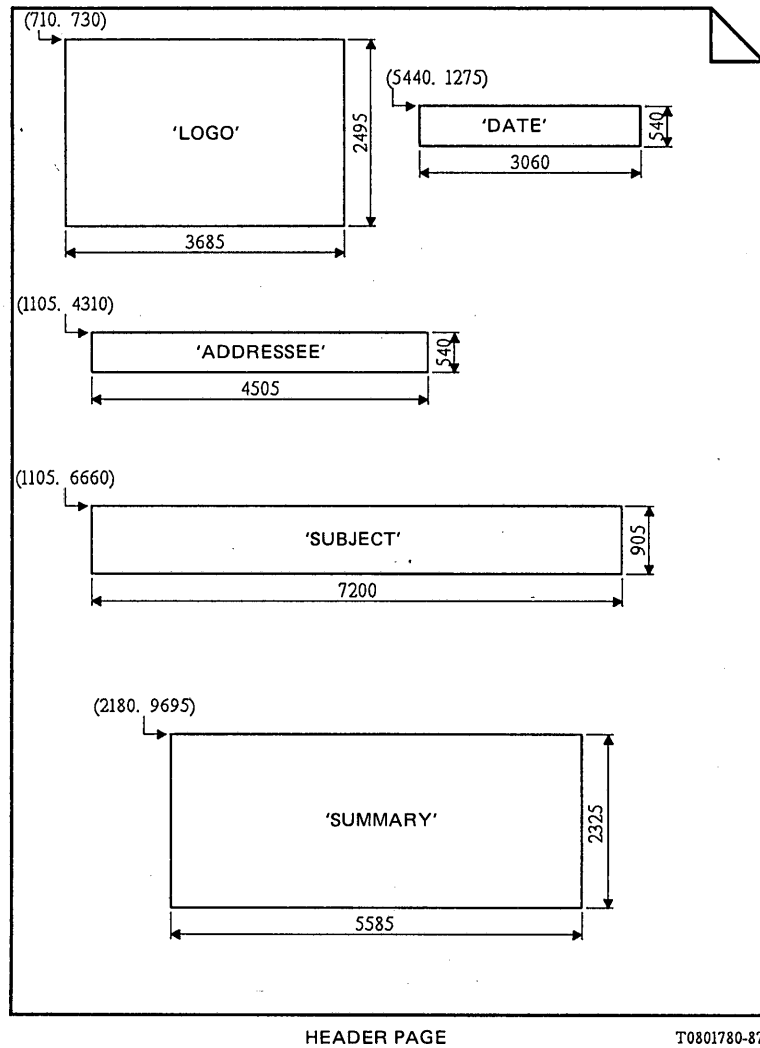
FIGURE B-4/T.412

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

FIRST BODY PAGE   T0801790-87
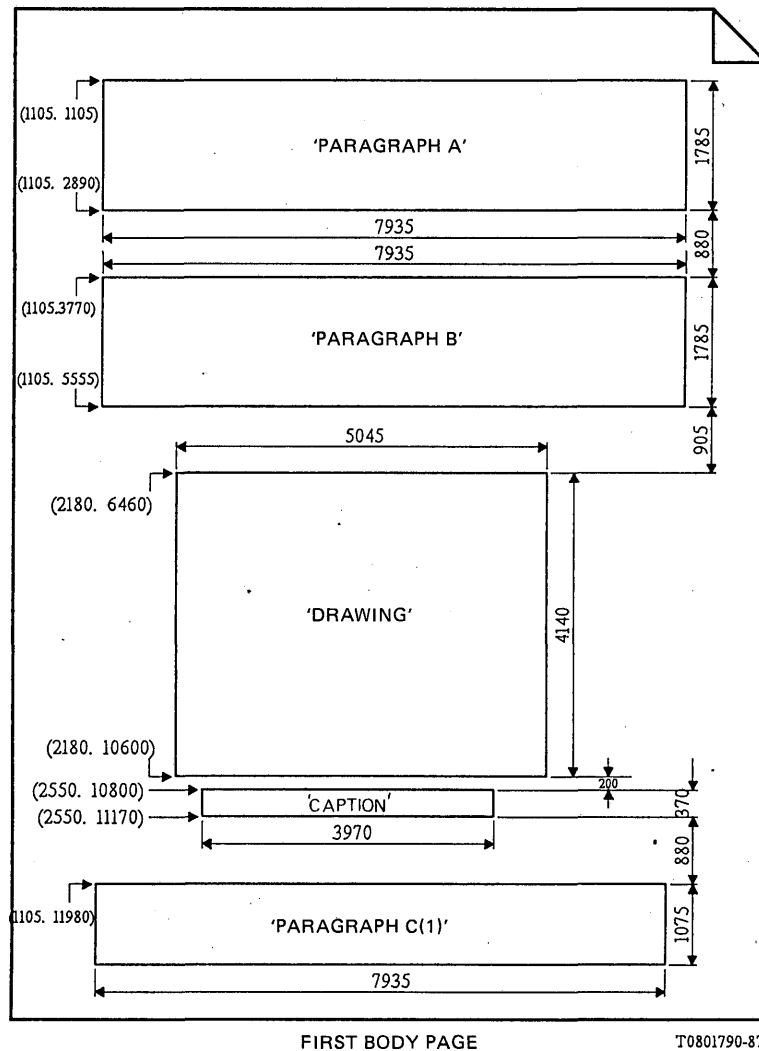
FIGURE B-5/T.412

Layout of "letter" showing pages and blocks (2)
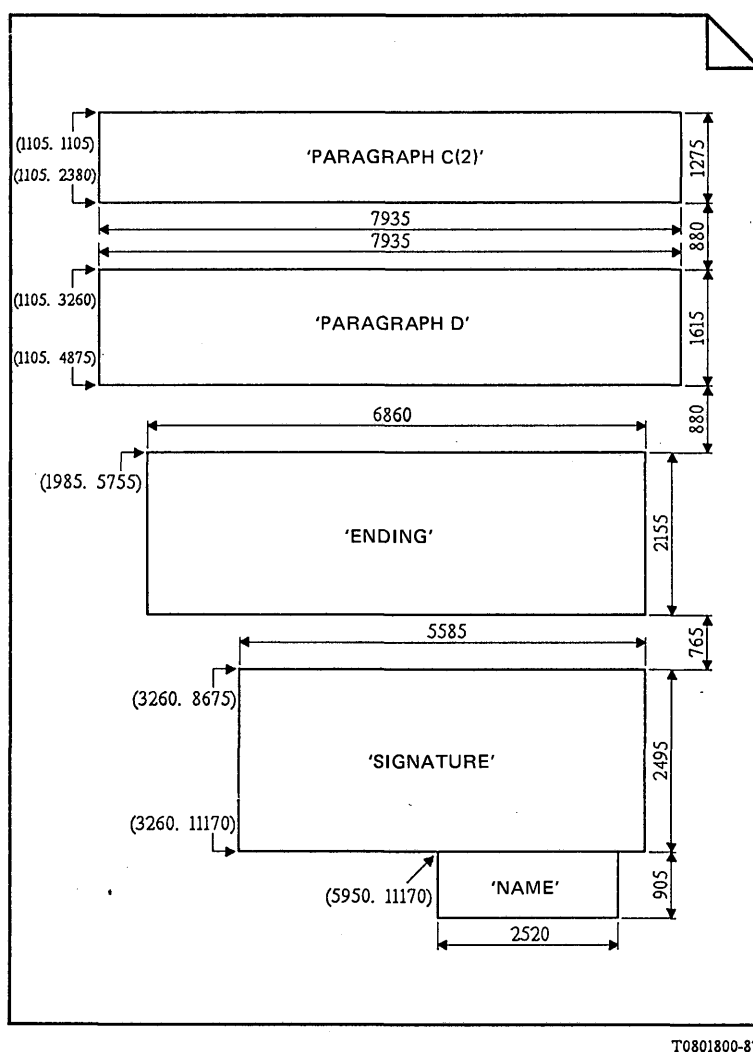
FIGURE B-6/T.412

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).
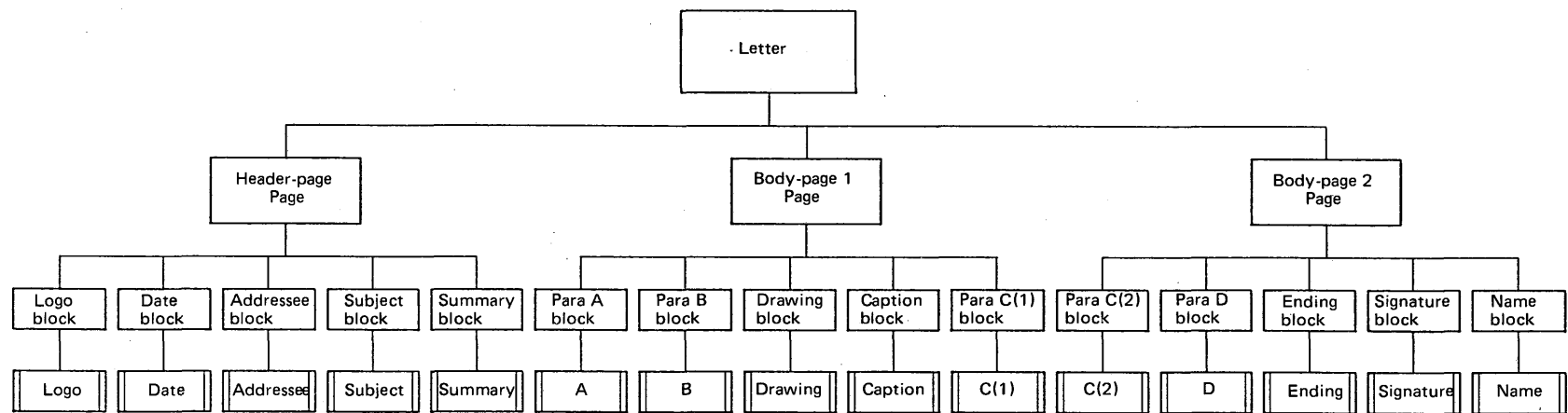
FIGURE B-7/T.412

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"

Object type	BLOCK
Object identifier	1 0 3
User-visible name	"Subject"
Position	HP = 1105, VP = 6660
Dimensions	HD = 7200, VD = 905
Line Spacing	300
Content Portions	0
Content Identifier - Layout	1 0 3 0
Content Information	"SUBJECT: PROPOSED EXAMPLE TO CLARIFY THE DOCUMENT\n ARCHITECTURE MODEL"
Object type	BLOCK
Object identifier	1 0 4
User-visible name	"Summary"
Position	HP = 2180, VP = 9695
Dimensions	HD = 5585, VD = 2325
Alignment	JUSTIFIED
Content Portions	0
Content identifier - Layout	1 0 4 0
Content information	/*Formatted string of SUMMARY-*/
Object type	PAGE
Object identifier	1 1
User-visible name	"Body page 1"
Dimensions	HD = 9920, VD = 14030
Subordinates	0, 1, 2, 3, 4
Object type	BLOCK
Object identifier	1 1 0
User-visible name	"Para A"
Position	HP = 1105, VP = 1105
Dimensions	HD = 7935, VD = 1785
Line spacing	300
Alignment	JUSTIFIED
Content portions	0
Content identifier - Layout	1 1 0 0
Content information	/*Formatted string of A's*/
Object type	BLOCK
Object identifier	1 1 1
User-visible name	"Para B"
Position	HP = 1105, VP = 3770
Dimensions	HD = 7935, VD = 1785
Line spacing	400
Alignment	JUSTIFIED
Content portions	0
Content identifier - Layout	1 1 1 0
Content information	/*Formatted string of B's*/
Object type	BLOCK
Object identifier	1 1 2
User-visible name	"Drawing"
Position	HP = 2180, VP = 6460
Dimensions	HD = 5045, VD = 4140
Content architecture class	FORMATTED GEOMETRIC GRAPHICS
Content portions	0

Content identifier - Layout Content information	1 1 2 0 /*Ordered set of geometric-graphics content elements for the diagram*/
Object type Object identifier User-visible name Position Dimensions Content portions	BLOCK 1 1 3 "Caption" HP = 2550, VP = 10800 HD = 3970, VD = 370 0
Content identifier - Layout Content information	1 1 3 0 /*Formatted string for the caption*/
Object type Object identifier User-visible name Position Dimensions Line spacing Alignment Content portions	BLOCK 1 1 4 "Para C(1)" HP = 1105, VP = 11980 HD = 7935, VD = 1075 300 JUSTIFIED 0
Content identifier - Layout Content information	1 1 4 0 /*Formatted string of C's*/
Object type Object identifier User-visible name Dimensions Subordinates	PAGE 1 2 "Body Page 2" HD = 9920, VD = 14030 0, 1, 2, 3, 4
Object type Object identifier User-visible name Position Dimensions Line spacing Alignment Content portions	BLOCK 1 2 0 "Para C(2)" HP = 1105, VP = 1105 HD = 7935, VD = 1275 300 JUSTIFIED 0
Content identifier = Layout Content information	1 2 0 0 /*Formatted string of C's*/
Object type Object identifier User-visible name Position Dimensions Line spacing Alignment Content portions	BLOCK 1 2 1 "Para D" HP = 1105, VP = 3260 HD = 7935, VD = 1615 300 JUSTIFIED 0
Content identifier = Layout Content information	1 2 1 0 /*Formatted string of D's*/

Object type Object identifier User-visible name Position Dimensions Line spacing Alignment Content portions	BLOCK 1 2 2 "Ending" HP = 1985, VP = 5755 HD = 6860, VD = 2155 300 JUSTIFIED 0
Content identifier - Layout Content information	1 2 2 0 /*Formatted string of FORMAL ENDING*/
Object type Object identifier User-visible name Position Dimensions Content architecture class Content portions	BLOCK 1 2 3 "Signature" HP = 3260, VP = 8675 HD = 5585, VD = 2495 FORMATTED RASTER GRAPHICS 0
Content identifier - Layout Content information	1 2 3 0 /*Array of raster-graphics content elements for the signature*/
Object type Object identifier User-visible name Position Dimensions Line spacing Content portions	BLOCK 1 2 4 "Name" 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.

```
                            ┌──────────────┐
                            │    Letter    │
                            └──────────────┘
                    ┌──────────────┴────────────────────────────────────────────┐
              ┌──────────┐                                              ┌──────────┐
              │  Header  │                                              │   Body   │
              └──────────┘                                              └──────────┘
```
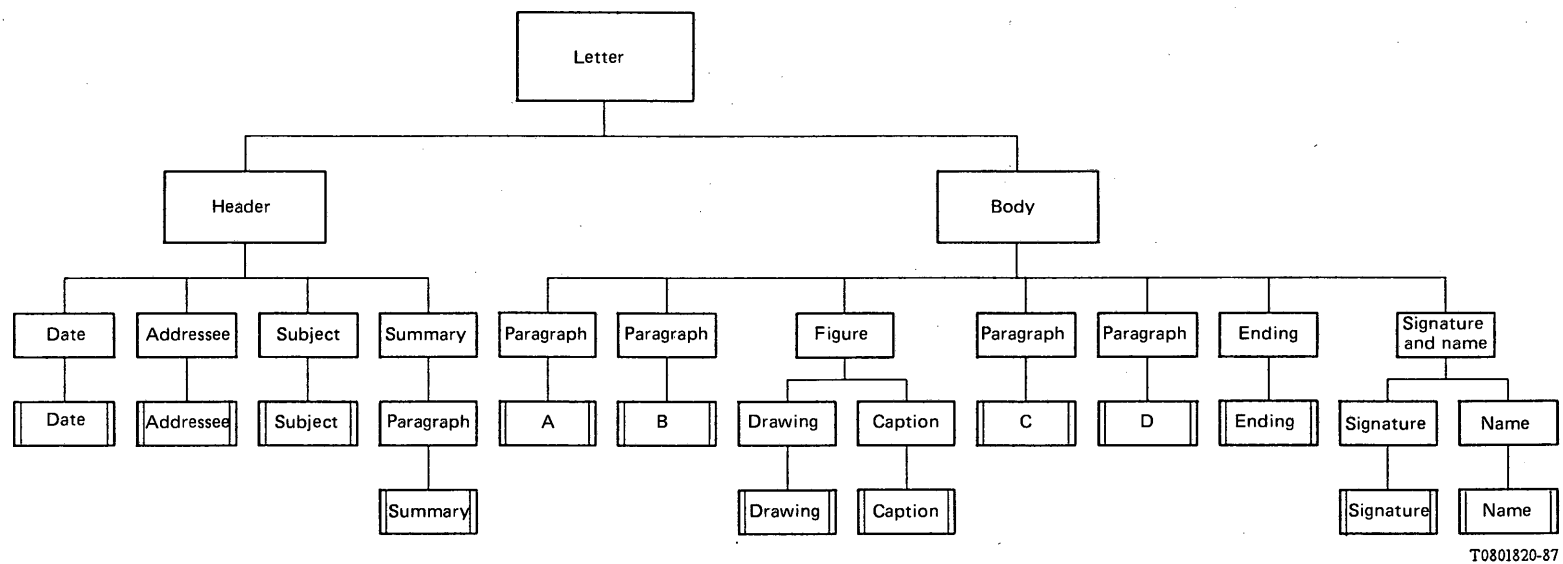

FIGURE B-8/T.412

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	DOCUMENT LOGICAL ROOT
Object identifier	3
User-visible name	"Letter"
Subordinates	0, 1
Default value lists	list for basic logical objects: attribute: content architecture class value: 'processable character'

Object type	COMPOSITE LOGICAL
Object identifier	3 0
User-visible name	"Header"
Subordinates	0, 1, 2, 3

Object type	BASIC LOGICAL
Object identifier	3 0 0
User-visible name	"Date"
Content portions	0

Content identifier - Logical	3 0 0 0
Content information	"CESSON 26 JUNE 1985"

Object type	BASIC LOGICAL
Object identifier	3 0 1
User-visible name	"Addressee"
Content portions	0

Content identifier - Logical	3 0 1 0
Content information	"To members of ISO/TC97/SC18/WG3"

Object type	BASIC LOGICAL
Object identifier	3 0 2
User-visible name	"Subject"
Presentation style	5 0
Content portions	0

Content identifier - Logical	3 0 2 0
Content information	"SUBJECT: PROPOSED EXAMPLE TO CLARIFY THE DOCUMENT ARCHITECTURE MODEL"

Object type	COMPOSITE LOGICAL
Object identifier	3 0 3
User-visible name	"Summary"
Subordinates	0

Object type 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

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"

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.

T0801830-87
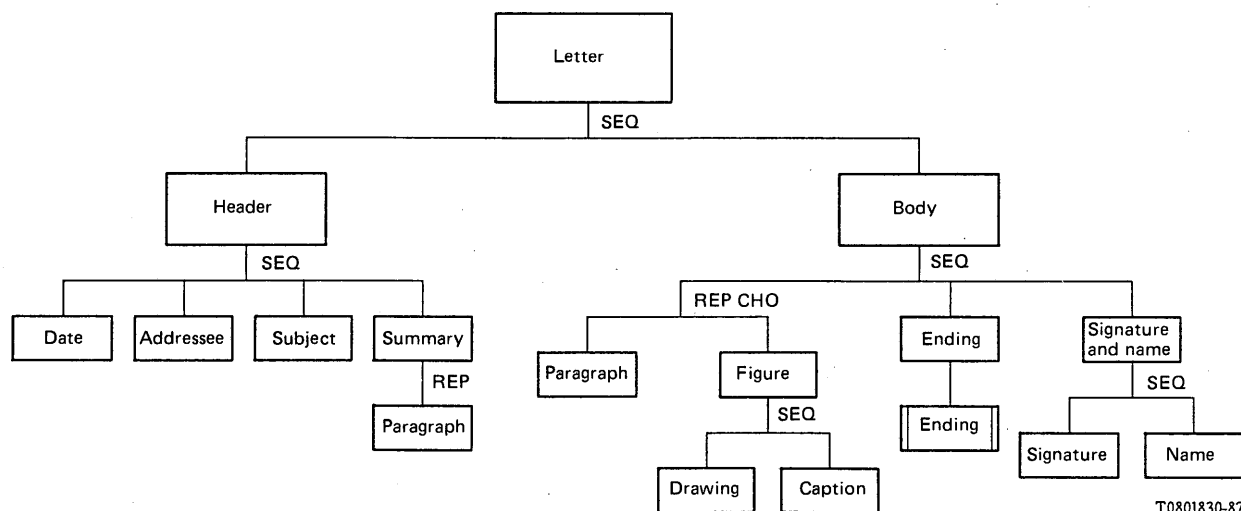
FIGURE B-9/T.412

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).

T0801841-89
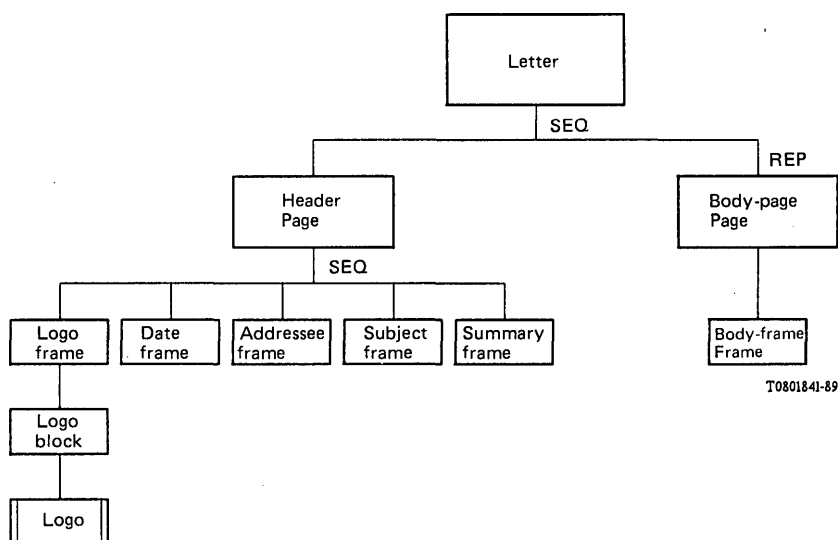
FIGURE B-10/T.412

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.

Object type Object class identifier User-visible name Generator for subordinates	DOCUMENT LOGICAL ROOT 2 (Letter) "Letter" SEQ (Header, body)
Object type Object class identifier User-visible name Generator for subordinates	COMPOSITE LOGICAL 2 0 (Header) "Header" SEQ (Date, addressee, subject, summary)
Object type Object class identifier User-visible name Layout style Content architecture class	BASIC LOGICAL 2 0 0 (Date) "Date" 4 0 PROCESSABLE CHARACTERS
Object type Object class identifier User-visible name Layout style Content architecture class	BASIC LOGICAL 2 0 1 (Addressee) "Addressee" 4 1 PROCESSABLE CHARACTERS
Object type Object class identifier User-visible name Layout style Presentation style Content architecture class	BASIC LOGICAL 2 0 2 (Subject) "Subject" 4 2 5 0 PROCESSABLE CHARACTERS
Object type Object class identifier User-visible name Layout style Generator for subordinates	COMPOSITE LOGICAL 2 0 3 (Summary) "Summary" 4 3 REP Summary-paragraph
Object type Object class identifier User-visible name Layout style Presentation style Content architecture class	BASIC LOGICAL 2 0 3 1 (Summary-paragraph) "Summary-paragraph" 4 4 5 1 PROCESSABLE CHARACTERS
Object type Object class identifier User-visible name Layout style Generator for subordinates	COMPOSITE LOGICAL 2 1 (Body) "Body" 4 5 SEQ (REP CHO) (Body-paragraph, figure, ending, signature-and-name)
Object type Object class identifier User-visible name Layout style Generator for subordinates	COMPOSITE LOGICAL 2 1 0 (Figure) "Figure" 4 6 SEQ (Drawing, caption)

Object type	BASIC LOGICAL
Object class identifier	2 1 0 0 (Drawing)
User-visible name	"Drawing"
Content architecture class	FORMATTED PROCESSABLE GEOMETRIC GRAPHICS
Layout style	4 7
Object type	BASIC LOGICAL
Object class identifier	2 1 0 1 (Caption)
User-visible name	"Caption"
Layout style	⸀ 4 8
Content architecture class	PROCESSABLE CHARACTERS
Object type	BASIC LOGICAL
Object class identifier	2 1 1 (Body-paragraph)
User-visible name	"Body-paragraph"
Layout style	4 9
Presentation style	5 3
Content architecture class	PROCESSABLE CHARACTERS
Object type	BASIC LOGICAL
Object class identifier	2 1 2 (Ending)
User-visible name	"Ending"
Layout style	4 10
Presentation style	5 4
Content portions	0 (Ending-content)
Content architecture class	PROCESSABLE CHARACTERS
Content identifier - Logical	2 1 2 0 (Ending-content)
Content information	/*Unformatted string for ending*/
Object type	COMPOSITE LOGICAL
Object class identifier	2 1 3 (Signature-and-Name)
User-visible name	"Signature-and-name"
Generator for subordinates	SEQ (Signature, name)
Object type	BASIC LOGICAL
Object class identifier	2 1 3 0 (Signature)
User-visible name	"Signature"
Content architecture class	PROCESSABLE FORM RASTER GRAPHICS
Layout style	4 11
⸀Object type	BASIC LOGICAL
Object class identifier	2 1 3 1 (Name)
User-visible name	"Name"
Layout style	4 12
Presentation style	5 0
Content architecture class	PROCESSABLE CHARACTERS

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

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

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

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.
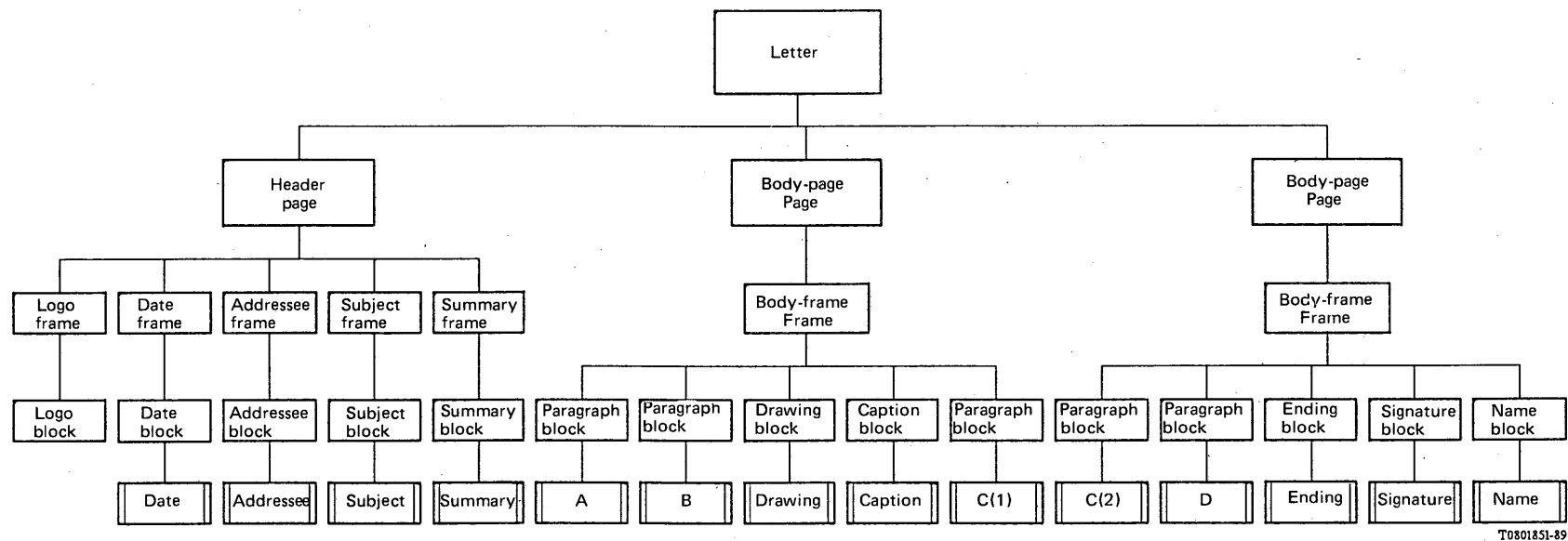
FIGURE B-11/T.412

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	DOCUMENT LAYOUT ROOT Object
Identifier	1
Object class	0 (Letter)
User-visible name	"Letter"
Subordinates	0, 1, 2
Object type	PAGE
Object identifier	1 0
Object class	0 0 (Header)
User-visible name	"Header"
Subordinates	0, 1, 2, 3, 4
Object type	FRAME
Object identifier	1 0 0
Object class	0 0 0 (Logo Frame)
Subordinates	0
Object type	BLOCK
Object identifier	1 0 0 0
Object class	0 0 0 0 (Logo)
User-visible name	"Logo"
Content architecture class	FORMATTED FORM RASTER GRAPHICS
Object type	FRAME
Object identifier	1 0 1
Object class	0 0 1 (Date)
User-visible name	"Date"
Subordinates	0
Object type	BLOCK
Object identifier	1 0 1 0
Position	HP = 395, VP = 710
Dimensions	HD = 3060, VD = 540
Content portions	0
Content architecture class	FORMATTED-PROCESSABLE CHARACTERS

Object type	FRAME
Object identifier	1 0 2
Object class	0 0 2 (Addressee)
User-visible name	"Addressee"
Subordinates	0

Object type	BLOCK
Object identifier	1 0 2 0
Dimensions	HD = 4505, VD = 540
Content portions	0
Content architecture class	FORMATTED-PROCESSABLE CHARACTERS

Object type	FRAME
Object identifier	1 0 3
Object class	0 0 3 (Subject)
User-visible name	"Subject"
Subordinates	0

Object type	BLOCK
Object identifier	1 0 3 0
Dimensions	HD = 7200, VD = 905
Presentation style	5 0
Content portions	0
Content architecture class	FORMATTED-PROCESSABLE CHARACTERS

Object type	FRAME
Object identifier	1 0 4
Object class	0 0 4 (Summary)
User-visible name	"Summary"
Subordinates	0

Object type	BLOCK
Object identifier	1 0 4 0
Dimensions	HD = 5585, VD = 2325
Presentation style	5 1
Content portions	0
Content architecture class	FORMATTED-PROCESSABLE CHARACTERS

Object type	PAGE
Object identifier	1 1
Object class	0 1 (Body-page)
User-visible name	"Body"
Subordinates	0

Object type	FRAME
Object identifier	1 1 0
Object class	0 1 0 (Body-frame)
User-visible name	"Body"
Subordinates	0, 1, 2, 3, 4

Object type	BLOCK
Object identifier	1 1 0 0
Position	HP = 540, VP = 540
Dimensions	HD = 7935, VD = 1785
Presentation style	5 3
Content portions	0
Content architecture class	FORMATTED-PROCESSABLE CHARACTERS

Object type	BLOCK
Object identifier	1 1 0 1
Position	HP = 540, VP = 3205
Dimensions	HD = 7935, VD = 1785
Presentation style	5 3
Line spacing	400
Content portions	0
Content architecture class	FORMATTED-PROCESSABLE CHARACTERS
Object type	BLOCK
Object identifier	1 1 0 2
Position	HP = 1615, VP = 5895
Dimensions	HD = 5045, VD = 4140
Content architecture class	FORMATTED-PROCESSABLE GEOMETRIC GRAPHICS
Content portions	0
Object type	BLOCK
Object identifier	1 1 0 3
Position	HP = 1985, VP = 10235
Dimensions	HD = 3970, VD = 370
Content portions	0
Content architecture class	FORMATTED-PROCESSABLE CHARACTERS
Object type	BLOCK
Object identifier	1 1 0 4
Position	HP = 540, VP = 11485
Dimensions	HD = 7935, VD = 1075
Presentation style	5 3
Content portions	0
Content architecture class	FORMATTED-PROCESSABLE CHARACTERS
Object type	PAGE
Object identifier	1 2
Object class	0 1 (Body-page)
User-visible name	"Body"
Subordinates	0
Object type	FRAME
Object identifier	1 2 0
Object class	0 1 0 (Body-frame)
User-visible name	"Body"
Subordinates	0, 1, 2, 3, 4
Object type	BLOCK
Object identifier	1 2 0 0
Position	HP = 540, VP = 540
Dimensions	HD = 7935, VD = 1275
Presentation style	5 3
Content portions	0
Content architecture class	FORMATTED-PROCESSABLE CHARACTERS
Object type	BLOCK
Object identifier	1 2 0 1
Position	HP = 540, VP = 2695
Dimensions	HD = 7935, VD = 1615
Presentation style	5 3
Content portions	0
Content architecture class	FORMATTED-PROCESSABLE CHARACTERS

Object type Object identifier Position Dimensions Presentation style Content portions Content architecture class	BLOCK 1 2 0 2 HP = 1820, VP = 5190 HD = 6860, VD = 2155 5 4 0 FORMATTED-PROCESSABLE CHARACTERS
Object type Object identifier Position Dimensions Content type Content portions	BLOCK 1 2 0 3 HP = 2695, VP = 8110 HD = 5585, VD = 2495 FORMATTED-PROCESSABLE RASTER GRAPHICS 0
Object type Object identifier Position Dimensions Presentation style Content portions Content architecture class	BLOCK 1 2 0 4 HP = 5385, VP = 10605 HD = 2520, VD = 905 5 0 0 FORMATTED-PROCESSABLE CHARACTERS
Content identifier - Layout Content identifier - Logical Content information	1 0 1 0 0 3 0 0 0 "CESSON, 26 JUNE 1985"
Content identifier - Layout Content identifier - Logical Content information	1 0 2 0 0 3 0 1 0 "To members of ISO/TC97/SC18/WG3"
Content identifier - Layout Content identifier - Logical Content information	1 0 3 0 0 3 0 2 0 "SUBJECT: PROPOSED EXAMPLE TO CLARIFY THE DOCUMENT\n ARCHITECTURE MODEL"
Content identifier - Layout Content identifier - Logical Content information	1 0 4 0 0 3 0 3 0 0 /*Formatted string of SUMMARY-*/
Content identifier - Layout Content identifier - Logical Content information	1 1 0 0 0 3 1 0 0 /*Formatted string of A's*/
Content identifier - Layout Content identifier - Logical Content information	1 1 0 1 0 3 1 1 0 /*Formatted string of B's*/
Content identifier - Layout Content identifier - Logical Content information	1 1 0 2 0 3 1 2 0 0 /*Ordered set of geometric-graphics content elements for the diagram*/.
Content identifier - Layout Content identifier - Logical Content information	1 1 0 3 0 3 1 2 1 0 /*Formatted string for the caption*/
Content identifier - Layout Content identifier - Logical Content information	1 1 0 4 0 3 1 3 0 /*Formatted string of C's*/.

Content identifier - Layout Content identifier - Logical Content information	1 2 0 0 0 3 1 3 1 /*Formatted string of C's*/-
Content identifier - Layout Content identifier - Logical Content information	1 2 0 1 0 3 1 4 0 /*Formatted string of D's*/
Content identifier - Layout Content information	1 2 0 2 0 /*Formatted string of ENDING*/
Content identifier - Layout Content identifier - Logical Content information	1 2 0 3 0 3 1 6 0 0 /*Array of raster-graphics content elements for the signature*/.
Content identifier - Layout Content identifier - Logical Content information	1 2 0 4 0 3 1 6 1 0 "Miss Aude HEA\nDocument architect"

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.
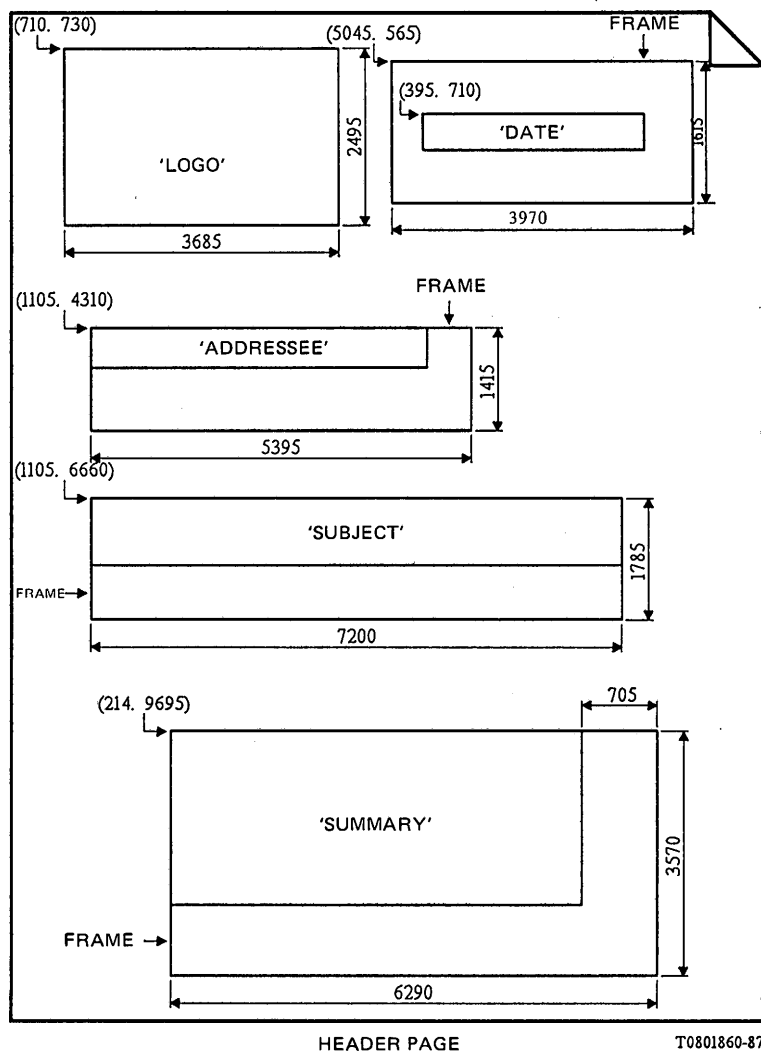
FIGURE B-12/T.412

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

FIRST BODY PAGE

T0801871-89
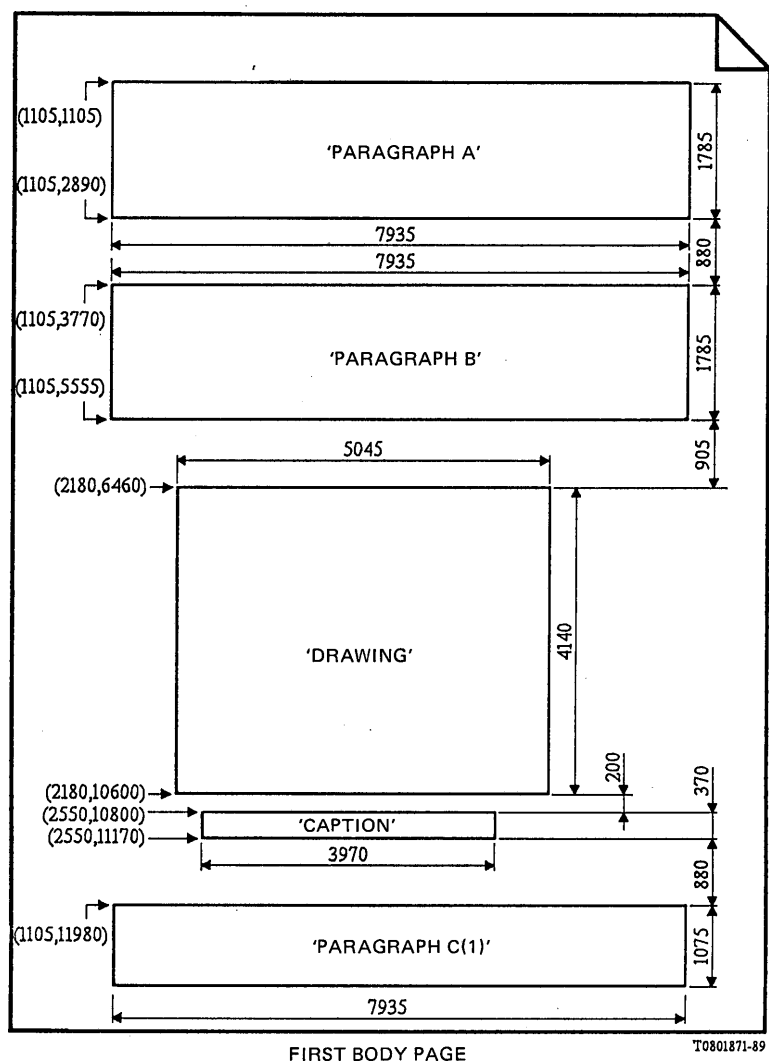
FIGURE B-13/T.412

Layout structure of "letter" showing "pages", "frames" and "blocks" (2)
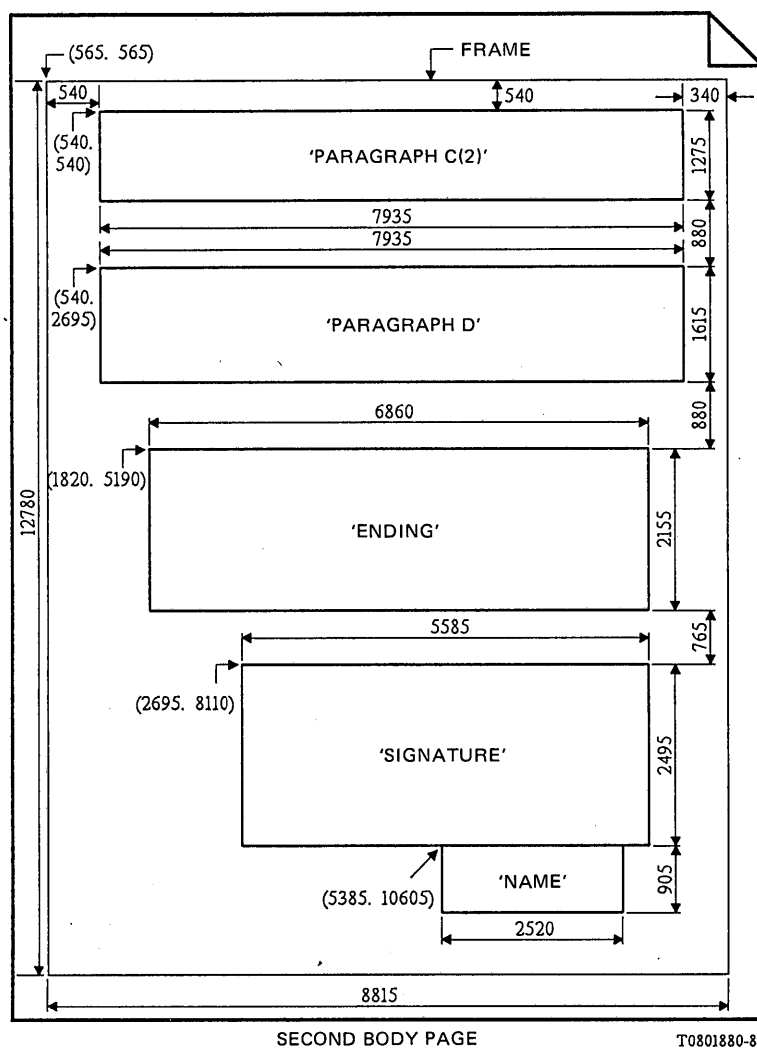
FIGURE B-14/T.412

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.

C.2 *Attributes applicable to FDA document architecture levels*

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 level   NM Attribute non-mandatory at this level
D Attribute defaultable at this level
\<blank entry\> Not applicable

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

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

F1

Reference
point of F1

B

A

F11

VP = A
HP = B

Reference
point of F11

T0801891-89
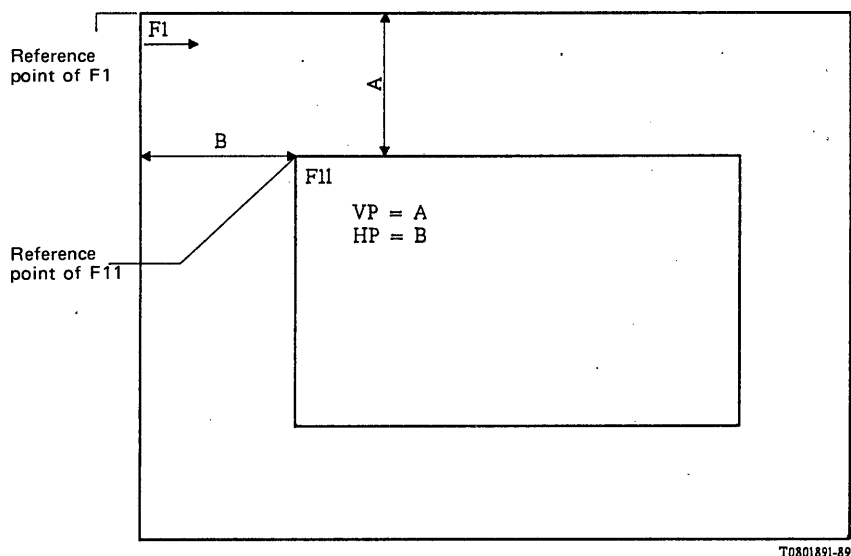
FIGURE D-1/T.412

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
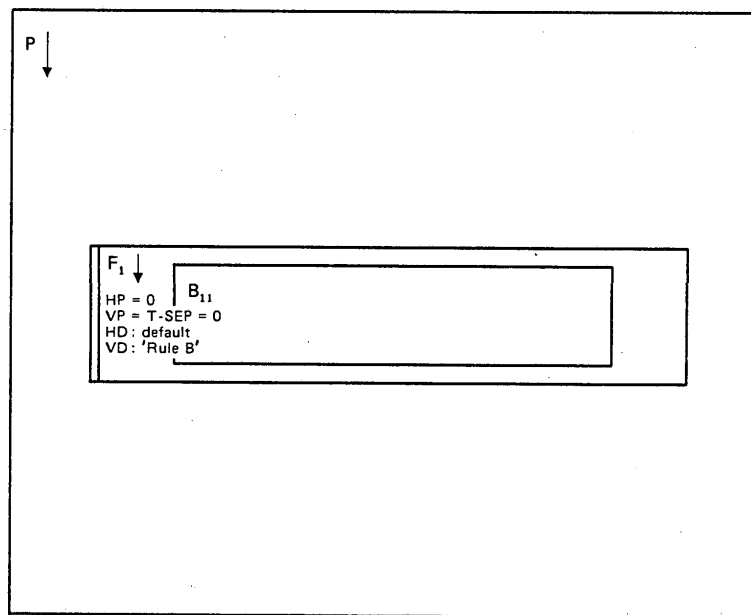
Illustration

T0801900-87

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.
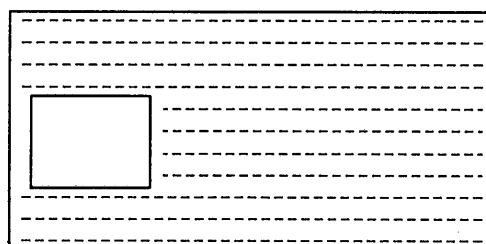
T0801910-87

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 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
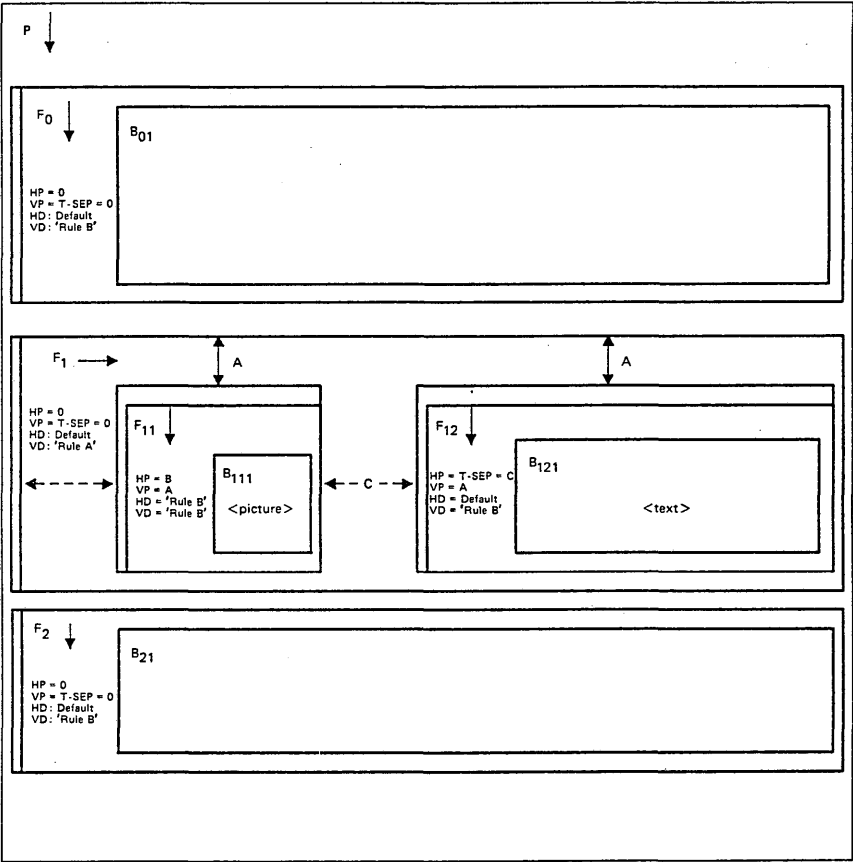
T0801920-87

FIGURE D-3/T.412

"Wrapping" of text around a left aligned picture

Variations

1)   If the picture is required to be right aligned:
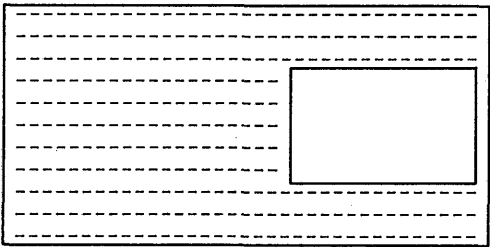
T0801930-87

then the layout path for frame F1 would be reversed (180°), 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:
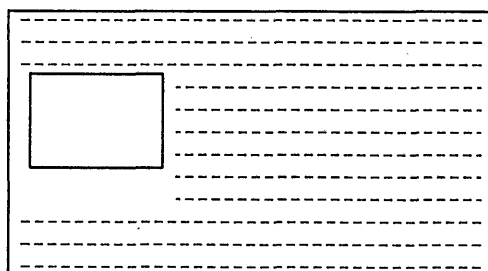
T0801940-87

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:
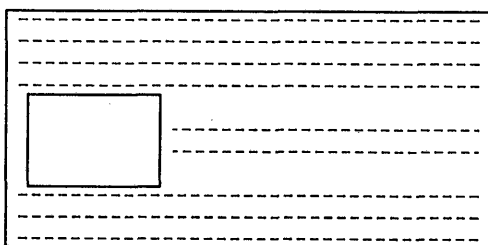
T0801950-87

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.

Figure 1
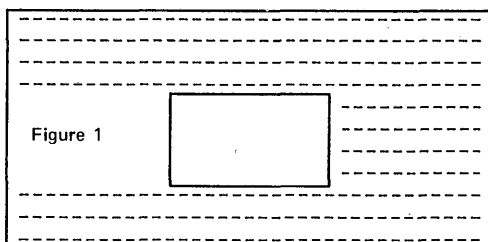
T0801960-87

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
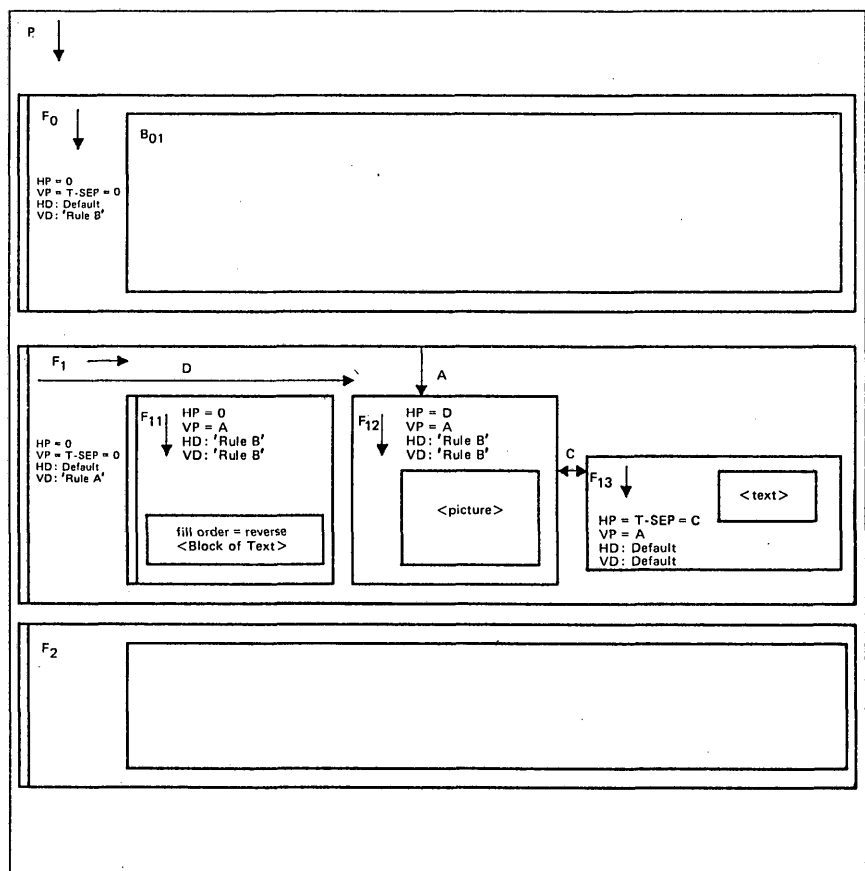
FIGURE D-4/T.412

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.
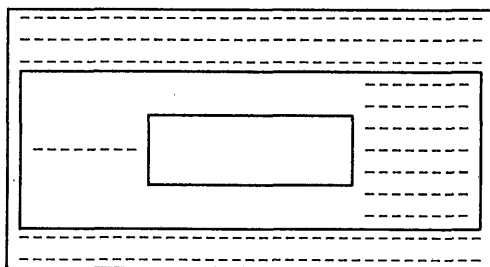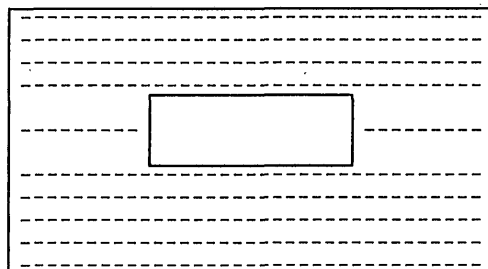
T0801980-87

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.
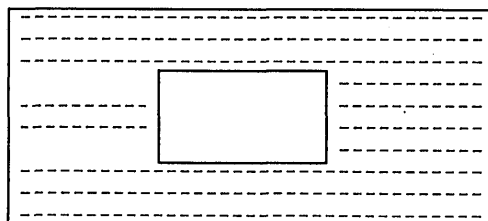
T0801990-87

    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.
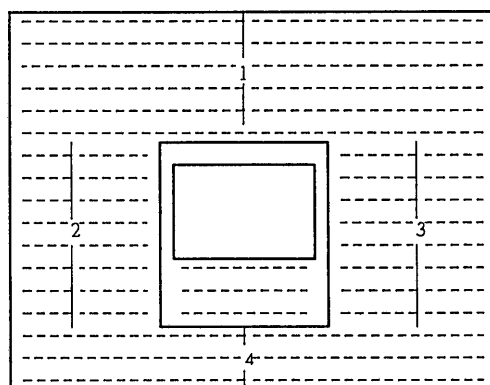
T0802000-87

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.
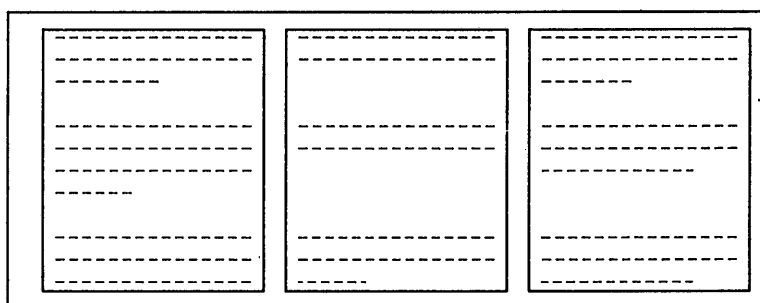
T0802010-87

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° and a number of subordinate frames would be specified, one for each line of Japanese text to be placed in this column. These frames would be balanced, within each of these subordinate frames the layout path is 270°. Therefore the Japanese text would take up as little space in the vertical dimension as possible.

Illustration

T0802021-89
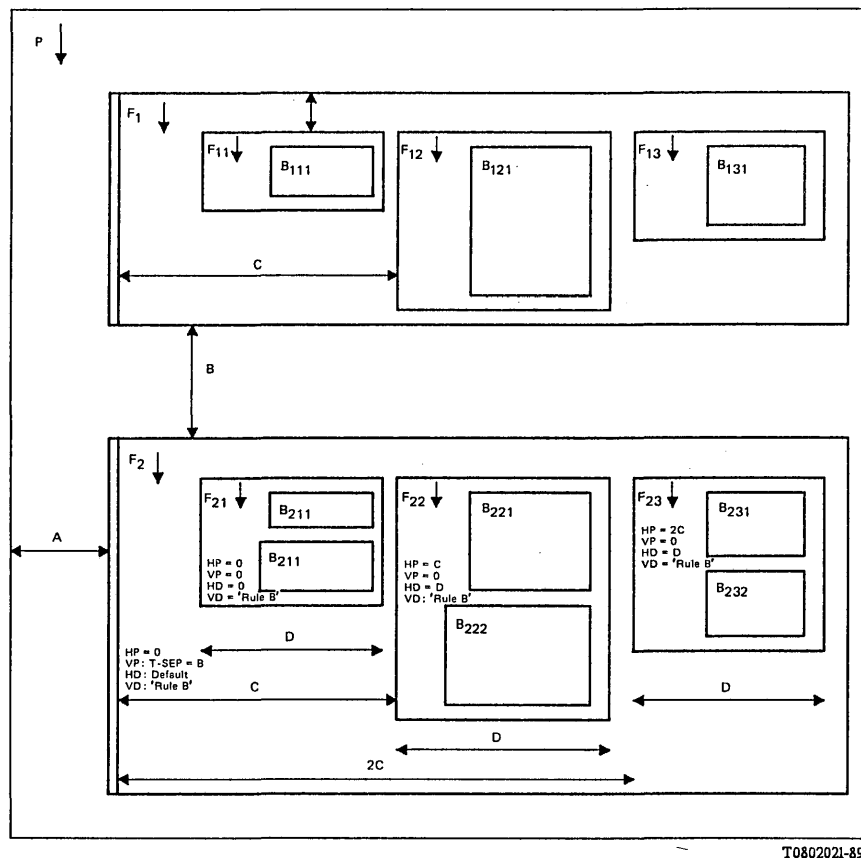
FIGURE D-5/T.412

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

T0802030-87
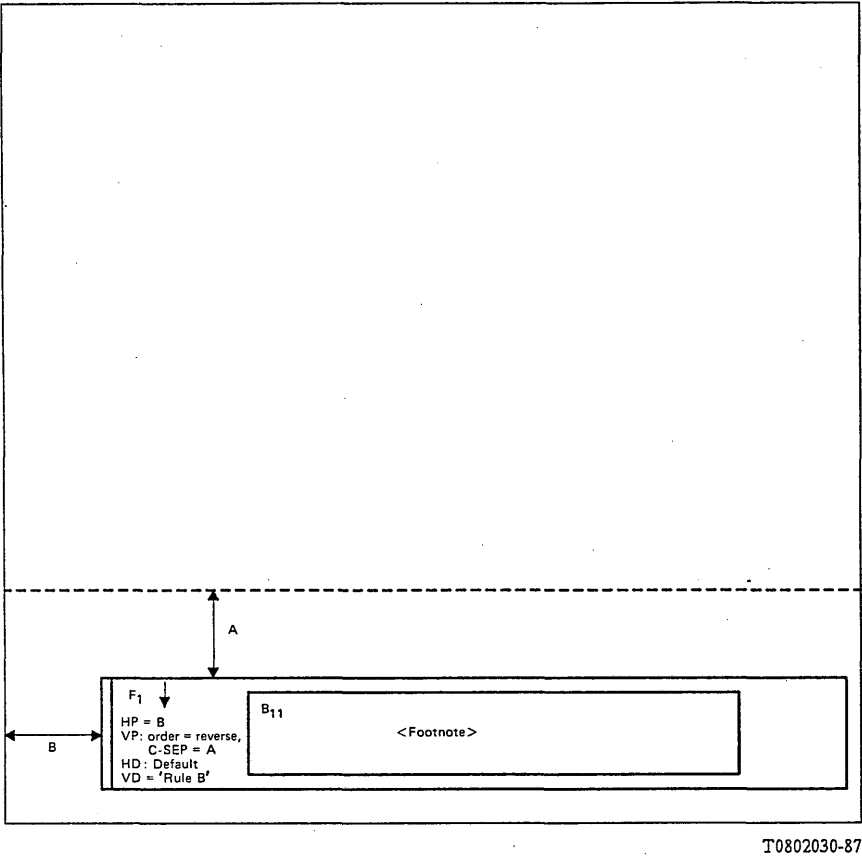
FIGURE D-6/T.412

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 ...)
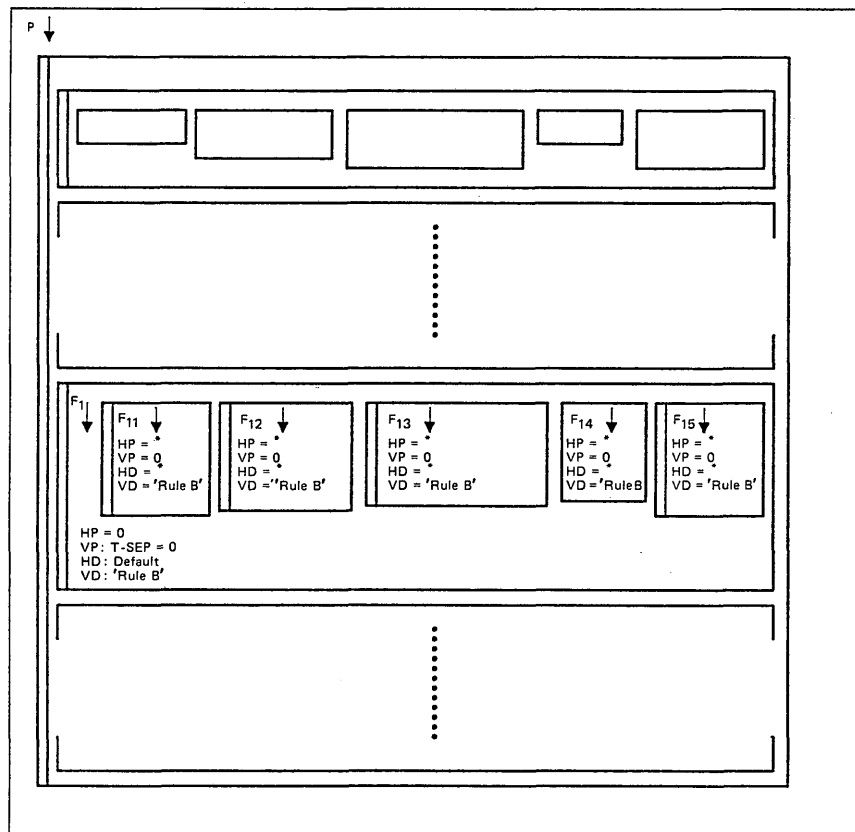
T0802040-87

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.
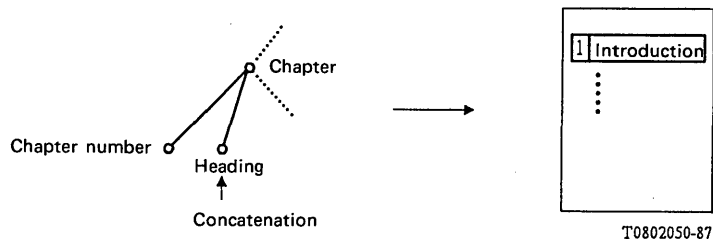
FIGURE D-8/T.412

Concatenation of chapter number and title
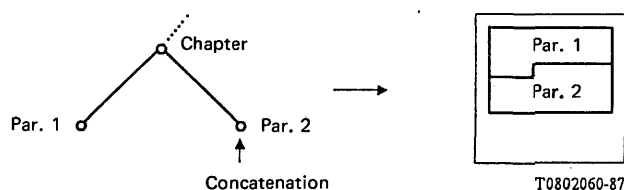
FIGURE D-9/T.412

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° 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.
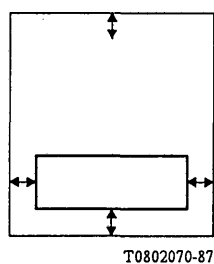
FIGURE D-10/T.412

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.
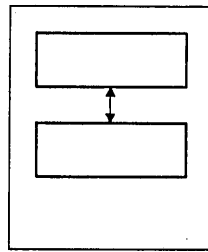
T0802080-87

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.

T0802090-87

Layout path 270   Layout path 0
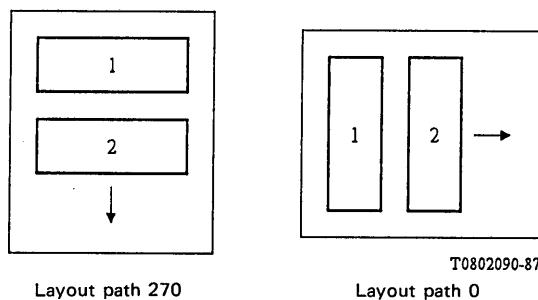
FIGURE D-12/T.412

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°), 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.
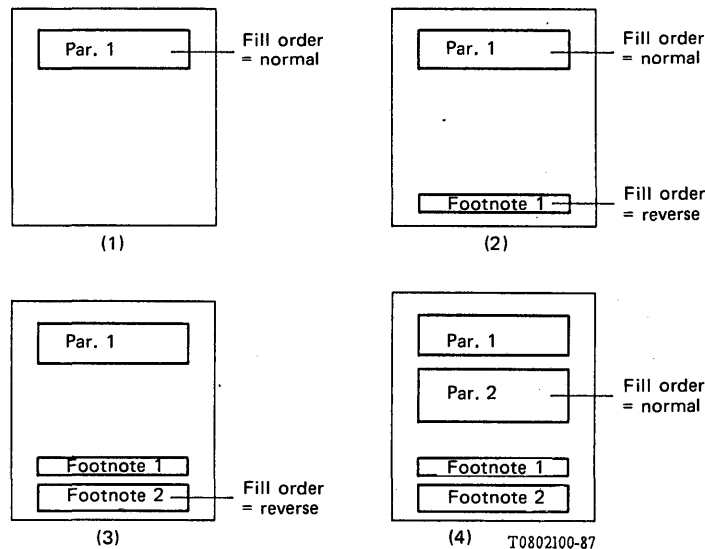
FIGURE D-13/T.412

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°):

- '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".
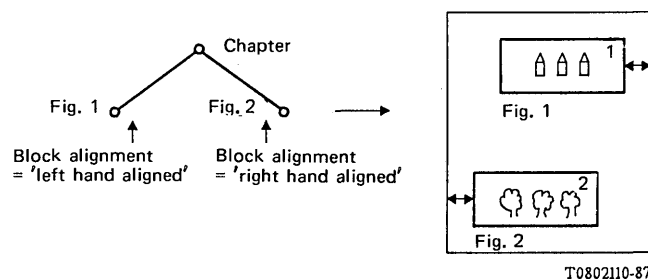
FIGURE D-14/T.412

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'.
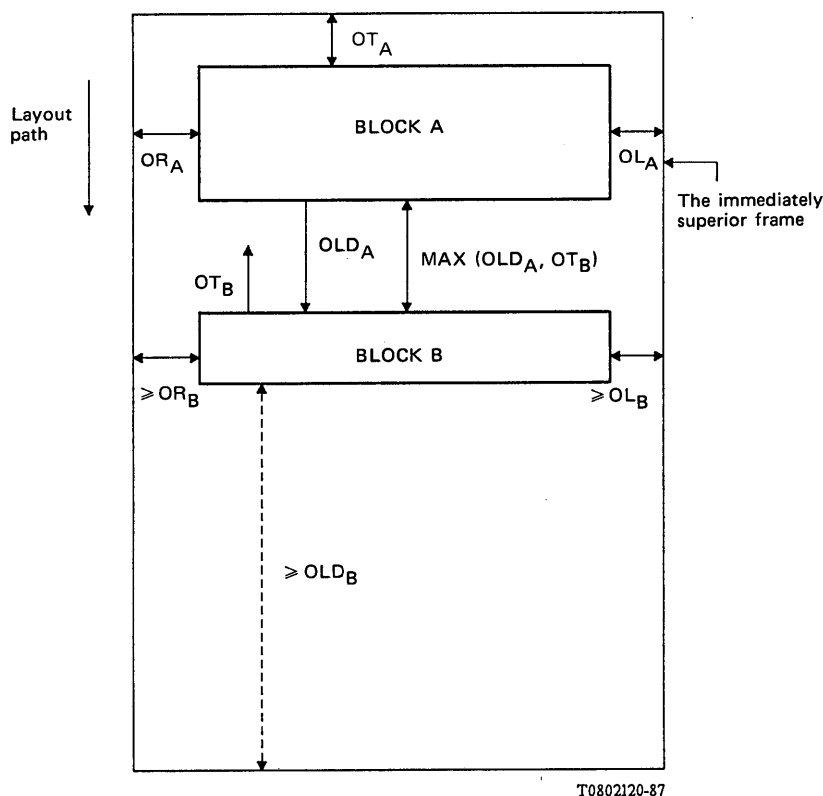
FIGURE D-15/T.412

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.
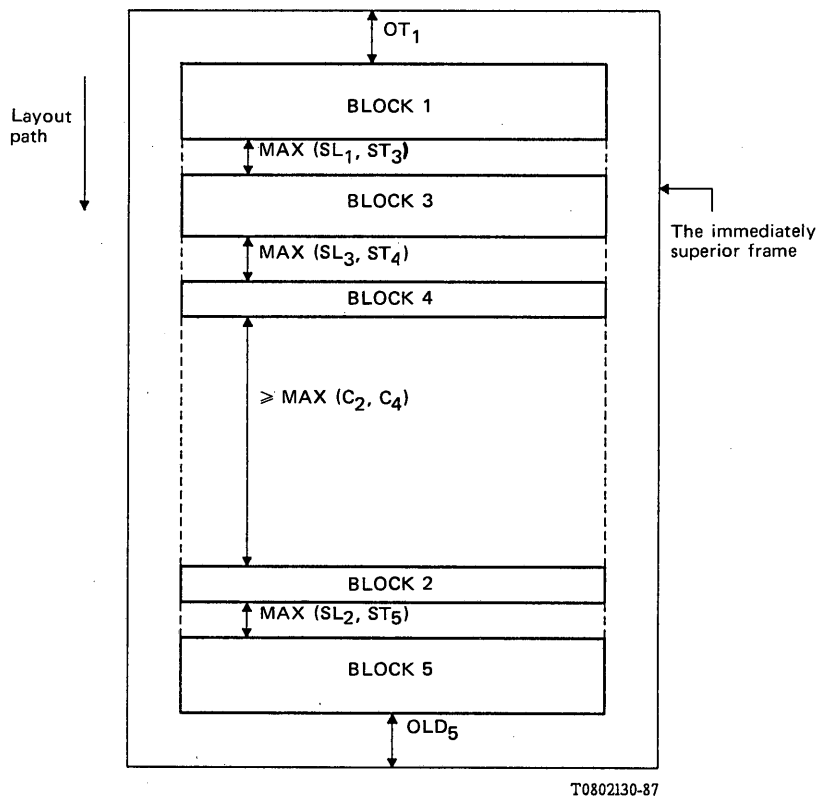
FIGURE D-16/T.412

Illustration of layout of blocks in normal and reverse order

Figure D-17/T.412 illustrates the combined use of the attributes "offset" and "separa-tion" 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.
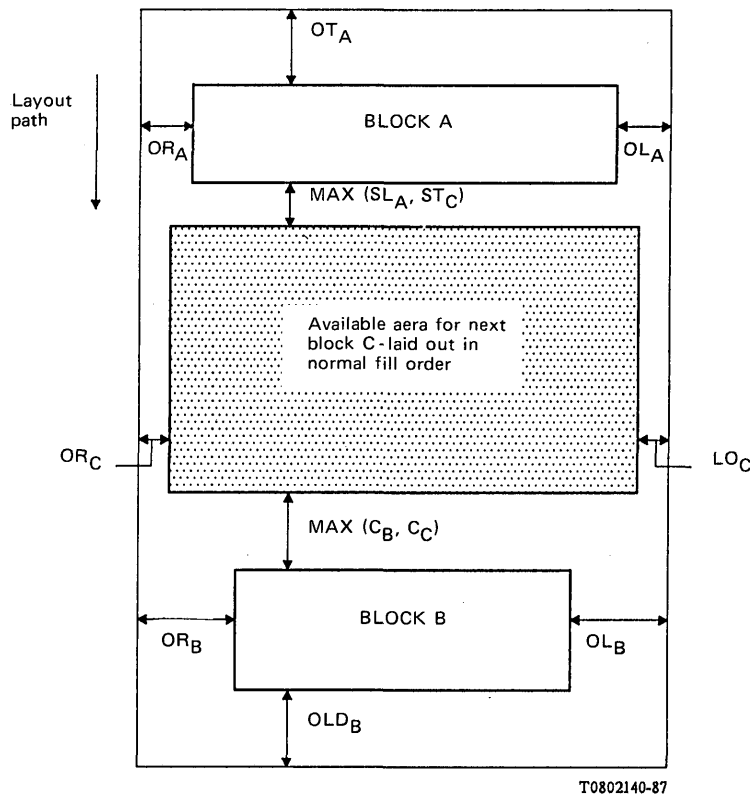
FIGURE D-17/T.412

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.

T0802150-87
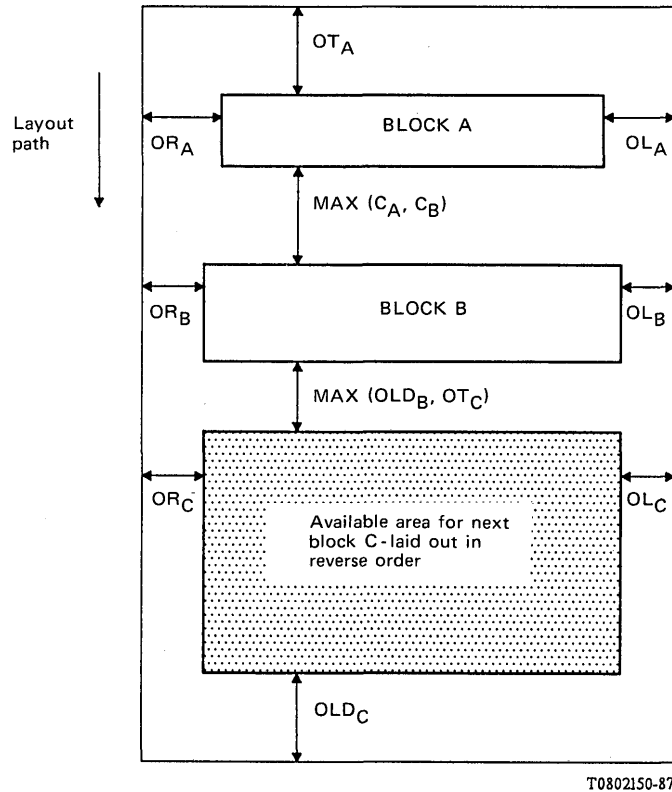
FIGURE D-18/T.412

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.

T0802180-87
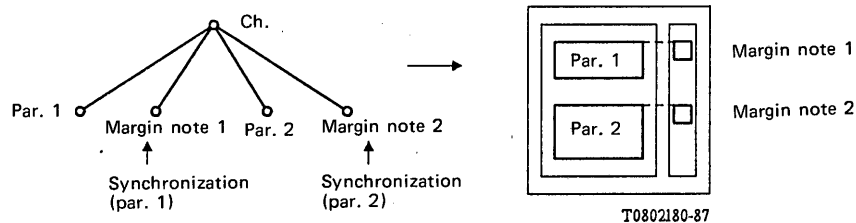
FIGURE D-19/T.412

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.

T0802190-87

Page 1   Page 2
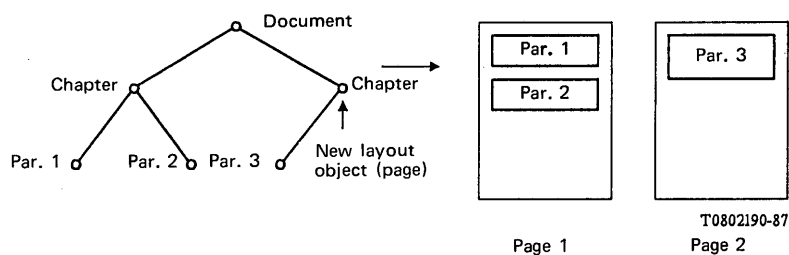
FIGURE D-20/T.412

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.

T0802200-87

Page 1   Page 2
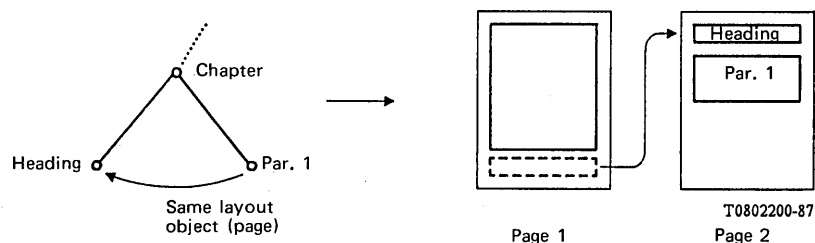
FIGURE D-21/T.412

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.
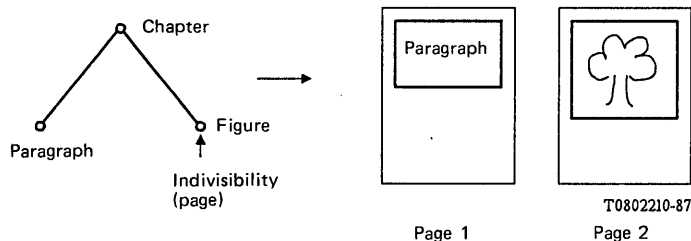
FIGURE D-22/T.412

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.

T0802220-87
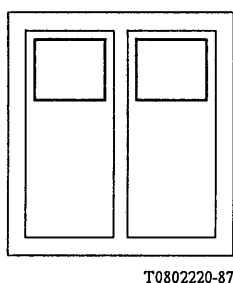
FIGURE D-23/T.412

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'.
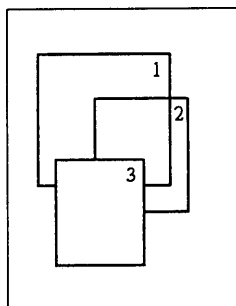
T0802230-87

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'.
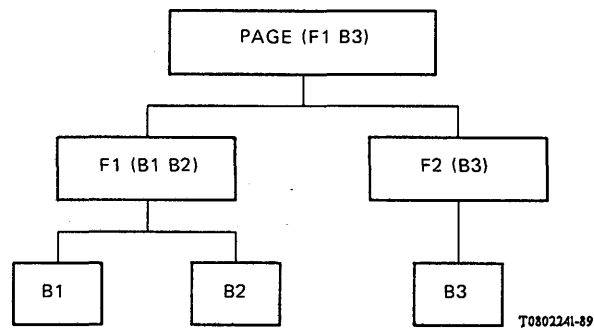
FIGURE D-25/T.412

Sequential layout order
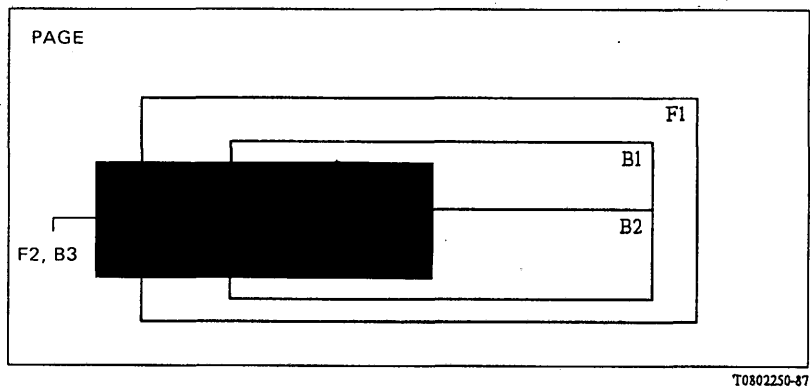
FIGURE D-26/T.412

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.
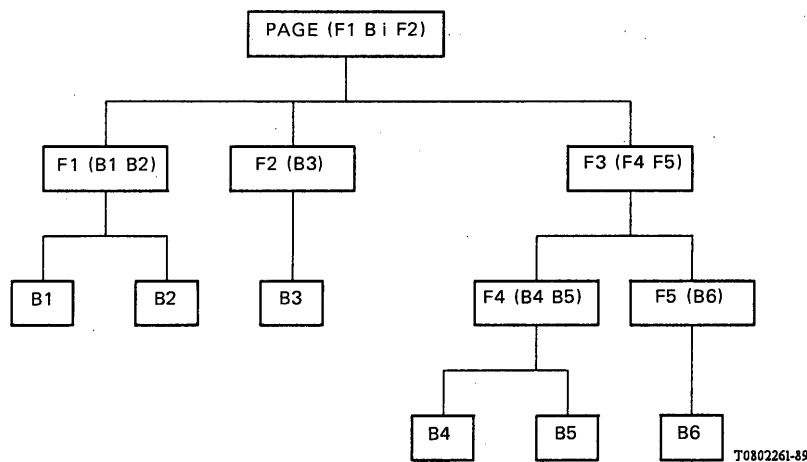
FIGURE D-27/T.412

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.

T0802270-87
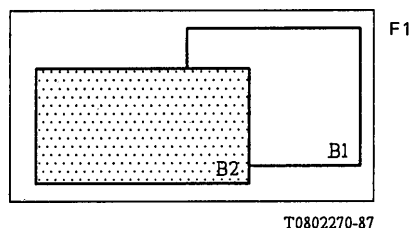
FIGURE D-28/T.412

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).

T0802280-87
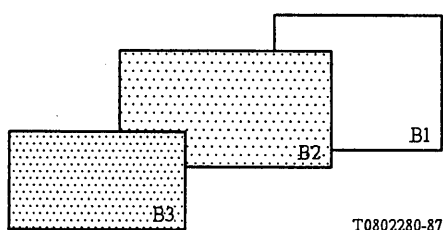
FIGURE D-29/T.412

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.

T0802290-87
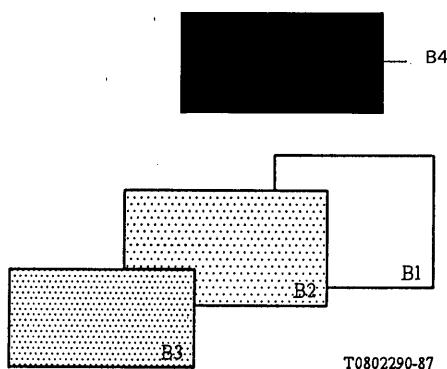
FIGURE D-30/T.412

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).
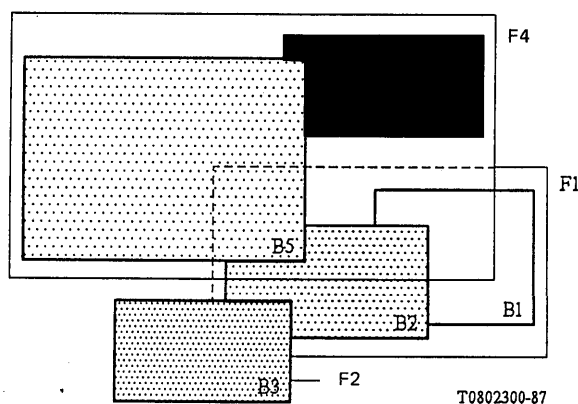
FIGURE D-31/T.412

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.
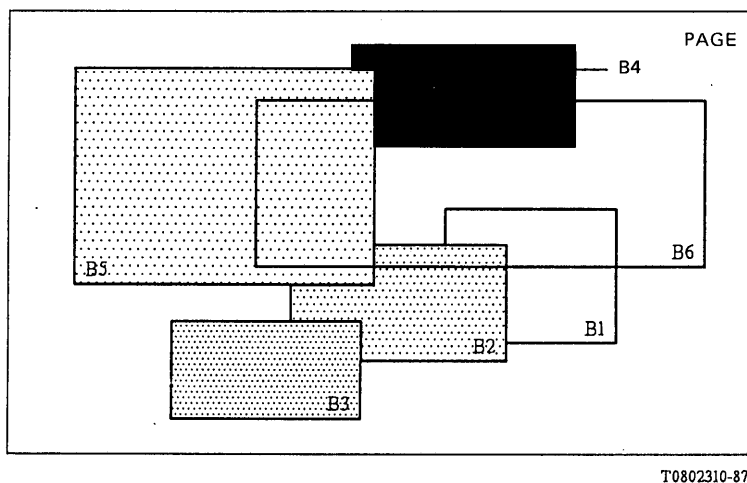
FIGURE D-32/T.412

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.
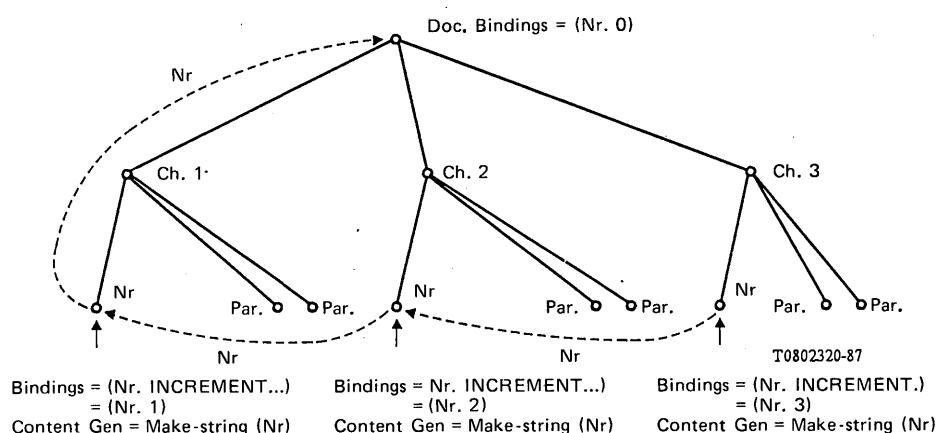
FIGURE D-33/T.412

Use of bindings and content generator for numbering chapters

(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 lists 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

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

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

OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT –

DOCUMENT PROFILE[1)]

CONTENTS

1   Scope

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.

1.2   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.

[1)]   This text is aligned with the final text of the corresponding International Standard ISO 8613-4.

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.

1.3   This Recommendation

-    defines the purpose of the document profile;

-    specifies the attributes which constitute the document profile.

Annex A specifies the format of personal names used in various document profile attributes.

Annex B specifies a minimum set of document profile attributes that is to be supported by document application profiles.

Annex C contains an example of a document profile.

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.

-    Recommendation T.50 (1984): International alphabet No. 5.

-    Recommendation X.208 (1988): Specification of abstract syntax notation one (ASN.1).

-    ISO 2022 (1986): Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques.

-    ISO 6937-2 (1983): Information processing - Coded character sets for text communication - Part 2: Latin alphabetic and non-alphabetic graphic characters.

-    ISO 8601: Data elements and interchange formats - Information interchange - Representation of dates and times.[2]

-    ISO 9541-5: Information processing - Font and character information interchange - Part 5: Font attributes and character model.[2]

-    ISO 9541-6: Information processing - Font and character information interchange - Part 6: Font and character attribute subsets and applications.[2]

3   Definitions

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

4   Purpose of the document profile

The document profile provides information by means of attributes which pertain to the document as a whole. It includes information for processing the document (e.g. reformatting, editing, filing/retrieval).

A document profile may be interchanged or stored without the body of the document. In such a case, the attribute "local file references" may be used to indicate the location of the document.

The document profile contains information for use by human beings and for machine processing.

---

[2]   To be published.

5 Content of the document profile

This section is a list of attributes that may occur within the document profile.

5.1 *Values of document profile attributes*

The value of each attribute is either user-specified or specified in this Recommendation.

Where attribute values consist of character strings, the document profile character set is used. This set consists of SPACE, CARRIAGE RETURN, LINE FEED and a set of graphic characters. In the absence of the attribute "profile character sets," this set of graphic characters consists of the 73 graphic characters of the minimum sub-repertoire of ISO 6937-2.

5.2 *Presence of document constituents*

The attributes defined in this section indicate the presence of document constituents associated with the document body, whether or not interchanged with the document profile.

5.2.1 *Generic layout structure*

This attribute is used if and only if the document contains any layout object class descriptions.

The value of this attribute (if specified) is one of 'factor set', 'partial generator set' or 'complete generator set'.

5.2.2 *Specific layout structure*

This attribute is used if and only if the document contains any layout object descriptions.

The value of this attribute (if specified) is 'present'.

5.2.3 *Generic logical structure*

This attribute is used if and only if the document contains any logical object class descriptions.

The value of this attribute (if specified) is one of 'factor set', 'partial generator set' or 'complete generator set'.

5.2.4 *Specific logical structure*

This attribute is used if and only if the document contains any logical object descriptions.

The value of this attribute (if specified) is 'present'.

5.2.5 *Layout styles*

This attribute is used if and only if the document contains any layout styles.

The value of this attribute (if specified) is 'present'.

5.2.6 *Presentation styles*

This attribute is used if and only if the document contains any presentation styles.

The value of this attribute (if specified) is 'present'.

5.2.7 *External-document class*

This attribute is used if and only if the document refers to one or more of the generic layout structure, generic logical structure, layout styles and presentation styles defined in an external-document class description. This attribute has no effect if any generic layout or generic logical structure is present in the document.

The value of this attribute is either an ASN.1 object identifier or a string of characters from the document profile character set.

This value is equal to the value of the document profile attribute "document reference" of the document referred to.

Constraints on the permitted external-document classes may be specified by the document application profile.

5.2.8 *Resource-document*

This attribute is used if and only if the document refers to a resource-document.

The value of this attribute is either an ASN.1 object identifier or a string of characters from the document profile character set.

This value is equal to the value of the document profile attribute "document reference" of the document referred to.

5.2.9 *Resources*

This attribute provides a mapping between names and identifiers of object class descriptions within the body of the document in order that the document may be used as a resource-document.

The value of this attribute consists of one or more pairs, each pair comprising a character string, representing the name, and an object class identifier.

The characters used in resource names are limited to those of the minimum sub-repertoire of ISO 6937-2.

5.3 *Document characteristics*

5.3.1 *Document application profile*

This attribute specifies the document application profile that pertains to the document.

The value of this attribute is either an ASN.1 object identifier or an integer.

The integer value is used only to identify the document application profile for Group 4 Facsimile, Class 1, defined in Recommendation T.503. In this case, the value of the integer is 2.

The absence of this attribute indicates that all constituents, attributes and attribute values defined in Recommendations T.411 to T.418 are permitted to be used, that all features are basic, and that the default values are those defined in the T.410 Series of Recommendations.

5.3.2 *Document application profile defaults*

This attribute specifies the default attribute values, specified in the document application profile, which are different from the values specified in the T.410 Series of Recommendations.

The document architecture attributes for which a document application profile may define non-standard values are:

- content architecture class;
- content type;
- dimensions, only in the case that this attribute applies to a component of object type page;
- transparency;
- colour;
- border;
- layout path;
- page position;
- medium type;
- block alignment;
- type of coding.

Specifications of the presentation attributes and coding attributes for which a document application profile may define non-standard default values are given in Recommendations T.416, T.417 and T.418.

The value of this attribute is a list of default values for any of the defaultable attributes (document architecture attributes as well as content architecture attributes) for which the document application profile defines a default value different from the default value defined in the T.410 Series of Recommendations.

5.3.3 *Document architecture class*

This attribute specifies the document architecture class used in the document.

The value of this attribute is 'formatted', 'processable' or 'formatted processable', representing one of the document architecture classes defined in Recommendation T.412.

5.3.4 *Content architecture classes*

This attribute specifies the content architecture classes used in the document.

The value of this attribute consists of one or more ASN.1 object identifiers referring to content architecture classes defined in the T.410 Series of Recommendations or in other standards.

Constraints on the permitted content architecture classes may be specified by the document application profile. The value of each object identifier must designated a content architecture class defined in accordance with the rules specified in Recommendation T.412.

Note – If no document application profile is specified, this attribute can only take values of identifiers of content architecture classes specified in Recommendations T.416, T.417 and T.418 (see § 5.3.1).

5.3.5 *Interchange format class*

This attribute specifies the interchange format class used to represent the document.

The value of this attribute is 'A' or 'B', representing one of the interchange format classes defined in Recommendation T.415.

5.3.6 *ODA version*

This attribute identifies the Document Architecture Standard or Recommendation, and its version, to which the document conforms.

The value of this attribute consists of two parameters. The value of the first parameter is a string of characters from the document profile character set which identifies the Document Architecture Standard or Recommendation. The value of the second parameter is a complete representation of a calendar date (as defined in ISO 8601), which indicates that the document conforms to the version and addenda current on this date.

5.3.7 *Non-basic document characteristics*

5.3.7.1 *Profile character sets*

This attribute specifies the graphic character set(s), other than the character set specified in § 5.1, used in those document profile attributes that consist of character strings.

The value of this attribute consists of the escape sequence(s) used to announce and to designate the set(s) in accordance with ISO 2022 and the register of ISO 2375.

5.3.7.2 *Comments character sets*

This attribute specifies the graphic character set(s), other than the default character set specified in Recommendation T.412, used in the document architecture attributes "user-readable comments" and "user-visible name".

The value of this attribute consists of the escape sequence(s) used to announce and to designate the set(s) in accordance with ISO 2022 and the register to ISO 2375.

5.3.7.3 *Alternative representation character sets*

This attribute specifies the graphic character set(s), other than the default character set specified in Recommendation T.412, used in the document architecture attributes "user-readable comments" and "user-visible name".

The value of this attribute consists of the escape sequence(s) used to announce and to designate the set(s) in accordance with ISO 2022 and the register to ISO 2375.

5.3.7.4 *Document constituent attributes*

5.3.7.4.1 *Page dimensions*

This attribute specifies the non-basic values of the attribute "dimensions" of layout objects of type 'page' used in the document.

The value of this attribute consists of one or more pairs of page dimensions. A pair of page dimensions shall be included in this attribute when the horizontal component and/or the vertical component of the page dimensions exceed the corresponding components of the basic page dimensions specified by the document application profile.

Each pair of page dimensions is represented by a pair of positive integers specifying the horizontal and vertical components of the page dimensions in scaled measurement units.

5.3.7.4.2 *Medium types*

This attribute specifies the non-basic values of the attribute "medium type" used in the document.

The value of this attribute consists of one or more groups of parameter values. Each group consists of the parameter "nominal page size" and/or the parameter "side of sheet", and details one non-basic medium type used in the document. The meaning and format of each parameter is defined in Recommendation T.412.

5.3.7.4.3 *Layout paths*

This attribute specifies the non-basic values of the attribute "layout path" used in the document.

The value of this attribute consists of one or more values of the attribute "layout path", as defined in Recommendation T.412.

5.3.7.4.4 *Protections*

This attribute specifies the non-basic values of the attribute "protection" used in the document.

The value of this attribute consists of one or more values of the attribute "protection", as defined in Recommendation T.412.

5.3.7.4.5 *Block alignments*

This attribute specifies the non-basic values of the attribute "block alignment" used in the document.

The value of this attribute consists of one or more values of the attribute "block alignment", as defined in Recommendation T.412.

5.3.7.4.6 *Fill orders*

This attribute specifies the non-basic values of the attribute "fill order" used in the document.

The value of this attribute consists of one or more values of the attribute "fill order", as defined in Recommendation T.412.

5.3.7.4.7 *Transparencies*

This attribute specifies the non-basic values of the attribute "transparency" used in the document.

The value of this attribute consists of one or more values of the attribute "transparency", as defined in Recommendation T.412.

5.3.7.4.8 *Colours*

This attribute specifies the non-basic values of the attribute "colour" used in the document.

The value of this attribute consists of one or more values of the attribute "colour", as defined in Recommendation T.412.

5.3.7.4.9 *Borders*

This attribute specifies the non-basic values of the attribute "border" used in the document.

The value of this attribute consists of one or more values of the attribute "border", as defined in Recommendation T.412.

5.3.7.4.10 *Page positions*

This attribute specifies the non-basic values of the attribute "page position" used in the document.

The value of this attribute consists of one or more values of the attribute "page position", as defined in Recommendation T.412.

5.3.7.4.11 *Types of coding*

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

The value of this attribute consists of one or more values of the attribute "type of coding", as defined in Recommendation T.412.

5.3.7.5 *Coding attributes*

This attribute consists of one or more sets of coding attribute values used in the document. Each set pertains to a single content type and consists of coding attribute values that are specified as non-basic by the document application profile.

The names of the sets of coding attribute values are:

- character coding attributes;

- raster-graphics coding attributes;

- geometric-graphics coding attributes.

5.3.7.6 *Presentation features*

This attribute consists of one or more sets of presentation features used in the document. Each set pertains to a single content type and consists of presentation features that are specified as non-basic by the document application profile.

Presentation features consist of presentation attribute values, control function parameter values, sets of content elements and their parameter values.

The names of the sets of presentation features are:

- character presentation;

- raster-graphics presentation features;

- geometric-graphics presentation features.

5.3.8 *Non-basic structure characteristics*

5.3.8.1 *Number of objects per page*

This attribute specifies the maximum number of specific layout objects per page used in the document. This attribute is only specified if the maximum number of objects per page exceeds the value specified by the document application profile.

5.3.9 *Additional document characteristics*

5.3.9.1 *Unit scaling*

This attribute specifies a scaling factor that is to be applied to all attributes and numeric control function parameters that specify absolute or relative positions and dimensions.

The value of this attribute is a pair of integers m and n which indicate that these positions and dimensions are to be interpreted as being expressed in units equal to m/n BMU. The name of this unit is scaled measurement unit (SMU).

In the absence of this attribute, the scaled measurement unit is equal to 1 BMU.

5.3.9.2 *Fonts list*

This attribute specifies the character font(s) used in the document.

The value of this attribute consists of one or more pairs, each pair comprising an integer, representing the font identifier, and a font reference. A font reference consists of a subset of the font attributes defined in ISO 9541-5. Appropriate subsets are defined in ISO 9541-6.

A font is referred to from within the document by means of the associated integer font identifier.

Note – The font attributes defined in ISO 9541-5 include both a structured font name and a set of character attributes.

5.4 *Document management attributes*

5.4.1 *Document description*

5.4.1.1 *Title*

This attribute gives the name of the document as specified by the author.

The value of this attribute consists of a string of characters from the document profile character set.

5.4.1.2 *Subject*

This attribute contains information to indicate the subject of the document.

The value of this attribute consists of a string of characters from the document profile character set.

5.4.1.3 *Document reference*

This attribute identifies the document. The attribute's value is used to refer to the document from other documents (see §§ 5.2.7, 5.2.8, 5.4.5).

The value of this attribute is either an ASN.1 object identifier or a string of characters from the document profile character set.

5.4.1.4 *Document type*

This attribute specifies the type of document, e.g. memorandum, letter, report, resource. This attribute specifies only an informal name; it does not specify a relation to a particular document class description.

The value of this attribute consists of a string of characters from the document profile character set.

5.4.1.5 *Abstract*

This attribute contains information to summarize the document.

The value of this attribute consists of a string of characters from the document profile character set.

5.4.1.6 *Keywords*

This attribute specifies one or more character strings that permit logical associations to be made about the content of the document.

The value of this attribute consists of string(s) of characters from the document profile character set.

5.4.2 *Dates and times*

5.4.2.1 *Document date and time*

This attribute specifies the date and, optionally, the time of day that the originator associates with the document.

The value of this attribute consists of a date character string and, optionally, a time of day character string, in accordance with ISO 8601.

5.4.2.2 *Creation date and time*

This attribute specifies the date and, optionally, the time of day when the document was created.

The value of this attribute consists of a sequence of parameters. Each parameter consists of a date character string and, optionally, a time of day character string, in accordance with ISO 8601.

5.4.2.3 *Local filing date and time*

This attribute specifies the date and, optionally, the time of day when the document was filed. When more than one entry occurs, the last entry indicates the most recent local filing date and time.

The value of this attribute consists of a date character string and, optionally a time of day character string, in accordance with ISO 8601.

5.4.2.4 *Expiry date and time*

This attribute specifies the date and, optionally, the time of day after which the document is considered to be invalid.

The value of this attribute consists of a date character string and, optionally, a time of day character string, in accordance with ISO 8601.

5.4.2.5 *Start date and time*

This attribute specifies the date and, optionally, the time of day after which the document is considered to be valid.

The value of this attribute consists of a date character string and, optionally, a time of day character string, in accordance with ISO 8601.

5.4.2.6 *Purge date and time*

This attribute specifies the date and, optionally, the time of day after which the document can be purged from wherever it is stored.

The value of this attribute consists of a date character string and, optionally, a time of day character string, in accordance with ISO 8601.

5.4.2.7 *Release date and time*

This attribute specifies the date and, optionally, the time of day after which the document can be released from any restrictions specified in the attribute "security classification".

The value of this attribute consists of a date character string and, optionally a time of day character string, in accordance with ISO 8601.

5.4.2.8 *Revision history*

This attribute specifies the history of the document, indicating when, where and by whom the document was created or revised.

The value of this attribute consists of a sequence of groups of parameters. Each group forms an entry in the history. The first group in the sequence provides information on the creation of the document. The last group in the sequence provides information on the current version of the document. Each group consists of the following optional parameters:

a)   revision date and time;

b)   version number;

c)   revisor(s);

d)   version reference;

e)   user comments;

where

- "revision date and time" indicates the date and, optionally, the time of day on which a revision occurred; the format is in accordance with ISO 8601;

- "version number" indicates the version number of the document resulting from the revision; the format is a string of characters from the document profile character set;

- "revisor(s)" identifies the person(s) who carried out a revision; the identification may include the specification of the revision; the identification may include the specification of the name of one or more individuals, information to identify their position within an organization, and the name of the organization; the format consists of the following three optional sub-parameters:

  1) name(s);

  2) position;

  3) organization;

  where

  - "name(s)" has the format of one or more personal names, according to the format specified in Annex A;

  - "position" is a string of characters from the document profile character set;

  - "organization" is a string of characters from the document profile character set;

  - "version reference" is either an ASN.1 object identifier or a string of characters from the document profile character set; the value of this parameter is equal to the value of the document profile attribute "document reference" of the document referred to;

  - "user comments" describes the revisions made; the value of this parameter is a string of characters from the document profile character set.

5.4.3 Originators

5.4.3.1 Organizations

This attribute identifies the originating organization(s) associated with the document.

The value of this attribute consists of string(s) of characters from the document profile character set.

5.4.3.2 Preparers

This attribute identifies the name(s) of the person(s) and/or organization(s) responsible for the preparation of the document.

The value of this attribute consists of one or more entries. Each entry has two optional parameters:

- "personal name of preparer", according to the format specified in Annex A;

- "preparer's organization", a string of characters from the document profile character set, which indicates information about the organization responsible for preparing the document, e.g. name, address, telephone number.

5.4.3.3 Owners

This attribute identifies the current administrator(s) of the document.

The value of this attribute consists of one or more entries. Each entry has two optional parameters:

- "personal name of preparer", according to the format specified in Annex A;

- "owner's organization", a string of characters from the document profile character set, which indicates information about the organization which owns the document, e.g. name, address, telephone number.

5.4.3.4 *Authors*

This attribute identifies the author(s) of the document.

The value of this attribute consists of one or more entries. Each entry has two optional parameters:

- "personal name of author", according to the format specified in Annex A;

- "author's organization", a string of characters from the document profile character set, which indicates information about the organization responsible for the authorship of the document, e.g. name, address, telephone number.

5.4.4 *Other user information*

5.4.4.1 *Copyright*

The value of this attribute consists of one or more entries. Each entry has two optional parameters:

- "copyright information", identifying the name(s) of the legal party (parties) in whom the copyright of the document is vested; the value of this parameter consists of string(s) of characters from the document profile character set;

- "copyright dates", specifying the date(s) associated with the copyright by the holder(s) identified by the parameter "copyright information"; the value of this parameter consists of string(s) of characters representing date(s) in accordance with ISO 8601.

5.4.4.2 *Status*

This attribute specifies the document status, e.g. working paper, draft proposal.

The value of this attribute consists of a string of characters from the document profile character set.

5.4.4.3 *User-specific codes*

This attribute specifies additional user-specific code(s), e.g. contract number, project number, budget code.

The value of this attribute consists of a string(s) of characters from the document profile character set.

5.4.4.4 *Distribution list*

This attribute specifies a list of the intended recipients of the document.

The value of this attribute consists of one or more entries. Each entry has two optional parameters:

- "personal name of recipient", according to the format specified in Annex A;

- "recipient's organization", a string of characters from the document profile character set, which indicates information about the organization with which the recipient is associated, e.g. name, address, telephone number.

5.4.4.5 *Additional information*

This attribute may be used for information that cannot be specified by any other attribute of the document profile.

This attribute can have any value.

5.4.5 *External references*

5.4.5.1 *References to other documents*

This attribute specifies reference(s) to any other associated document(s). It consists of one or more entries.

The value of each entry is either an ASN.1 object identifier or a string of characters from the document profile character set.

This value is equal to the value of the document profile attribute "document reference" of the document referred to.

5.4.5.2 *Superseded documents*

This attribute specifies reference(s) to document(s) superseded by the current document. It consists of one or more entries.

The value of each entry is either an ASN.1 object identifier or a string of characters from the document profile character set.

This value is equal to the value of the document profile attribute "document reference" of the document referred to.

5.4.6 *Local file references*

This attribute specifies where a copy (copies) of the document may be found. It consists of one or more entries, one for each location where a copy (copies) of the document may be found.

Each entry consists of up to three optional parameters, namely

-   the file name;

-   the location of the document;

-   user comments.

The file name is a string of characters that can be used to identify the document uniquely in a filing system. The second parameter specifies the location of the document in a filing system, for example the name of the filing system, the name of the directory and folder in which the document is contained. The third parameter is used to provide user-readable comments.

The value of each parameter is a string of characters from the document profile character set.

5.4.7 *Content attributes*

5.4.7.1 *Document size*

This attribute represents the estimated size of the whole document, expressed as a number of 8-bit bytes, where the estimate must not be less than the actual size. The size includes that of the document profile and the document body (if present).

The value of this attribute is an integer, which is not less than the number of 8-bit bytes in the document data stream.

5.4.7.2 *Number of pages*

This attribute specifies the number of pages in the specific layout structure (if any) of the document.

The value of this attribute is an integer.

5.4.7.3 *Languages*

This attribute specifies the primary language(s) in which the content of the document is written.

The value of this attribute consists of string(s) of characters from the document profile character set.

5.4.8 *Security information*

These attributes provide security information only and are not intended to ensure security measures.

5.4.8.1 *Authorization*

This attribute identifies the person or organization approving or authorizing the document.

The value of this attribute consists either of a personal name with the format specified in Annex A, or the name of an organization consisting of a string of characters from the document profile character set.

5.4.8.2 *Security classification*

This attribute specifies the security classification assigned by the document owner(s) relating to such aspects as its visibility, reproduction, storage, audit and destruction requirements.

The value of this attribute consists of a string of characters from the document profile character set.

5.4.8.3 *Access rights*

This attribute specifies the access right(s) to the document relating to its privacy, as defined by the current owner(s) of the document.

The value of this attribute consists of string(s) of characters from the document profile character set.

ANNEX A

(to Recommendation T.414)

Format of personal names

ˊ(Normative)

The format of a personal name consists of a group of up to four parameters:

a)   surname;

b)   given name;

c)   initials;

d)   title.

The "surname" parameter is the family name of a person. This parameter is mandatory.

The "givenname" parameter is the name by which a person is commonly known. This parameter is optional.

The "initials" parameter consists of a sequence of the initial characters of any names other than "surname" and "givenname", in the order in which they are normally written. This parameter is optional.

The "title" parameter is the title by which a person is normally addressed in documents. This parameter is optional.

The format of each of the parameters is a string of characters from the document profile character set.

Note - The format defined above is the same as that given in Recommendation X.411 (1984).

ANNEX B

(to Recommendation T.414)

Minimum set of document profile attributes to be supported by document application profiles

(Normative)

The document profile attributes listed in Table B-1/T.414 shall be included in any document profile level defined as a part of a document application profile, subject to the conditions specified in the explanatory notes referred to in the second column of the Table.

The requirements for the use of these attributes in an actual instance of interchange of a document profile are indicated by the classifications "M" (mandatory) and "NM" (non-mandatory) in the third column of the Table.

TABLE B-1/T.414

Minimum set of document profile attributes

Attributes	Condition	Classification
Presence of document constituents		
- Generic layout structure	(1)	NM
- Specific layout structure	(1)	NM
- Generic logical structure	(1)	NM
- Specific logical structure	(1)	NM
- Layout styles	(1)	NM
- Presentation styles	(1)	NM
- External document class	(2)	NM
- Resource document	(3)	NM
- Resources	(4)	NM
Document characteristics		
- Document application profile		NM
- Document application profile defaults	(5)	NM
- Document architecture class		M
- Content architecture classes		M
- Interchange format class		M
- ODA version		M
Non-basic document characteristics		
- Page dimensions	(6)	NM
- Medium types	(6)	NM
- Layout paths	(6)	NM
- Transparencies	(6)	NM
- Colours	(6)	NM
- Protections	(6)	NM
- Block alignments	(6)	NM
- Fill orders	(6)	NM
- Borders	(6)	NM
- Page positions	(6)	NM
- Types of coding	(6)	NM
- Coding attributes	(6)	NM
- Presentation features	(6)	NM
Non-basic structure characteristics		
- Number of objects per page	(7)	NM
Additional document characteristics		
- Unit scaling	(8)	NM
- Fonts list	(9)	NM
Document management attributes		
- Document reference		M

(1) to be supported only if the document architecture level supports the corresponding constituents;

(2) to be supported only if the document application profile permits references to external document classes;

(3) to be supported only if the document application profile permits references to resource documents;

(4) to be supported only if the document application profile permits documents to act as resource documents;

(5) to be supported only if the document application profile defines any non-standard default values;

(6) to be supported only if the document application profile defines any of the corresponding non-basic features;

(7) to be supported only if the document application profile distinguishes between basic and non-basic numbers of objects per page;

(8) to be supported only if the document application profile supports the corresponding feature;

(9) to be supported only if the document application profile permits character fonts to be designated.

ANNEX C

(to Recommendation T.414)

Example

(Information)

Note – The following example of a document profile is provided in the form of a table for easy reference back to the sections that describe the attributes. Not all document profile attributes are used in the example.

TABLE C-1/T.414

Example of a document profile

Section	Attribute	Value
5.2	Presence of document constituents	
5.2.1	Generic layout structure	'complete generator set'
5.2.2	Specific layout structure	'present'
5.2.4	Specific logical structure	'present'
5.2.8	Resource document	Finance Master
		Widget Inc.
		4511 McKenzie,
		Atlanta, Georgia, USA

TABLE C-1/T.414 *(cont.)*

Section	Attribute	Value
5.3	Document characteristics	
5.3.1	Document application profile	[TOP 5.5][a]
5.3.2	Document application profile defaults	Dimensions 10200, 13200
		Transparency 'opaque'
5.3.3	Document architecture class	'formatted processable'
5.3.4	Content architecture classes	[f-p characters][a]
5.3.5	Interchange format class	'A'
5.3.6	ODA version	ISO 8613, 1987-12-15
5.3.7	Non-basic document characteristics	
5.3.7.4	Document constituent attributes	
5.3.7.4.1	Page dimensions	13200, 10200
5.3.7.4.2	Medium types	10200, 13200, 'recto'
5.3.7.4.4	Protections	'protected'
5.3.9	Additional document characteristics	
5.3.9.1	Unit scaling	12, 10
5.3.9.2	Fonts list	0, [FONT 0];
		1, [FONT 1][b]
5.4	Document management attributes	
5.4.1	Document description	
5.4.1.1	Title	May finance report
5.4.1.2	Subject	May results
5.4.1.3	Document reference	May financial preliminary
5.4.1.4	Document type	Report
5.4.1.5	Abstract	The current figures show an improvement in return on assets but still show an undercapitalization of production capacity.
5.4.1.6	Keywords	Finance, Financial, May, Return on assets
5.4.2	Dates and times	
5.4.2.1	Document date and time	1988-06-05
5.4.2.2	Creation date and time	1988-05-23, 16:29:57
5.4.2.3	Local filing date and time	1988-06-05, 11:51:03
5.4.2.4	Expiry date and time	1989
5.4.2.6	Purge date and time	1989-12-31
5.4.2.7	Release date and time	1989-06-05
5.4.3	Originators	
5.4.3.1	Organizations	Widget Inc., Finance and Control
5.4.3.2	Preparers	Maltby, Reginald P., Accountant
5.4.3.3	Owners	Widget Inc., 4511 McKenzie,
5.4.3.4	Authors	Atlanta, Georgia, USA Dewey, Cheatam & Howe CPA

[a] The actual value of this attribute would be an ASN.1 object identifier.
[b] The actual value of this attribute would be structured according to ISO 9541-6.

TABLE C-1/T.414 *(cont.)*

Section	Attribute	Value
5.4.4	Other user information	
5.4.4.1	Copyright	Widget Inc., 1988
5.4.4.2	Status	May final report
5.4.4.4	Distribution list	D. Marks, Accountant
		B. Bucks, VP Finance
		James K. Pencil, Audits
		D. Duck, Controller
5.4.4.5	Additional information	Signature receipt required
5.4.5	External references	
5.4.5.1	References to other documents	April finance report,
		May balance,
		May accounting preliminary
5.4.5.2	Superseded documents	May financical A
5.4.6	Local file references	mayfin.
		financial-previous
		mayfin.
		financial-current
5.4.7	Content attributes	
5.4.7.1	Document size	40447
5.4.7.2	Number of pages	16
5.4.7.3	Languages	US English
5.4.8	Security information	
5.4.8.1	Authorization	Widget Inc., Finance
5.4.8.2	Security classification	Company Financial
5.4.8.3	Access rights	Finance Group

OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT –

OPEN DOCUMENT INTERCHANGE FORMAT (ODIF)[1)]

CONTENTS

1   *Scope*

2   *Normative references*

3   *Definition*

4   *Document representation*

5   *Open document interchange format (ODIF)*

1   Scope

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.

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

[1)]   This text is aligned with the final text of the corresponding International Standard ISO 8613-5.

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.

1.3	This Recommendation

-	defines the format of the data stream used to interchange documents structured in accordance with Recommendation T.412;

-	defines the representation of the constituents which may appear in an interchanged document.

Annex A is a summary of an encoding scheme that may be used for interchanging documents.

Annex B is a summary of the application class tag assignments made in this Recommendation.

Annex C is a summary of the ASN.1 object identifier value assignments made in this Recommendation.

Annex D contains examples of data streams.

Note 1 - This Recommendation does not specify the coded representation of content elements.

Note 2 - Data formats for presentation attributes and coding attributes are defined in other Recommendations of the T.410 Series.

## 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.

-	Rec. X.208 (1988): Specification of abstract syntax notation one (ASN.1).

-	Rec. X.209 (1988): Specification of basic encoding rules for abstract syntax notation one (ASN.1).

-	ISO 8601: Data elements and interchange formats - Information interchange - Representation of dates and times.[2]

-	ISO 9541-6: Information processing - Font and character information interchange - Part 6: Font and character attribute subsets and applications.[2]

---

[2]	To be published.

3 Definitions

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

4 Document representation

A document structured in accordance with the T.410 Series of Recommendations is represented for interchange by the open document interchange format (ODIF). ODIF is an abstract data syntax in which the constituents and attributes of the document are represented by a hierarchy of data structures and data items, specified using the abstract syntax notation ASN.1 defined in Recommendation X.209.

The coded representation of each data structure or data item is obtained by applying a set of encoding rules.

ODIF is specified in § 5.

Note - ASN.1 is a formal description method that allows data types relevant to an application to be specified in terms of other data types, including basic types such as "integer" and "octet string" which are defined in Recommendation X.208 itself. Basic encoding rules for ASN.1 are defined in Recommendation X.209 and are summarized in Annex A.

5 Open document interchange format (ODIF)

5.1 *General description*

A document structured in accordance with the T.410 Series of Recommendations is represented by a data stream consisting of one or more data structures of the following types:

- document profile descriptor;
- layout object descriptor;
- layout object class descriptor;
- logical object descriptor;
- logical object class descriptor;
- presentation style descriptor;
- layout style descriptor;
- text unit.

These data structures are called *interchange data elements*.

Within a data stream, the interchange data elements are ordered in accordance with certain rules which are specified below. This Recommendation defines two such sets of rules; they are called *interchange format class A* and *interchange format class B*.

Which of these sets of rules applies to a given data stream is indicated in the document profile descriptor. In all cases, a data stream contains one and only one document profile descriptor which is always the first interchange data element in the data stream. The document profile descriptor may be the only data structure in the data stream.

When an ODIF data stream is used as part of an ASN.1 external data type, the abstract system shall be formed by an ASN.1 sequence-of-type, referring to the interface data element type and the encoding of the data value shall consist of an integral number of octets, formed by applying the ASN.1 basic encoding rules, and the value of the associated ASN.1 object identifier shall be { 2 8 0 0 }

Note - The manner of incorporating the interchange data elements, or the external data type, in an application protocol or the manner of mapping them on service data units (in an OSI environment) is not defined by this Recommendation.

5.2 *Interchange format class A*

According to interchange format class A, a data stream consists of one document profile descriptor and, optionally, one or more interchange data elements of the following types:

- layout object descriptor;
- layout object class descriptor;
- logical object descriptor;
- logical object class descriptor;
- presentation style descriptor;
- layout style descriptor;
- text unit.

The order of the interchange data elements is as follows:

a) document profile descriptor;
b) layout object class descriptor;
c) logical object class descriptor;
d) text units representing generic content portions;
e) presentation style descriptor;
f) layout style descriptor;
g) layout object descriptor;
h) logical object descriptor;
i) text units representing specific content portions.

Within each of the groups of layout object descriptors and logical object descriptors, the order of the descriptors is equal to the sequential order defined in Recommendation T.412.

If the data stream contains layout object descriptors, the text units representing specific content portions are ordered according to the sequential layout order; otherwise, they are ordered according to the sequential logical order.

Within each of the other groups of interchange data elements, the order is arbitrary.

5.3 *Interchange format class B*

According to interchange format class B, a data stream consists of one document profile descriptor and, optionally, one or more interchange data elements of the following types:

- layout object descriptor;
- layout object class descriptor;
- presentation style descriptor;
- text unit.

Interchange format class B can be used only to represent documents that do not contain any specific or generic logical structure, i.e. documents that conform to the formatted document architecture class.

The order of the interchange data elements is as follows:

a) document profile descriptor;
b) layout object class descriptors and associated text units;
c) presentation style descriptors;
d) layout object descriptors and associated text units.

Within the group of layout object class descriptors and associated text units, the order is such that a group of descriptors that have identical identifiers, except for the last number of each identifier, follow each other in the data stream without any other descriptor between them. However, each descriptor of an object class for a basic layout object is followed immediately by the associated text units.

Within the group of layout object descriptors and associated text units, the order of the descriptors is equal to the sequential order defined in Recommendation T.412. However, each descriptor of a basic layout object is followed immediately by the associated text units.

Within the group of presentation style descriptors, the order is arbitrary.

5.4 *Descriptors and text units*

A document profile descriptor, layout object descriptor, layout object class descriptor, logical object descriptor, logical object class descriptor, presentation style descriptor or layout style descriptor consists of simple and composite data items representing the attributes of the constituent concerned.

The document profile, each object class, each style and each object is represented by one descriptor.

A text unit consists of two parts:

a) an attribute field, i.e. a data structure consisting of simple and composite data items representing the attributes of the content portion concerned.

b) an information field, i.e. a data structure that is either a data item or a set of data items representing the content elements making up the content portion concerned.

Each content portion is represented by one text unit.

The data formats of the interchange data elements are specified in §§ 5.5 to 5.12, using the abstract syntax notation ASN.1 defined in Recommendation X.208.

Note - Paragraphs 5.5 to 5.12 by themselves do not completely define the data stream format; additional rules are specified in §§ 5.1 to 5.4 of this Recommendation and in other Recommendations of the T.410 Series. For example, the keyword OPTIONAL merely indicates that a particular data structure or data item is not part of every instance of the containing data structure; the conditions controlling the presence or absence of the data structure or data item are specified in Recommendations T.412 and T.414.

5.5 *Interchange data elements*

Interchange-Data-Elements { 2 8 1 5 5 }

DEFINITIONS   ::=   BEGIN

EXPORTS Interchange-Data-Element;

IMPORTS   Document-Profile-Descriptor
          FROM Document-Profile-Descriptor   -- *see § 5.6*
          Layout-Class-Descriptor, Layout-Object-Descriptor,
          FROM Layout-Descriptors   -- *see § 5.8*
          Logical-Class-Descriptor, Logical-Object-Descriptor
          FROM Logical-Descriptors   -- *see § 5.9*
          Presentation-Style-Descriptor, Layout-Style-Descriptor
          FROM Style-Descriptors   -- *see § 5.10*
          Text-Unit FROM Text-Unit;   -- *see § 5.12*

Interchange-Data-Element   .   ::=   CHOICE {

          document-profile   [0]   IMPLICIT Document-Profile-Descriptor,
          layout-object-class   [1]   IMPLICIT Layout-Class-Descriptor,
          layout-object   [2]   IMPLICIT Layout-Object-Descriptor,

content-portion	[3]	IMPLICIT Text-Unit,
logical-object-class	[5]	IMPLICIT Logical-Class-Descriptor,
logical-object	[6]	IMPLICIT Logical-Object-Descriptor,
presentation-style	[7]	IMPLICIT Presentation-Style-Descriptor,
layout-style	[8]	IMPLICIT Layout-Style-Descriptor}

END

5.6   *Document profile descriptor*

Document-Profile-Descriptor { 2 8 1 5 6 }

DEFINITIONS   ::=   BEGIN

EXPORTS   Document-Profile-Descriptor;

IMPORTS   Resource-Name, Object-or-Class-Identifier
           FROM Identifiers-and-Expressions   *-- see § 5.7*
           Measure-Pair, Transparency, Colour, Dimension-Pair, One-Of-Four-Angles,
           Border, Medium-Type, FROM Layout-Descriptors   *-- see § 5.8*
           Protection FROM Logical-Descriptors   *-- see § 5.9*
           Content-Architecture-Class, Content-Type,
           Block-Alignment, Fill-Order
           FROM Style-Descriptors   *-- see § 5.10*
           Type-of-Coding FROM Text-Units   *-- see § 5.12*
           Character-Content-Defaults, Char-Presentation-Feature,
           Character-Coding-Attribute
           FROM Character-Profile-Attribute { 2 8 1 6 4 }   *-- see Recommendation T.416*
           Raster-Gr-Content-Defaults, Ra-Gr-Presentation-Feature,
           Ra-Gr-Coding-Attribute
           FROM Raster-Gr-Profile-Attributes { 2 8 1 7 4 }   *-- see Recommendation T.417*
           Geo-Gr-Content-Defaults, Geo-Gr-Presentation-Feature,
           Geo-Gr-Coding-Attribute
           FROM Geo-Gr-Profile-Attributes { 2 8 1 8 4 }   *-- see Recommendation T.418*
           Font-Reference FROM ISO9541-FONTS { 1 9541 6 1 0 };   *-- see   ISO 9541-6*

Document-Profile-Descriptor	::=	SET {	
generic-layout-structure		[0]	IMPLICIT NumericString OPTIONAL,
specific-layout-structure		[1]	IMPLICIT NumericString OPTIONAL,
generic-logical-structure		[4]	IMPLICIT NumericString OPTIONAL,
specific-logical-structure		[5]	IMPLICIT NumericString OPTIONAL,
presentation-styles		[6]	IMPLICIT NumericString OPTIONAL,
layout-styles		[7]	IMPLICIT NumericString OPTIONAL,

-- for the generic structures:
-- 'partial-generator-set ' is represented by "0 ",
-- 'complete-generator set ' is represented by, "1",
-- 'factor-set ' is represented by "2",
-- for the other cases, the Numeric string has the value 'present '
-- represented by "1".

external-document-class	[9]	Document-Reference OPTIONAL,
resource-document	[10]	Document-Reference OPTIONAL,
resources	[11]	IMPLICIT SET OF SET {
resource-identifier		Resource-Name
object-class-identifier		Object-or-Class-Identifier} OPTIONAL,
document-characteristics	[2]	IMPLICIT Document-Characteristics OPTIONAL,
document-management-attributes	[3]	IMPLICIT Document-Management-Attributes
		OPTIONAL}

Document-Characteristics	::=	SET {	
document-application-profile		CHOICE {	
		[0]	IMPLICIT INTEGER {
			group-4-facsimile (2)}
		[4]	IMPLICIT OBJECT IDENTIFIER} OPTIONAL,

doc-appl-profile-defaults	[10]	IMPLICIT Doc-Appl-Profile-Defaults OPTIONAL,
document-architecture-class	[1]	IMPLICIT INTEGER { formatted (0), processable (1), formatted-processable (2)} OPTIONAL,
content-architecture-classes	[5]	IMPLICIT SET OF OBJECT IDENTIFIER OPTIONAL,
interchange-format-class	[6]	IMPLICIT INTEGER { if - a (0), if - b (1)} OPTIONAL,
oda-version standard-or-recommendation publication-date	[8]	IMPLICIT SEQUENCE { Character-Data, Date-and-Time} OPTIONAL,
non-basic-doc-characteristics	[2]	IMPLICIT Non-Basic-Doc-Characteristics OPTIONAL,
non-basic-struc-characteristics	[3]	IMPLICIT Non-Basic-Struc-Characteristics OPTIONAL,
additional-doc-characteristics	[9]	IMPLICIT Additional-Doc-Characteristics OPTIONAL}

Doc-Appl-Profile-Defaults	::=	SET {	
document-architecture-defaults		[0]	IMPLICIT Document-Architecture-Defaults OPTIONAL,
character-content-defaults		[1]	IMPLICIT Character-Content-Defaults OPTIONAL,
raster-gr-content-defaults		[2]	IMPLICIT Raster-Gr-Content-Defaults OPTIONAL,
geo-gr-content-defaults		[3]	IMPLICIT Geo-Gr-Content-Defaults OPTIONAL,

```
--    the following tags are reserved for additional types of content defaults:
--    [4]  videotex, for use in conjunction with CCITT  Recommendations
--    [5]  audio
--    [6]  dynamic-graphics
```

external-capt-arch-defaults	[7]	IMPLICIT SEQUENCE OF EXTERNAL OPTIONAL}}

Document-Architecture-Defaults	::=	SET {	
content-architecture-class		[0]	IMPLICIT Content-Architecture-Class OPTIONAL,
content-type		[1]	IMPLICIT Content-Type OPTIONAL,
page-dimensions		[2]	IMPLICIT Measure-Pair OPTIONAL,
transparency		[3]	IMPLICIT Transparency OPTIONAL,
colour		[4]	IMPLICIT Colour OPTIONAL,
layout-path		[5]	IMPLICIT One-Of-Four-Angles OPTIONAL,
medium-type		[6]	IMPLICIT Medium-Type OPTIONAL,
block-alignment		[7]	IMPLICIT Block-Alignment OPTIONAL,
border		[8]	IMPLICIT Border OPTIONAL,
page-position		[9]	IMPLICIT Measure-Pair OPTIONAL,
type-of-coding		[10]	IMPLICIT Type-of-Coding OPTIONAL}

Non-Basic-Doc-Characteristics	::=	SET {	
profile-character-sets		[5]	IMPLICIT OCTET STRING OPTIONAL,
comments-character-sets		[1]	IMPLICIT OCTET STRING OPTIONAL,
alternative-repr-char-sets		[6]	IMPLICIT OCTET STRING OPTIONAL,

```
--    each of these octet strings represents a string of escape sequence
```

page-dimensions	[2]	IMPLICIT SET OF Dimension-Pair OPTIONAL,
medium-types	[8]	IMPLICIT SET OF Medium-Type OPTIONAL,
layout-paths	[21]	IMPLICIT SET OF One-Of-Four-Angles OPTIONAL,
transparencies	[22]	IMPLICIT SET OF Transparency OPTIONAL,
protections	[23]	IMPLICIT SET OF Protection OPTIONAL,
block-alignments	[24]	IMPLICIT SET OF Block-Alignment OPTIONAL,
fill-orders	[25]	IMPLICIT SET OF Fill-Order OPTIONAL,
colours	[26]	IMPLICIT SET OF Colours OPTIONAL,
borders	[27]	IMPLICIT SET OF Border OPTIONAL,
page-positions	[28]	IMPLICIT SET OF Measure-Pair OPTIONAL,
types-of-coding	[29]	IMPLICIT SET OF Type-of-Coding OPTIONAL,

```
char-presentation-features              [9]   IMPLICIT SET OF Char-Presentation-Feature
                                                                              OPTIONAL,
ra-gr-presentation-features             [4]   IMPLICIT SET OF Ra-Gr-Presentation-Feature
                                                                              OPTIONAL,
geo-gr-presentation-features            [12]  IMPLICIT SET OF Geo-Gr-Presentation-Feature
                                                                              OPTIONAL,
```

-- *the following tags are reserved for additional types of presentation features:*
--   *[13] videotex, for use in conjunction with CCITT Recommendations*
--   *[14] audio*
--   *[15] dynamic-graphics*

```
character-coding-attributes             [16]  IMPLICIT SET OF Character-Coding-Attribute
                                                                              OPTIONAL,
ra-gr-coding-attributes                 [3]   IMPLICIT SET OF Ra-Gr-Coding-Attribute
                                                                              OPTIONAL,
geo-gr-coding-attributes                [17]  IMPLICIT SET OF Geo-Gr-Coding-Attribute
                                                                              OPTIONAL,
```

-- *the following tags are reserved for additional types of coding attributes:*
--   *[18] videotex, for use in conjunction with CCITT Recommendations*
--   *[19] audio*
--   *[20] dynamic-graphics*

```
ext-non-basic-pres-features             [10]  IMPLICIT SEQUENCE OF EXTERNAL
                                                                              OPTIONAL,
ext-non-basic-coding attributes         [11]  IMPLICIT SEQUENCE OF EXTERNAL
                                                                              OPTIONAL}

Non-Basic-Struc-Characteristics     ::= SET {
    number-of-objects-per-page          [0]   IMPLICIT INTEGER OPTIONAL}

Additional-Doc-Characteristics      ::= SET {
    unit-scaling                        [3]   IMPLICIT SEQUENCE {INTEGER, INTEGER}
                                                                              OPTIONAL,
    fonts-list                          [2]   IMPLICIT Fonts-List OPTIONAL}

Fonts-List                          ::= SET OF SET {
    font-identifier                         INTEGER,
    font-reference                          Font-Reference}

Document-Management-Attributes      ::= SET {
    document-description                [7]   IMPLICIT Document-Description OPTIONAL,
    dates-and-times                     [0]   IMPLICIT Dates-and-Times OPTIONAL,
    originators                         [1]   IMPLICIT Originators OPTIONAL,
    other-user-information              [2]   IMPLICIT Other-User-Information OPTIONAL,
    external-references                 [3]   IMPLICIT External-References OPTIONAL,
    local-file-references               [4]   IMPLICIT Local-File-References OPTIONAL,
    content-attributes                  [5]   IMPLICIT Content-Attributes OPTIONAL,
    security-information                [6]   IMPLICIT Security-Information OPTIONAL}

Document-Description                ::= SET {
    title                               [0]   IMPLICIT Character-Data OPTIONAL,
    subject                             [1]   IMPLICIT Character-Data OPTIONAL,
    document-type                       [2]   IMPLICIT Character-Data OPTIONAL,
    abstract                            [3]   IMPLICIT Character-Data OPTIONAL,
    keywords                            [4]   IMPLICIT SET OF Character-Data OPTIONAL,
    document-reference                  [5]   IMPLICIT Document-Reference OPTIONAL}

Character-Data                      ::= [APPLICATION 3] IMPLICIT OCTET STRING
```

-- *string of characters from the sets designated by the attribute "profile character sets",*
-- *plus carriage return and line feed*

```
Document-Reference                  ::= CHOICE {
    unique-reference                        OBJECT IDENTIFIER,
    descriptive-reference                   Character-Data}
```

Dates-and-Times	::=	SET {

```
Dates-and-Times                    ::=   SET {
    document-date-and-time              [0]   IMPLICIT Date-and-Time OPTIONAL,
    creation-date-and-time              [1]   IMPLICIT Date-and-Time OPTIONAL,
    local-filing-date-and-time          [2]   IMPLICIT SEQUENCE OF Date-and-Time
                                                  OPTIONAL,
    expiry-date-and-time                [3]   IMPLICIT Date-and-Time OPTIONAL,
    start-date-and-time                 [4]   IMPLICIT Date-and-Time OPTIONAL,
    purge-date-and-time                 [5]   IMPLICIT Date-and-Time OPTIONAL,
    release-date-and-time               [6]   IMPLICIT Date-and-Time OPTIONAL,
    revision-history                    [7]   IMPLICIT SEQUENCE OF SET {
        revision-date-and-time              [0]   IMPLICIT Date-and-Time OPTIONAL,
        version-number                      [1]   IMPLICIT Character-Data OPTIONAL
        revisors                            [2]   IMPLICIT SET OF SET {
            names                               [0]   IMPLICIT SET OF Personal-Name
                                                          OPTIONAL,
            position                            [1]   IMPLICIT Character-Data OPTIONAL,
            organization                        [2]   IMPLICIT Character-Data OPTIONAL}
                                                          OPTIONAL,
        version-reference                   [3]   Document-Reference OPTIONAL,
        user-comments                       [4]   IMPLICIT Character-Data OPTIONAL}
                                                          OPTIONAL}

Date-and-Time                      ::=   [APPLICATION 4] IMPLICIT PrintableString

    --   string of characters representing a date and, optionally, a time in accordance with
    --   ISO 8601

Originators                        ::=   SET {
    organizations                       [0]   IMPLICIT SET OF Character-Data OPTIONAL,
    preparers                           [1]   IMPLICIT SEQUENCE OF SET {
        personal-name                       [0]   IMPLICIT Personal-Name OPTIONAL,
        organization                        [1]   IMPLICIT Character-Data OPTIONAL}
                                                          OPTIONAL,
    owners                              [2]   IMPLICIT SEQUENCE OF SET {
        personal-name                       [0]   IMPLICIT Personal-Name OPTIONAL,
        organization                        [1]   IMPLICIT Character-Data OPTIONAL}
                                                          OPTIONAL,
    authors                             [3]   IMPLICIT SEQUENCE OF SET {
        personal-name                       [0]   IMPLICIT Personal-Name OPTIONAL,
        organization                        [1]   IMPLICIT Character-Data OPTIONAL}
                                                          OPTIONAL,

Personal-Name                      ::=   [APPLICATION 6] IMPLICIT SET {
    surname                             [0]   IMPLICIT Character-Data,
    givenname                           [1]   IMPLICIT Character-Data OPTIONAL,
    initials                            [2]   IMPLICIT Character-Data OPTIONAL,
    title                               [3]   IMPLICIT Character-Data OPTIONAL}

Other-User-Information             ::=   SET {
    copyright                           [0]   IMPLICIT SET OF SET {
        copyright-information               [0]   IMPLICIT SET OF Character-Data
                                                          OPTIONAL,
        copyright-dates                     [1]   IMPLICIT SET OF Date-and-Time
                                                          OPTIONAL}
                                                          OPTIONAL,
    status                              [1]   IMPLICIT Character-Data OPTIONAL,
    user-specific-codes                 [2]   IMPLICIT SET OF Character-Data OPTIONAL,
    distribution-list                   [3]   IMPLICIT SEQUENCE OF SET {
        personal-name                       [0]   IMPLICIT Personal-Name OPTIONAL,
        organization                        [1]   IMPLICIT Character-Data OPTIONAL}
                                                          OPTIONAL,
    additional-information              [5]   ANY OPTIONAL}

External-References                ::=   SET {
    references-to-other-documents       [0]   IMPLICIT SET OF Document-Reference
                                                          OPTIONAL,
    superseded-documents                [1]   IMPLICIT SET OF Document-Reference
                                                          OPTIONAL}
```

```
Local-File-Reference              ::=  SET OF SET {
        file-name                         [0]   IMPLICIT Character-Data OPTIONAL,
        location                          [1]   IMPLICIT Character-Data OPTIONAL,
        user-comments                     [2]   IMPLICIT Character-Data OPTIONAL}

Content-Attributes                ::=  SET {
        document-size                     [1]   IMPLICIT INTEGER OPTIONAL,
        number-of-pages                   [2]   IMPLICIT INTEGER OPTIONAL,
        languages                         [4]   IMPLICIT INTEGER OF Character-Data
                                                                       OPTIONAL}

Security-Information              ::=  SET {
        authorization                     CHOICE {
            person                           [0]   IMPLICIT Personal-Name,
            organization                     [4]   IMPLICIT Character-Data} OPTIONAL,
        security-classification           [1]   IMPLICIT Character-Data OPTIONAL,
        access-rights                     [2]   IMPLICIT SET OF Character-Data OPTIONAL}

                        END
```

5.7   *Identifiers and expressions*

```
Identifiers-and-Expressions { 2 8 1 5 7 }

DEFINITIONS                       ::=  BEGIN

EXPORTS   Content-Portion-Identifier, Object-or-Class-Identifier,
          Style-Identifier, Layout-Category-Name,
          Resource-Name, Binding-Name,
          Construction-Expression, Object-Id-Expression,
          Numeric-Expression, String-Expression;

IMPORTS   Layout-Object-Type
          FROM Layout-Descriptors                          -- see § 5.8
          Logical-Object-Type
          FROM Logical-Descriptors;                        -- see § 5.9

Content-Portion-Identifier        ::=  [APPLICATION 0] IMPLICIT PrintableString
```

       -- *only digits and space are used in the present version of the standard; other characters*
       -- *are reserved for extensions*

```
Object-or-Class-Identifier        ::=  [APPLICATION 1] IMPLICIT PrintableString
```

       -- *only digits and space are used in the present version of the standard; other characters*
       -- *are reserved for extensions; a "null" value is represented by an empty string*

```
Style-Identifier                  ::=  [APPLICATION 5] IMPLICIT PrintableString
```

       -- *only digits and space are used in the present version of the standard, other characters*
       -- *are reserved for extensions; a "null" value is represented by an empty string*

```
Layout-Category-Name              ::=  PrintableString
```

       -- *a "null" value is represented by an empty string*

```
Resource-Name                     ::=  PrintableString

Binding-Name                      ::=  PrintableString

Construction-Expression           ::=  CHOICE {
        construction-type                 Construction-Type,
        single-term-construction          [3]   Construction-Term}

Construction-Type                 ::=  CHOICE {
        sequence-construction             [0]   IMPLICIT Term-Sequence,
        aggregate-construction            [1]   IMPLICIT Term-Sequence,
        choice-construction               [2]   IMPLICIT Term-Sequence}
```

Term-Sequence	::=	SEQUENCE OF Construction-Term
Construction-Term	::=	CHOICE {
required-construction-factor		[0] Construction-Factor,
optional-construction-factor		[1] Construction-Factor,
repetitive-construction-factor		[2] Construction-Factor,
optional-repetitive-factor		[3] Construction-Factor}
Construction-Factor	::=	CHOICE {
object-class-identifier		Object-Class-Identifier,
construction-type		Construction-Type}
Object-Id-Expression	::=	CHOICE {
current-object-function		[0] IMPLICIT NULL,
preceding-object-function		[1] Object-Id-expression,
superior-object-function		[3] Object-Id-Expression,
current-instance-function		[4] Current-Instance-Function}
Numeric-Expression	::=	CHOICE {
numeric-literal		[0] IMPLICIT INTEGER,
increment-application		[1] Numeric-Expression,
decrement-application		[2] Numeric-Expression,
ordinal-application		[3] CHOICE {
identifier		Object-or-Class-Identifier,
expression		Object-Id-Expression},
binding-reference		[4] IMPLICIT Binding-Reference}
Binding-Reference	::=	SET {
object-reference		CHOICE {
identifier		Object-or-Class-Identifier,
expression		Binding-Selection-Function},
binding-identifier		Binding-Name}
Binding-Selection-Function	::=	CHOICE {
current-object-function		[0] IMPLICIT NULL,
preceding-function		[1] Object-Id-Expression,
superior-function		[3] Object-Id-Expression,
current-instance-function		[4] Current-Instance-Function}
Current-Instance-Function	::=	SEQUENCE {
first-parameter		CHOICE {
identifier		[0] IMPLICIT Object-or-Class-Identifier,
layout-object-type		[1] IMPLICIT Layout-Object-Type,
logical-object-type		[2] IMPLICIT Logical-Object-Type},
second-parameter		CHOICE {
identifier		Object-or-Class-Identifier,
expression		Object-Id-Expression}}
String-Expression	::=	SEQUENCE OF Atomic-String-Expression
Atomic-String-Expression	::=	CHOICE {
string-literal		[0] IMPLICIT OCTET STRING,
binding-reference		[2] IMPLICIT Binding-Reference,
make-string-application		[3] Numeric-Expression,
upper-alpha-application		[4] Numeric-Expression,
lower-alpha-application		[5] Numeric-Expression,
upper-roman-application		[6] Numeric-Expression,
lower-roman-application		[7] Numeric-Expression}

END

5.8   *Layout descriptors*

Layout-Descriptors { 2 8 1 5 8 }

DEFINITIONS   ::=   BEGIN

EXPORTS   Layout-Object-Descriptor, Layout-Class-Descriptor,
           Layout-Object-Type, Transparency, Comment-String,
           Binding-Pair, One-Of-Four-Angles, Measure-Pair,
           Dimension-Pair, Medium-Pair, Colour, Border;

```
IMPORTS   Object-or-Class-Identifier, Style-Identifier,
          Layout-Category-Name, Resource-Name, Binding-Name,
          Construction-Expression, Object-Id-Expression,
          Numeric-Expression, String-Expression
          FROM Identifiers-and-Expressions                    -- see § 5.7
          Presentation-Attributes
          FROM Style-Descriptors                              -- see § 5.10
          Default-Value-Lists-Layout
          FROM Default-Value-Lists;                           -- see § 5.11


Position-Spec              ::=   SET {
    offset                       [0]   IMPLICIT SET {
       leading                         [0]   IMPLICIT INTEGER OPTIONAL,
       trailing                        [1]   IMPLICIT INTEGER OPTIONAL,
       left-hand                       [2]   IMPLICIT INTEGER OPTIONAL,
       right-hand                      [3]   IMPLICIT INTEGER OPTIONAL}
                                                              OPTIONAL,
    separation                   [1]   IMPLICIT SET {
       leading                         [0]   IMPLICIT INTEGER OPTIONAL,
       trailing                        [1]   IMPLICIT INTEGER OPTIONAL,
       centre                          [2]   IMPLICIT INTEGER OPTIONAL}
                                                              OPTIONAL,
    alignment                    [2]   IMPLICIT INTEGER {
                                       right-hand (0), centred (1),
                                       left-hand (2)}  OPTIONAL,
    fill-order                   [3]   IMPLICIT INTEGER {
                                       normal (0), reverse (1)} OPTIONAL}

Dimension-Pair            ::=   SEQUENCE {
    horizontal                   [0]   IMPLICIT INTEGER,
    vertical                     CHOICE {
       fixed                           [0]   IMPLICIT INTEGER,
       variable                        [1]   IMPLICIT INTEGER}}

Dimension-Spec            ::=   SEQUENCE {
    horizontal                   Dimension,
    vertical                     Dimension}

Dimension                 ::=   CHOICE {
    fixed                        [0]   IMPLICIT INTEGER,
    rule-a                       [1]   IMPLICIT SET {
       minimum                         [0]   IMPLICIT INTEGER OPTIONAL,
       maximum                         [1]   IMPLICIT INTEGER OPTIONAL},
    rule-b                       [2]   IMPLICIT SET {
       minimum                         [0]   IMPLICIT INTEGER OPTIONAL,
       maximum                         [1]   IMPLICIT INTEGER OPTIONAL},
    maximum-size                 [3]   IMPLICIT NULL}

Transparency              ::=   INTEGER {transparent (0), opaque (1)}

Comment-String            ::=   OCTET STRING

    --   string of characters from the sets designated by the document profile attribute "comments
    --   character sets", plus code extension control functions, carriage return and line feed


Binding-Pair              ::=   SET {
    binding-identifier           [0]   IMPLICIT Binding-Name,
    binding-value                CHOICE {
                                 [1]   Object-Id-Expression,
                                 [2]   Numeric-Expression,
                                 [3]   String-Expression,
                                 [4]   IMPLICIT Object-or-Class Identifier,
                                 [5]   IMPLICIT INTEGER,
                                 [6]   IMPLICIT OCTET STRING}}

One-Of-Four-Angles        ::=   INTEGER {d0 (0), d90 (1), d180 (2), d270 (3)}
```

```
Measure-Pair                      ::=   SEQUENCE {
    horizontal                          [0]   IMPLICIT INTEGER,
    vertical                            [0]   IMPLICIT INTEGER}

Medium-Type                       ::=   SEQUENCE {
    nominal-page-size                   Measure-Pair OPTIONAL,
    side-of-sheet                       INTEGER {unspecified (0), recto (1), verso (2)}
                                                                      OPTIONAL}

Colour                            ::=   INTEGER {colourless (0), white (1)}

Border                            ::=   SET {
    left-hand-edge                      [0]   IMPLICIT Border-Edge OPTIONAL,
    right-hand-edge                     [1]   IMPLICIT Border-Edge OPTIONAL,
    trailing-edge                       [2]   IMPLICIT Border-Edge OPTIONAL,
    leading-edge                        [3]   IMPLICIT Border-Edge OPTIONAL}

Border-Edge                       ::=   SET {
    line-width                          [0]   IMPLICIT INTEGER OPTIONAL,
    line-type                           [1]   IMPLICIT INTEGER {
                                              invisible (0), solid (1),
                                              dashed (2), dot (3),
                                              dash-dot (4),
                                              dash-dot-dot (5)} OPTIONAL,
    freespace-width                     [2]   IMPLICIT INTEGER OPTIONAL}

    --   a "null" border edge is represented by an empty set

Layout-Object-Descriptor          ::=   SEQUENCE {
    object-type                         Layout-Object-Type OPTIONAL,
    descriptor-body                     Layout-Object-Descriptor-Body OPTIONAL}

Layout-Object-Type                ::=   INTEGER {document-layout-root (0),
                                              page-set (1), page (2),
                                              frame (3), block (4)}

Layout-Object-Descriptor-Body     ::=   SET {
    object-identifier                   Object-or-Class-Identifier OPTIONAL,
    subordinates                        [0]   IMPLICIT SEQUENCE OF NumericString
                                                                      OPTIONAL,
    content-portions                    [1]   IMPLICIT SEQUENCE OF NumericString
                                                                      OPTIONAL,
    object-class                        [2]   IMPLICIT Object-or-Class-Identifier OPTIONAL,
    position                            [3]   IMPLICIT Measure-pair OPTIONAL,
    dimensions                          [4]   IMPLICIT Dimension-Pair OPTIONAL,
    transparency                        [5]   IMPLICIT Transparency OPTIONAL,
    presentation-attributes             [6]   IMPLICIT Presentation-Attributes OPTIONAL,
    default-value-lists                 [7]   IMPLICIT Default-Value-Lists-Layout OPTIONAL,
    user-readable-comments              [8]   IMPLICIT Comment-String OPTIONAL,
    bindings                            [9]   IMPLICIT SET OF Binding-Pair OPTIONAL,
    layout-path                         [11]  IMPLICIT One-Of-Four-Angles OPTIONAL,
    imaging-order                       [12]  IMPLICIT SEQUENCE OF NumericString
                                                                      OPTIONAL,
    permitted-categories                [13]  IMPLICIT SET OF Layout-Category-Name
                                                                      OPTIONAL,

    --   a "null" value is represented by an empty set

    user-visible-name                   [14]  IMPLICIT Comment-String OPTIONAL,
    page-position                       [15]  IMPLICIT Measure-Pair OPTIONAL,
    medium-page                         [16]  IMPLICIT Medium-Type OPTIONAL,
    presentation-style                  [17]  IMPLICIT Style-Identifier OPTIONAL,
    balance                             [21]  IMPLICIT SET OF
                                              Object-or-Class-Identifier OPTIONAL,

    --   a "null" value is represented by an empty set

    colour                              [22]  IMPLICIT Colour OPTIONAL,
    border                              [23]  IMPLICIT Border OPTIONAL,
    application-comments                [25]  IMPLICIT OCTET STRING OPTIONAL}
```

```
Layout-Class-Descriptor              ::=  SEQUENCE {
    object-type                               Layout-Object-Type,
    descriptor-body                           Layout-Class-Descriptor-Body}

Layout-Class-Descriptor-Body         ::=  SET {
    object-class-identifier                   Object-or-Class-Identifier,
    generator-for-subordinates                [0]   Construction-Expression OPTIONAL,
    content-portions                          [1]   IMPLICIT SEQUENCE OF NumericString
                                                                         OPTIONAL,
    position                              CHOICE {
        fixed-position                        [3]   IMPLICIT Measure-Pair,
        variable-position                     [26]  IMPLICIT Position-Spec} OPTIONAL,
    dimensions                                [4]   IMPLICIT Dimension-Spec OPTIONAL,
    transparency                              [5]   IMPLICIT Transparency OPTIONAL,
    presentation-attributes                   [6]   IMPLICIT Presentation-Attributes OPTIONAL,
    default-value-lists                       [7]   IMPLICIT Default-Value-Lists-Layout OPTIONAL,
    user-readable-comments                    [8]   IMPLICIT Comment-String OPTIONAL,
    bindings                                  [9]   IMPLICIT SET OF Binding-Pair OPTIONAL,
    content-generator                         [10]  IMPLICIT String-Expression OPTIONAL,
    layout-path                               [11]  IMPLICIT One-Of-Four-Angles OPTIONAL,
    permitted-categories                      [13]  IMPLICIT SET OF Layout-Category-Name
                                                                         OPTIONAL,

    --   a "null" value is represented by an empty set

    user-visible-name                         [14]  IMPLICIT Comment-String OPTIONAL,
    page-position                             [15]  IMPLICIT Measure-Pair OPTIONAL,
    medium-type                               [16]  IMPLICIT Medium-Type OPTIONAL,
    presentation-style                        [17]  IMPLICIT Style-Identifier OPTIONAL,
    logical-source                            [18]  IMPLICIT Object-or-Class-Identifier OPTIONAL,
    balance                                   [21]  IMPLICIT SET OF Object-Id-Expression
                                                                         OPTIONAL,

    --   a "null" value is represented by an empty set

    colour                                    [22]  IMPLICIT Colour OPTIONAL,
    border                                    [23]  IMPLICIT Border OPTIONAL,
    resource                                  [24]  IMPLICIT Resource-Name OPTIONAL,
    application-comments                      [25]  IMPLICIT OCTET STRING OPTIONAL}

                                     END
```

5.9   *Logical descriptors*

```
Logical-Descriptors { 2 8 1 5 9 }

DEFINITIONS                          ::=  BEGIN

EXPORTS   Logical-Object-Descriptor, Logical-Class-Descriptor,
          Logical-Object-Type, Protection;

IMPORTS   Object-or-Class-Identifier, Style-Identifier,
          Resource-Name, Construction-Expression, String-Expression
             FROM Identifiers-and-Expressions                        -- see § 5.7
          Comment-String, Binding-pair
             FROM Layout-Descriptors                                 -- see § 5.8
          Presentation-Attributes
             FROM Style-Descriptors                                  -- see § 5.10
          Default-Value-Lists-Logical
             FROM Default-Value-Lists;                               -- see § 5.11

Logical-Object-Descriptor            ::=  SEQUENCE {
    object-type                               Logical-Object-Type OPTIONAL,
    descriptor-body                           Logical-Object-Descriptor-Body OPTIONAL}

Logical-Object-Type                  ::=  INTEGER {document-logical-root (0),
                                              composite-logical-object (1),
                                              basic-logical-object (2)}
```

```
Logical-Object-Descriptor-Body          ::=  SET {
    object-identifier                        Object-or-Class-Identifier OPTIONAL,
    subordinates                             [0]  IMPLICIT SEQUENCE OF NumericString
                                                       OPTIONAL,
    content-portions                         [1]  IMPLICIT SEQUENCE OF NumericString
                                                       OPTIONAL,
    object-class                             [2]  IMPLICIT Object-or-Class-Identifier OPTIONAL,
    presentation-attributes                  [6]  IMPLICIT Presentation-Attributes OPTIONAL,

    --  only  for  use  for  the  attribute  content-architecture-class,  the  content  architecture
    --  specific attributes can only be referenced by use of presentation style

    default-value-lists                      [7]  IMPLICIT Default-Value-Lists-Logical OPTIONAL,
    user-readable-comments                   [8]  IMPLICIT Comment-String OPTIONAL,
    bindings                                 [9]  IMPLICIT SET OF Binding-Pair OPTIONAL,
    content-generator                        [10] IMPLICIT String-Expression OPTIONAL,
    user-visible-name                        [14] IMPLICIT Comment-String OPTIONAL,
    presentation-style                       [17] IMPLICIT Style-Identifier OPTIONAL,
    layout-style                             [19] IMPLICIT Style-Identifier OPTIONAL,
    protection                               [20] IMPLICIT Protection OPTIONAL,
    application-comments                     [25] IMPLICIT OCTET STRING OPTIONAL}

Logical-Class-Descriptor                 ::=  SEQUENCE {
    object-type                              Logical-Object-Type,
    descriptor-body                          Logical-Class-Descriptor-Body}

Logical-Class-Descriptor-Body            ::=  SET {
    object-class-identifier                  Object-or-Class-Identifier,
    generator-for-subordonates               [0]  Construction-Expression OPTIONAL,
    content-portions                         [1]  IMPLICIT SEQUENCE OF NumericString
                                                       OPTIONAL,
    presentation-attributes                  [6]  IMPLICIT Presentation-Attributes OPTIONAL,

    --  only  for  use  for  the  attribute  content-architecture-class,  the  content  architecture
    --  specific attributes can only be referenced by use of presentation style

    default-value-lists                      [7]  IMPLICIT Default-Value-Lists-Logical OPTIONAL,
    user-readable-comments                   [8]  IMPLICIT Comment-String OPTIONAL,
    bindings                                 [9]  IMPLICIT SET OF Binding-Pair OPTIONAL,
    content-generator                        [10] IMPLICIT String-Expression OPTIONAL,
    user-visible-name                        [14] IMPLICIT Comment-String OPTIONAL,
    presentation-style                       [17] IMPLICIT Style-Identifier OPTIONAL,
    layout-style                             [19] IMPLICIT Style-Identifier OPTIONAL,
    protection                               [20] IMPLICIT Protection OPTIONAL,
    resource                                 [24] IMPLICIT Resource-Name OPTIONAL,
    application-comments                     [25] IMPLICIT OCTET STRING OPTIONAL}

Protection                               ::=  INTEGER {unprotected (0), protected (1)}


                                  END
```

5.10 *Style descriptors*

Style-Descriptors { 2 8 1 5 10 }

DEFINITIONS   ::= BEGIN

EXPORTS   Presentation-Style-Descriptor, Presentation-Attributes,
          Content-Type, Content-Architecture-Class,
          Layout-Style-Descriptor, Fill-Order, Block-Alignment;

```
IMPORTS   Object-or-Class-Identifier, Style-Identifier,
          Layout-Category-Name, Object-Id-Expression,
          FROM Identifiers-and-Expressions                        -- see § 5.7
          Comment-String, Transparency, Colour, Border,
          Layout-Object-Type
          FROM Layout-Descriptors                                 -- see § 5.8
          Character-Attributes
          FROM Character-Presentation-Attributes { 2 8 1 6 2 }    -- see Recommendation T.416
          Raster-Graphics-Attributes
          FROM Raster-Gr-Presentation-Attributes { 2 8 1 7 2 }    -- see Recommendation T.417
          Geometric-Graphics-Attributes
          FROM Geo-Gr-Presentation-Attributes { 2 8 1 8 2 }       -- see Recommendation T.418
```

```
Presentation-Style-Descriptor       ::=  SET {
    style-identifier                      Style-Identifier,
    user-readable-comments                [0]   IMPLICIT Comment-String OPTIONAL,
    user-visible-name                     [1]   IMPLICIT Comment-String OPTIONAL,
    transparency                          [2]   IMPLICIT Transparency OPTIONAL,
    presentation-attributes               [3]   IMPLICIT Presentation-Attributes OPTIONAL,
    colour                                [4]   IMPLICIT Colour OPTIONAL,
    border                                [5]   IMPLICIT Border OPTIONAL}

Presentation-Attributes             ::=  SET {
    content-type                          Content-Type OPTIONAL,
    content-architecture-class            Content-Architecture-Class OPTIONAL,
    character-attributes                  [0]   IMPLICIT Character-Attributes OPTIONAL,
    raster-graphics-attributes            [1]   IMPLICIT Raster-Graphics-Attributes OPTIONAL,
    geometric-graphics-attributes         [2]   IMPLICIT Geometric-Graphics-Attributes
                                                        OPTIONAL,

--   the  following  tags  are  reserved  for  additional  types  of  presentation  attributes:
--   [3]  videotex for use in conjunction with CCITT Recommendations
--   [4]  audio
--   [5]  dynamic-graphics

    ext-cont-arch-pres-attributes         [6]   IMPLICIT SEQUENCE OF EXTERNAL
                                                        OPTIONAL}

Content-Type                        ::=  [APPLICATION 2] IMPLICIT INTEGER {
                                            formatted-raster-graphics (1)}

Content-Architecture-Class          ::=  OBJECT IDENTIFIER

Layout-Style-Descriptor             ::=  SET {
    style-identifier                      Style-Identifier,
    user-readable-comments                [0]   IMPLICIT Comment-String OPTIONAL,
    user-visible-name                     [1]   IMPLICIT Comment-String OPTIONAL,
    layout-directives                     [4]   IMPLICIT Layout-Directives OPTIONAL}

Layout-Directives                   ::=  SET {
    indivisibility                        CHOICE {
        to-layout-object-class            [0]   IMPLICIT Object-or-Class-Identifier,
        to-layout-category                [1]   IMPLICIT Layout-Category-Name,
        to-layout-object-type             [2]   IMPLICIT Layout-Object-Type,
        null                              [15]  IMPLICIT NULL} OPTIONAL,
    separation                            [3]   IMPLICIT Separation OPTIONAL,
    offset                                [4]   IMPLICIT Offset OPTIONAL,
    fill-order                            [5]   IMPLICIT Fill-Order OPTIONAL,
    concatenation                         [6]   IMPLICIT Concatenation OPTIONAL,
    new-layout-object                     CHOICE {
        to-layout-object-class            [7]   IMPLICIT Object-or-Class-Identifier,
        to-layout-category                [8]   IMPLICIT Layout-Category-Name,
        to-layout-object-type             [9]   IMPLICIT Layout-Object-Type,
        null                              [16]  IMPLICIT NULL} OPTIONAL,
    same-layout-object                    [10]  IMPLICIT Same-Layout-Object OPTIONAL,
    layout-object-class                   [11]  IMPLICIT Object-or-Class-Identifier OPTIONAL,
```

```
            layout-category            [12]  IMPLICIT Layout-Category-Name OPTIONAL,
            synchronization            CHOICE {
                                       [13]  IMPLICIT Object-or-Class-Identifier,
                                       [17]  Object-Id-Expression,
                                       [18]  IMPLICIT NULL} OPTIONAL,
            block-alignment            [14]  IMPLICIT Block-Alignment OPTIONAL}

Separation                    ::=      SET {
        leading                        [0]   IMPLICIT INTEGER OPTIONAL,
        trailing                       [1]   IMPLICIT INTEGER OPTIONAL,
        centre                         [2]   IMPLICIT INTEGER OPTIONAL}

Offset                        ::=      SET {
        right-hand                     [0]   IMPLICIT INTEGER OPTIONAL,
        left-hand                      [1]   IMPLICIT INTEGER OPTIONAL,
        trailing                       [2]   IMPLICIT INTEGER OPTIONAL,
        leading                        [3]   IMPLICIT INTEGER OPTIONAL}

Fill-Order                    ::=      INTEGER {normal (0), reverse (1)}

Concatenation                 ::=      INTEGER {non-concatenated (0), concatenated (1)}

Same-Layout-Object            ::=      SET {
        to-logical-object              CHOICE {
                                       [0]   IMPLICIT Object-or-Class-Identifier,
                                       [4]   Object-Id-Expression,
                                       [5]   IMPLICIT NULL},
                                       CHOICE {
        to-layout-object-class         [1]   IMPLICIT Object-or-Class-Identifier,
        to-layout-category             [2]   IMPLICIT Layout-Category-Name,
        to-layout-object-type          [3]   IMPLICIT Layout-Object-Type} OPTIONAL}

Block-Alignment               ::=      INTEGER {
                                           right-hand (0), left-hand (1),
                                           centred (2), null (3)}
```

<div align="center">END</div>

5.11 *Default value lists*

Default-Value-Lists { 2 8 1 5 11 }

```
DEFINITIONS                   ::=      BEGIN

EXPORTS  Default-Value-Lists-Logical, Default-Value-Lists-Layout;

IMPORTS  Style-Identifier, Layout-Category-Name
         FROM Identifier-and-Expressions                        -- see § 5.7
         Measure-Pair, One-Of-Four-Angles, Medium-Type,
         Dimension-Pair, Transparency, Colour, Border
         FROM Layout-Descriptors                                -- see § 5.8
         Protection FROM Logical-Descriptors                    -- see § 5.9
         Presentation-Attributes
         FROM Style-Descriptors;                                -- see § 5.10

Default-Value-Lists-Layout    ::=      SET {
        page-attributes                [2]   IMPLICIT Page-Attributes OPTIONAL,
        frame-attributes               [3]   IMPLICIT Frame-Attributes OPTIONAL,
        block-attributes               [4]   IMPLICIT Block-Attributes OPTIONAL}

Default-Value-Lists-Logical   ::=      SET {
        composite-logical-attributes   [5]   IMPLICIT Composite-Logical-Attributes
                                                                    OPTIONAL,
        basic-logical-attributes       [6]   IMPLICIT Basic-Logical-Attributes OPTIONAL}

Page-Attributes               ::=      SET {
        dimensions                     <  Attribute OPTIONAL,
        transparency                   <  Attribute OPTIONAL,
        presentation-attributes        <  Attribute OPTIONAL,
```

```
        page-position                          < Attribute OPTIONAL, .
        medium-page                            < Attribute OPTIONAL,
        presentation-style                     < Attribute OPTIONAL,
        colour                                 < Attribute OPTIONAL,

Frame-Attributes                    . ::=   SET {
        position                               < Attribute OPTIONAL,
        dimensions                             < Attribute OPTIONAL,
        transparency                           < Attribute OPTIONAL,
        layout-path                            < Attribute OPTIONAL,
        permitted-categories                   < Attribute OPTIONAL,
        colour                                 < Attribute OPTIONAL,
        border                                 < Attribute OPTIONAL}

Block-Attributes                      ::=   SET {
        position                               < Attribute OPTIONAL,
        dimensions                             < Attribute OPTIONAL,
        transparency                           < Attribute OPTIONAL,
        presentation-attributes                < Attribute OPTIONAL,
        presentation-style                     < Attribute OPTIONAL,
        colour                                 < Attribute OPTIONAL,
        border                                 < Attribute OPTIONAL}

Composite-Logical-Attributes          ::=   SET {
        protection                             < Attribute OPTIONAL,
        layout-style                           < Attribute OPTIONAL}

Basic-Logical-Attributes              ::=   SET {
        presentation-attributes                < Attribute OPTIONAL,

        --  only  for  use  for  the  attribute  content-architecture-class,  the  content  architecture
        --  specific attributes can only be referenced by use of presentation style

        protection                             < Attribute OPTIONAL,
        presentation-style                     < Attribute OPTIONAL,
        layout-style                           < Attribute OPTIONAL}

Attribute                             ::=   CHOICE {
        position                               [0]   IMPLICIT Measure-Pair,
        dimensions                             [1]   IMPLICIT Dimension-Pair,
        transparency                           [2]   IMPLICIT Transparency,
        presentation-attributes                [3]   IMPLICIT Presentation-Attributes,
        layout-path                            [4]   IMPLICIT One-Of-Four-Angles,
        page-position                          [5]   IMPLICIT Measure-Pair,
        medium-type                            [6]   IMPLICIT Medium-Type,
        permitted-categories                   [7]   IMPLICIT SET OF Layout-Category-Name,
        protection                             [8]   IMPLICIT Protection,
        presentation-style                     [9]   IMPLICIT Style-Identifier,
        layout-style                           [10]  IMPLICIT Style-Identifier,
        colour                                 [11]  IMPLICIT Colour,
        border                                 [12]  IMPLICIT Border}
```

END

5.12   *Text units*

Text-Units { 2 8 1 5 12 }

DEFINITIONS   ::=   BEGIN

EXPORTS   Text-Unit, Type-Of-Coding;

```
IMPORTS    Content-Portion-Identifier
           FROM Identifiers-and-Expressions          -- see § 5.7
           Character-Coding-Attributes
           FROM Character-Coding-Attributes { 2 8 1 6 3 }    -- see Recommendation T.416
           Raster-Gr-Coding-Attributes
           FROM Raster-Gr-Coding-Attributes { 2 8 1 7 3 }    -- see Recommendation T.417
           Geo-Gr-Coding-Attributes
           FROM Geo-Gr-Coding-Attributes { 2 8 1 8 3 }       -- see Recommendation T.418

Text-Unit                           ::=    SEQUENCE {
    content-portion-attributes             Content-Portion-Attributes OPTIONAL,
    content-information                    Content-Information}

Content-Portion-Attributes          ::=    SET {
    content-identifier-layout              Content-Portion-Identifier OPTIONAL,
    content-identifier-logical             [4]   IMPLICIT Content-Portion-Identifier OPTIONAL,
    type-of-coding                         Type-Of-Coding OPTIONAL,
    coding-attributes                      CHOICE {
        character-coding-attributes        [1]   IMPLICIT Character-Coding-Attributes,
        raster-gr-coding-attributes        [2]   IMPLICIT Raster-Gr-Coding-Attributes,
        geo-gr-coding-attributes           [7]   IMPLICIT Geo-Gr-Coding-Attributes,
        videotex-coding-attributes         [8]   IMPLICIT Videotex-Coding-Attributes,

    --    the following tags are reserved for additional types of coding attributes:
    --    [9] audio
    --    [10] dynamic-graphics

        ext-contr-arch-coding-attributes   [11]  IMPLICIT EXTERNAL} OPTIONAL,

    alternative-representation             [3]   IMPLICIT Alternative-Representation OPTIONAL}

Content-Information                 ::=    OCTET STRING

Type-Of-Coding       .              ::=    CHOICE {
                                           [0]   IMPLICIT INTEGER {t6 (1)},
                                           [6]   IMPLICIT OBJECT IDENTIFIER}

Alternative-Representation          ::=    OCTET STRING

    --    string of characters from the sets designated by the document profile attribute
    --    "alternative representation character sets", plus carriage return and line feed


                                    END
```

ANNEX A

(to Recommendation T.415)

Coded representation

(Informative)

This Annex is a summary of the basic encoding rules for the abstract syntax notation ASN.1 defined in Recommendation X.209.

The coded representation of each data structure or data item that constitutes, or constitutes part of, a descriptor or a text unit consists of a type field, a length field and a value field.

If the data item concerned is an elementary data item, then the type field specifies the elementary data type, the length field specifies the length of the value field, and the value field represents the value of the data item.

If the data structure or data item concerned is not elementary, then the type field identifies the attribute or group of attributes corresponding to the data structure or data item, the length field specifies the length of the value field, and the value field consists of one or more triplets, each of which is composed of a type field, a length field and a value field, representing the subordinate data structures and data items.

The *type field* (which is called "identifier octets" in Recommendation X.209) consists of one or more bytes. The bits of the first byte are used as follows:

bits 8 and 7:	tag class	(00: universal,
		01: application,
		01: context-specific,
		11: private);

bit 6:	contents encoding form (0: simple,
	1: structured);

bits 5 to 1:	00000 to 11110: tag number;
	11111 indicates a multi-octet type field.

The following tag numbers for universal tags have been assigned in Recommendations X.208 and X.209.

Built-in data types

0: end-of-contents
1: Boolean
2: integer
3: bit string
4: octet string
5: null
6: object identifier
7: object descriptor
8: external
9: real
10: enumerated
11: encrypted
16: sequence
17: set

Defined data types

18: numeric string
19: printable string
20: teletex string
21: videotex string
22: IA5 string
23: UTC time
24: generalized time
25: graphic character string
26: general string
27: visible string

Data items of type end-of-contents, Boolean, integer or null are simple (elementary data items). Sequences and sets are structured (data structures with subordinate data items). Data items of type bit string, octet string or any of the defined data types can be either simple or structured.

The *length field* consists of one or more bytes. It takes one of three forms: short, long and indefinite. The bits of the first byte are used as follows:

bit 8:	length field form	(0: short,
		1: long or indefinite);

bits 7 to 1:	if bit 8 = 0:	number of bytes of the value field;
	if bit 8 = 1:	number of bytes of the length field following the first byte;
		0000000 indicates the indefinite form of the length field.

A data structure or data item with an indefinite length field must be structured and must be terminated by a delimiter consisting of an end-of-contents (EOC) item. An EOC item consists of two bytes: a type field of one byte and a length field of one byte. Both are equal to zero. An EOC item has no value field.

ANNEX B

(to Recommendation T.415)

Application class tag assignments

(Informative)

The application class tag assignments made in various sections of this Recommendation are summarized in the table below.

Tag	Data type	Paragraph
APPLICATION 0	Content-Portion-Identifier	5.7
APPLICATION 1	Object-or-Class-Identifier	5.7
APPLICATION 2	Content-Type	5.10
APPLICATION 3	Character-Data	5.6
APPLICATION 4	Date-and-Time	5.6
APPLICATION 5	Style-Identifier	5.7
APPLICATION 6	Personal-Name	5.6

ANNEX C

(to Recommendation T.415)

Summary of object identifiers

(Informative)

Values of ASN.1 object identifiers are assigned in various sections in this Recommendation. These are summarized below.

Object identifier value	Meaning	Paragraph
{ 2 8 0 0 }	Identifies external data type	5.1
{ 2 8 1 5 5 }	Identifies module; interchange data elements	5.5
{ 2 8 1 5 6 }	Identifies module1; document profile descriptor	5.6
{ 2 8 1 5 7 }	Identifies module; identifiers and expressions	5.7
{ 2 8 1 5 8 }	Identifies module; layout descriptors	5.8
{ 2 8 1 5 9 }	Identifies module; logical descriptors	5.9
{ 2 8 1 5 10 }	Identifies module; style descriptors	5.10
{ 2 8 1 5 11 }	Identifies module; default value lists	5.11
{ 2 8 1 5 12 }	Identifies module; text units	5.12

ANNEX D

(to Recommendation T.415)

Examples

(Informative)

The first four examples in this annex consist of data streams representing various versions of the sample document in Annex B of Recommendation T.412.

Four versions of the sample document are considered:

- *Example 1:* specific layout structure only.
  This example is specified by § B.4.1, including Figure B-7/T.412 and Table B-1/T.412.

- *Example 2:* specific logical structure only.
  This example is specified by § B.4.2, including Figure B-8/T.412 and Tables B-2/T.412 and B-3/T.412.

- *Example 3:* specific logical structure, generic logical structure and generic layout structure.
  This example is specified by § B.5, including Figures B-8/T.412, B-9/T.412, B-10/T.412 and Tables B-4/T.412, B-5/T.412, B-6/T.412 and B-7/T.412.

- *Example 4:* specific layout structure only, as generated from the structures in example 3. This example is specified by § B.6, including Figure B-11/T.412 and Table B-8/T.412.

The four examples are presented in §§ D.1, D.2, D.3 and D.4 below. The data stream constituting each example is shown in two forms of notation:

a)   the ASN.1 notation for data values defined in Recommendation T.208;

b)   the hexadecimal notation of the encoded data values, after applying the basic encoding rules defined in Recommendation T.209.

The ASN.1 notation is shown on the left and the hexadecimal notation is shown on the right of each page. The symbol LL represents a length field of which the length is unknown.

In addition, § D.5 presents an example consisting of a data stream representing the sample document profile in Annex C of Recommendation T.414.

D.1   *Example 1:*   *Sample document from Annex B of Recommendation T.412; specific layout structure only*

document-profile{	A030
specific-layout-structure "1",	810130
document-characteristics {	A22B
document-architecture-class formatted,	810100
content-architecture-classes{	A512
{2 8 2 6 0},	060458020600
{2 8 2 7 0},	060458020700
{2 8 2 8 0}},	060458020800
interchange-format-class if-b,	860101
oda-version "ISO 8613: 1988"}},	880F49534F..31393838
layout-object{	A21B
object-type document-layout-root,	020100
descriptor-body{	3116
object-identifier "1",	410131
user-visible-name "Letter",	8E064C6574746572
subordinates{	A009
"0", "1", "2"}}},	120130120131120132

```
layout-object{
      object-type page,
      descriptor-body{
            object-identifier "1 0",
            user-visible-name "Header Page",
            dimensions{
                  horizontal 9920,
                  vertical fixed 14030},
            subordinates{
            "0", "1", "2", "3", "4"}}},
layout-object{
      object-type block,
      descriptor-body{
            object-identifier "1 0 0",
            user-visible-name "Logo",
            position{
                  horizontal 710,
                  vertical 730},
            dimensions{
                  horizontal 3685,
                  vertical fixed 2495},
            presentation-attributes{
                  content-architecture-class {2 9 2 7 0},
                  raster-graphics-attributes{
                  pel-transmission-density 5}},
            content-portions{"0"}}},
text-unit{
      content-portion-attributes{
            content-identifier-layout "1 0 0 0"},
            raster-gr-coding-attributes{
                  number-of-pels-per-line 737}},
      content-information{/*Array of raster
      graphic elements for the logo*/}},
layout-object{
      object-type block,
      descriptor-body{
            object-identifier "1 0 1",
            user-visible-name "Date",
            position{
                  horizontal 5440, .
                  vertical 1275},
            dimensions{,
                  horizontal 3060,
                  vertical fixed 540},
            content-portions{"0"}}},
text-unit{
      content-portion-attributes{
            content-identifier-layout "1 0 1 0"},
      content-information{"CESSON, 26 JUNE 1985"},

layout-object{
      object-type block,
      descriptor-body{
            object-identifier "1 0 2"
            user-visible-name "Addressee",
            position{
                  horizontal 1105,
                  vertical 4310},
            dimensions{
                  horizontal 4505,
                  vertical fixed 540},
            content-portions{"0"}}},
```

```
A232
020102
312D
4103312030
8E0B4865616465720D50616765
A408
800226C0
800236CE
A00F
120130120131120132120133120134
A238
020104
3133
41053120302030
8E044C6F676F
A308
800202C6
800202DA
A408
80020E65
800209BF
A60B
060458020700
A103
820105
A103120130
A3LL
31LL
400940073120302030302030
A204
800202E1
04LL..................
..............................
A22B
020104
3126
41053120302031
8E0444617465
A308
80021540
800204FB
A408
80020BF4
8002021C
A103120130
A323
310B
40094007312030203012030
0414434553534F4E2C...
...31393835
A230
020104
312B
41053120302032
8E09416464726573736565
A308
80020451
800210D6
A408
80021199
8002021C
A103120130
```

```
text-unit{
        content-portion-attributes{
                content-identifier-layout "1 0 2 0"},
        content-information{"To members of ISO/
                                TC97/SC18/WG3"}},
layout-object{
        object-type block,
        descriptor-body{
                object-identifier "1 0 3"
                user-visible-name "Subject",
                position{
                        horizontal 1105,
                        vertical 6660},
                dimensions{
                        horizontal 7200,
                        vertical fixed 905},
                presentation-attributes{
                        character-attributes{
                                line-spacing 300}},
                content-portions{"0"}}},
text-unit{
        content-portion-attributes{
                content-identifier-layout "1 0 3 0"},
        content-information{"SUBJECT: PROPOSED
                                EXAMPLE TO CLARIFY
                                THE DOCUMENT\n
                                ARCHITECTURE MODEL"}},
layout-object{
        object-type block,
        descriptor-body{
                object-identifier "1 0 4",
                user-visible-name "Summary",
                position{
                        horizontal 2180,
                        vertical 9695},
                dimensions{
                        horizontal 5585,
                        vertical fixed 2325},
                presentation-attributes{
                        character-attributes{
                                alignment justified}},
                content-portion{"0"}}},
text-unit{
        content-portion-attributes{
                content-identifier-layout "1 0 4 0"},
        content-information{/*Formatted string of
                                SUMMARY- */}},
layout-object{
        object-type page,
        descriptor-body{
                object-identifier "1 1",
                user-visible-name{"Body Page 1"},
                dimensions{
                        horizontal 9920,
                        vertical fixed 14030},
                subordinates{
                "0", "1", "2", "3", "4"}}},
layout-object{
        object-type block,
        descriptor-body{
                object-identifier "1 1 0",
                user-visible-name "Para A",
                position{
                        horizontal 1105,
```

A32E
310B
4009400731203020322030
041F546F206D656D626572
73 .. 574733
A234
020104
312F
41053120302033
8E075375626A656374
A308
80020451
800021A04
A408
80021C20
80020389
A604

8702012C
A103120130
A3LL
310B
4009400731203020332030
04LL5456424A4543553A..........

....
....
..........4D4F44454C
A233
020104
312E
41053120302034
8E0753756D6D617279
A308
80020884
800225DF
A408
800215D1
80020915
A603

880103
A103120130
A3LL
31LL
4009400731230320342030
04LL53554D4D4152415259......

....
A232
020102
312D
4103312031
8E0B426F6479205061676520
A408
800226C0
800236CE
A00F
120130120131120132120133120134
A236
020104
3131
41053120312030
8E06506172612041
A308
80020451

```
                vertical 1105,                              80020451
            dimensions{                                     A408
                horizontal 7935,                            80021EFF
                vertical fixed 1785},                        800206F9
            presentation-attributes{                        A607
                character-attributes{
                    line-spacing 300,                        8702012C
                    alignment justified}},                   880103
            content-portions("0")}},                         A103120130
text-unit{                                                   A3LL
        content-portion-attributes{                          310B
            content-identifier-layout "1 1 0 0"},            4009400731203120302030
        content-information{/*Formatted string               04LL414141..............
                            of A's*/}},                       .................................
layout-object{                                               A236
        object-type block,                                   020104
        descriptor-body{                                     3131
            object-identifier "1 1 1",                       41053120312031
            user-visible-name "Para B",                      8E06506172612042
            position{                                        A308
                horizontal 1105,                             80020451
                vertical 3770},                              80020EBA
            dimensions{                                      A408
                horizontal 7935,                             80021EFF
                vertical fixed 1785},                         800206F9
            presentation-attributes{                         A607
                character-attributes{
                    line-spacing 400,                        87020190
                    alignment justified}},                   880103
            content-portions("0")}},                         A103120130
text-unit{                                                   A3LL
        content-portion-attributes{                          310B
            content-identifier-layout "1 1 1 0"},            4009400731203120312030
        content-information{/*Formatted string               04LL4242................
                            of B's*/}},                        .................................
layout-object{                                               A236
        object-type block,                                   020104
        descriptor-body{                                     3131
            object-identifier "1 1 2",                       41053120312032
            user-visible-name "Drawing",                     8E0744726177686E67
            position{                                        A308
                horizontal 2180,                             80020884
                vertical 6460},                              80002193C
            dimensions{                                      A408
                horizontal 5045,                             800213B5
                vertical fixed 4140},                         8002102C
            presentation-attributes{                         A606
                content-architecture-class{2 8 2 8 0},       060458020800
            content-portions("0")}},                         A103120130
text-unit{                                                   A3LL
        content-portion-attributes{                          310B
            content-identifier-layout "1 1 2 0"},            4009400731203120322030
        content-information{/*Ordered set of                 04LL.....................
                            geometric graphics               ...............................
                            content elements for             ............................
                            the diagram*/}},                 ............................
layout-object{                                               A22E
        object-type block,                                   020104
        descriptor-body{                                     3129
            object-identifier "1 1 3",                       41053120312033
            user-visible-name "Caption",                     8E0743617074696F6E
            position{                                        A308
                horizontal 2550,                             800209F6
                vertical 10800},                              80022A30
```

```
                dimensions{
                    horizontal 3970
                    vertical fixed 370},
                content-portions{"0"}}},
text-unit
                content-portion-attributes{
                    content-identifier-layout "1 1 3 0"},
                content-information{/*Formatted string
                                for the caption*/}},
layout-object{
                object-type block,
                descriptor-body{
                    object-identifier "1 1 4",
                    user-visible-name{"Para C1"},
                    position{
                            horizontal 1105,
                            vertical 11980},
                    dimensions{
                            horizontal 7935,
                            vertical fixed 1075},
                    presentation-attributes{
                            character-attributes{
                                line-spacing 300,
                                alignment justified}}},
                content-portions{"0"}}},
text-unit{
                content-portion-attributes{
                    content-identifier-layout "1 1 4 0"},
                content-information{/*Formatted string
                                of C's*/}},
layout-object{
                object-type page,
                descriptor-body{
                    object-identifier "1 2",
                    user-visible-name "Body Page 2",
                    dimensions{
                            horizontal 9920,
                            vertical fixed 14030},
                subordinates{"0", "1", "2", "3", "4"}}},
layout-object{
                object-type block,
                descriptor-body{
                    object-identifier "1 2 0",
                    user-visible-name "Para C2",
                    position{
                            horizontal 1105,
                            vertical 1105},
                    dimensions{
                            horizontal 7035,
                            vertical fixed 1275,
                    presentation-attributes{
                            character-attributes{
                                line-spacing 300,
                                alignment justified}},
                content-portions{"0"}}},
text-unit{
            content-portion-attributes{
                    content-identifier-layout "1 2 0 0"},
                content-information{/*Formatted string
                                of C's*/}},
layout-object{
            object-type block,
            descriptor-body{
```

```
A408
80020F82
80020172
A103120130
A3LL
310B
4009400731203120332030
04LL63617074696F6E....
........................................
A239
020104
3134
41053120312034
8E09506172612043283129
A308
80020451
80022ED6
A408
80021EFF
80020433
A607

8702012C
880103
A103120130
A3LL
310B
4009400731203120342030
04LL....................
............................
A232
020102
312D
4103312032
8E0B426F647920506167652032
A408
800226C0
800236CE
A00F120130120131120132120133120134
A239
020104
3134
4105320322030
8E09506172612043283229
A308
80020451
80020451
A408
80021EFF
800206F9
A607

8702012C
880103
A003120130
A3LL
310B
4009400731203230302030
04LL4343...............
............................
A236
020104
3131
```

object-identifier "1 2 1",	41053120322031
user-visible-name "Para D",	8E06506172612044
position{	A308
horizontal 1105,	80020451
vertical 3260},	80020CBC
dimensions{	A408
horizontal 7935,	80021EFF
vertical fixed 1615},	8002064F
presentation-attributes{	A607
character-attributes{	
line-spacing 300,	8702012C
alignment justified}},	880103
content-portions("0")}},	A103120130
text-unit{	A3LL
content-portion-attributes{	310B
content-identifier-layout "1 2 1 0"},	4009400731203220312030
content-information{/*Formatted string	04LL444444..............
of D's*/}},	....
layout-object{	A236
object-type block,	020104
descriptor-body{	3131
object-identifier "1 2 2",	41053120322032
user-visible-name "Ending",	8E06456E64696E67
position{	A308
horizontal 1985,	800207C1
vertical 5755},	8002167B
dimensions{	A408
horizontal 6860,	80021ACC
vertical fixed 2155},	8002086B
presentation-attributes{	A607
character-attributes{	
line-spacing 300,	8702012C
alignment justified}},	880103
content-portions("0")}},	A103120130
text-unit{	A3LL
content-portion-attributes{	310B
content-identifier-layout "1 2 2 0"},	4009400731203220322030
content-information("Formatted string of	04LL464F52524D41402045
FORMAL ENDING"}},	4E44494E47..............
layout-object{	A23D
object-type block,	020104
descriptor-body{	3138
object-identifier "1 2 3",	41053120322033
user-visible-name "signature",	8E095369676E6174757265
position{	A308
horizontal 3260,	80020CBC
vertical 8675,	800221E3
dimensions{	A408
horizontal 5585,	800215D1
vertical fixed 2495},	800209BF
presentation-attributes{	A60B
content-architecture-class{2 8 2 7 0},	060458020700
raster-graphics-attributes{	A103
pel-transmission-density 5}},	820105
content-portions("0")}},	A103120130
text-unit{	A3LL
content-portion-attributes{	310B
content-identifier-layout "1 2 3 0"},	4009400731203220332030
raster-gr-coding-attributes{	A204
number-of-pels-per-line 1117},	800204D
content-information{/*Array of	04LL....................
raster-graphics content	....
element for the	....
signature*/}},	....

```
layout-object{                                          A231
      object-type block,                                020104
      descriptor-body{                                  312C
            object-identifier "1 2 4",                  41053120322034
            user-visible-name "Name",                   8E044E616D65
            position{                                   A308
                  horizontal 5950,                      8002173E
                  vertical 11170},                      80022BA2
            dimensions{                                 A408
                  horizontal 2520,                      800209D8
                  vertical fixed 905},                  80020389
            presentation-attributes{                    A604
                  character-attributes{line-spacing 300}},  8702012C
            content-portions({"0"}}},                   A103120130
text-unit{                                              A3LL
      content-portion-attributes{                       310B
            content-identifier-layout "1 2 4 0"},       4009400731203220342030
      content-information{"Miss Aude HEA\n               04LL4D697373..........
            Document Architect"}},                      .............................
```

D.2   *Example   2:   Sample   document   from   Annex   B   of   Recommendation   T.412;   specific   logical   structure only*

```
document-profile{                                       A033
      presentation-styles "1",                          860131
      specific-logical-structure "1",                   850131
      document-characteristics {                        A228
            document-architecture-class
                  {formatted-processable},              810101
            content-architecture-classes{               A512
                  {2 8 2 6 1},                          060458020600
                  {2 8 2 7 1},                          060458020701
                  {2 8 2 8 0}},                         060458020800
            interchange-format-class if-a,              860100
            oda-version "ISO 8613: 1988"}},             880F49534F..3139838
presentation-style{                                     A70D
      style-identifier "5 0",                           4503352030
      presentation-attributes{                          A306
            character-attributes{                       A004
                  line-spacing 300}}},                  8702012C
presentation-style{                                     A711
      style-identifier "5 1"}                           4503352031
      presentation-attributes{                          A30A
            character-attributes{                       A008
                  first-line-offset 1417,               97020589
                  line-spacing 300}}},                  8702012C
presentation-style{                                     A714
      style-identifier "5 2",                           4503352032
      presentation-attributes                           A30D
            character-attributes{                       A00B
                  first-line-offset 1417,               97020589
                  alignment justified,                  880103
                  line-spacing 300}}},                  8702012C
presentation-style{                                     A714
      style-identifier "5 3",                           · 4503352033
      presentation-attributes                           A30D
            character-attributes{                       A00B
                  first-line-offset 1020,               970203FC
                  alignment justified,                  880103
                  line-spacing 300"}}},                 8702012C
presentation-style{                                     A714
      style-identifier "5 4",                           4503352034
      presentation-attributes{                          A30D
```

```
      character-attributes{                                  A00B
            first-line-offset 1417,                          97020589
            alignment justified,                             880103
            line-spacing 400"}}},                            87020190
logical-object {                                             A624
      object-type document-logical-root"},                  020100
      descriptor-body {                                      311F
            object-identifier "3",                           410133
            user-visible-name "Letter",                      8E064C6574746572
            subordinates{"0","1"},                           A0061201300120131
                  default-value-lists{                       A70A
                  Default-Value-Lists-Logical
                        basic-logical-attributes             A608
                        presentation-attributes              3106
                              content-architecture-class     060458020601
                              {2 8 2 6 1}}}},
logical-object{                                              A620
      object-type composite-logical,                         020101
      descriptor-body{                                       311B
            object-identifier "3 0",                         4103332030
            user-visible-name "Header",                      8E06486561646572
            subordinates{"0","1","2","3"}}},                 A00C120130120131120132120133
logical-object{                                              A617
      object-type basic-logical,                             020102
      descriptor-body{                                       3112
            object-identifier "3 0 0",                       41053320302030
            user-visible-name "Date",                        8E0444617465
            content-portions{"0"},                           A103120130
logical-object{                                              A61C
      object-type basic-logical,                             020102
      descriptor-body{                                       3117
            object-identifier "3 0 1",                       410533200302031
            user-visible-name "Addressee",                   8E9416464726573736565
            content-portions{"0"}}},                         A103120130
logical-object{                                              A61F
      object-type basic-logical,                             020102
      descriptor-body{                                       311A
            object-identifier "3 0 2",                       41053320302032
            user-visible-name "Subject",                     8E075375626A656374
            presentation-style "5 0",                        9103352030
            content-portions{"0"},                           A103120130
logical-object{                                              A61A
      object-type composite-logical,                         020101
      descriptor-body{                                       3115
            object-identifier "3 0 3",                       41053320302033
            user-visible-name "Summary",                     8E0753756D6D617279
            subordinates{"0"}}},                             A003120130
logical-object{                                              A62B
      object-type basic-logical,                             020102
      descriptor-body{                                       3126
            object-identifier "3 0 3 0",                     410733203020332030
            user-visible-name "Summary-paragraph",           8E1153756D6D617279
                                                             2D706172616772617068
            presentation-style "5 1",                        9103352031
            content-portions{"0"}}},                         A103120130
logical-object{                                              A627
      object-type composite-logical,                         020101
      descriptor-body{                                       3122
            object-identifier "3 1",                         4103332031
            user-visible-name "Body",                        8E04426F6479
            subordinates{"0","1","2","3","4",                A015120130120131120132120133
                        "5","6"}}},                          12013412013451 20136
```

```
logical-object{
        object-type basic-logical,
        descriptor-body{
                object-identifier "3 1 0",
                user-visible-name "Paragraph A"
                presentation-style "5 2",
                content-portions{"0"},
logical-object{
        object-type basic-logical,
        descriptor-body{
                object-identifier "3 1 1",
                user-visible-name "Paragraph B"
                presentation-style "5 2",
                content-portions{"0"},
logical-object{
        object-type composite-logical,
        descriptor-body{
                object-identifier "3 1 2",
                user-visible-name "Figure"
                subordinates{"0","1"}}},
logical-object{
        object-type basic-logical,
        descriptor-body{
                object-identifier "3 1 2 0",
                user-visible-name "Drawing"
                presentation-attributes{
                content-architecture-class {2 8 2 8 0}},
                content-portions{"0"}}},
logical-object{
        object-type basic-logical,
        descriptor-body{
                object-identifier "3 1 2 1",
                user-visible-name "Caption"
                content-portions {"0"}}},
logical-object{
        object-type basic-logical,
        descriptor-body{
                object-identifier "3 1 3",
                user-visible-name "Paragraph C"
                presentation-style "5 2",
                content-portions{"0"}}},
logical-object{
        object-type basic-logical,
        descriptor-body{
                object-identifier "3 1 4",
                user-visible-name "Paragraph D"
                presentation-style "5 2",
                content-portions{"0"}}},
logical-object{
        object-type basic-logical,
        descriptor-body{
                object-identifier "3 1 5",
                user-visible-name "Ending"
                presentation-style "5 3",
                content-portions{"0"}}},
logical-object{
        object-type composite-logical,
        descriptor-body{
                object-identifier "3 1 6",
                user-visible-name "Signature and Name",

                subordinates{"0","1"}}},
```

```
A623
020102
311E
41053320312030
8E0B506172616772617061706820412041
9103352032
A103120130
A623
020102
311E
41053320312031
8E0B506172616772617261706820462
9103352032
A103120130
A61C
020101
3117
41053320322032
8E06466967757265
A006120130120131
A624
020102
311F
410733203120322030
8E0744726177686E67
A606
A60458020800
A103120130
A61C
020102
3117
410733203120322031
8E0743517074696F6E
A103120130
A623
020102
311E
41053320312033
8E0B50617261677261706820 43
91033522032
A103120130
A623
020102
311E
41053320312034
8E0B506172616772617061706820 43
9103352032
A103120130
A61E
020102
3119
41053320312035
8E06456E64696E67
9102252033
A103120130
A628
020101
3123
41053320312036
8E125369676E61747572652D
4E616D65
A006120130120131
```

```
logical-object{
        object-type basic-logical,
        descriptor-body{
                object-identifier "3 1 6 0",
                user-visible-name "Signature",
                presentation-attributes{
                        content-architecture-class
                                {2 8 2 7 1}},
                content-portions{"0"}}},
logical-object{
        object-type basic-logical,
        descriptor-body{
                object-identifier "3 1 6 1",
                user-visible-name "Name",
                presentation-style "5 0",
                content-portions{"0"}}},
text-unit{
        content-portion-attributes{
                content-identifier-logical "3 0 0 0"},
        content-information{"CESSON, 26 JUNE 1985"}},
text-unit{
        content-portion-attributes{
                content-identifier-logical "3 0 1 0"},
        content-information{"To members
                        of TC97/SC18/WG3"}},
text-unit{
        content-portion-attributes{
                content-identifier-logical "3 0 2 0"},
        content-information{"SUBJECT: PROPOSED
                        EXAMPLE TO CLARIFY THE
                        DOCUMENT ARCHITECTURE
                        MODEL"}},
text-unit{
        content-portion-attributes{
                content-identifier-logical "3 0 3 0 0"},
        content-information{/*Unformatted string of
                                SUMMARY-*/}},
text-unit{
        content-portion-attributes{
                content-identifier-logical "3 1 0 0"},
        content-information{/*Unformatted string of
                /              A's*/}},
text-unit{
        content-portion-attributes{
                content-identifier-logical "3 1 1 0"},
        content-information{/*Unformatted string of
                                B's*/}},
text-unit{
        content-portion-attributes{
                content-identifier-logical "3 1 2 0 0"},
        content-information{/*Ordered set of
                        geometric-graphics
                        content elements for
                        the diagram*/}},
text-unit{
        content-portion-attributes{
                content-identifier-logical "3 1 2 1 0"}
        content-information{/*Unformatted string
                                for the caption*/}},
text-unit{
        content-portion-attributes{
                content-identifier-logical "3 1 3 0"},
        content-information{/*Unformatted string of
                                C's*/}},
```

```
A626
020102
3121
410733203120362030
8E095369676E6174757265
A606
060458020701

A103120130
A61E
020102
3119
410733203120362031
8E044E616D65
9103352030
A103120130
A321
3109
840733203020312030
0414434553534F4E2C.......31393835
A32C
3109
840730203020312030

041F546F206D656D62657273...4733
A3LL
3109
840733203020322030
04LL5456424A4543553A........

.............4D4F44454C
A3LL
310B
840933203020203320302030
04LL53554D4D415259

A3LL
3109
840733203120302030

04LL414141.............
A3LL
3109
840733203120312030

04LL4242424242......
A3LL
310B
840933203120320302030
04LL4242424242.............

A3LL
310B
840933203120320312030
04LL63617074696F6E.....

A3LL
3109
840733203120332030

04LL4343434343...........
```

```
text-unit{                                              A3LL
      content-portion-attributes{                       3109
            content-identifier-logical "3 1 4 0"},      840733203120342030
      content-information{/*Unformatted string of
                                    D's*/}},             04LL44444444............
text-unit{                                              A3LL
      content-portion-attributes{                       3109
            content-identifier-logical "3 1 5 0"},      840733203120352030
      content-information{/*Unformatted string           04LL464F524D414C2D
                        for Ending*/}},                  454E44494E47........
text-unit{                                              A3LL
      content-portion-attributes{                       3111
            content-identifier-logical "3 1 6 0 0"},     4109332031203620302030
            raster-gr-coding-attributes{                 A204
                  number-of-pels-per-line 1117}},        8002045D
      content-information{"/*Array of                    04LLZZZZZZZZZZZZ........
                        raster-graphics                 ............................................
                        content elements for            ............................................
                        the signature*/}},              ............................................
text-unit{                                              A3LL
      content-portion-attributes{                       310B
            content-identifier-logical "3 1 6 1 0"},     840933203120362031 2030
      content-information{"Miss Aude HEA Document         04LL4D697373............
                        Architect"}},                   ................................
```

D.3   *Example 3:*   *Sample document from Annex B of Recommendation T.412; generic layout, generic logical and specific logical structures*

```
document-profile{                                       A03C
      generic-layout-structure "1",                     800131
      generic-logical-structure "1",                    840131
      presentation-styles "1",                          860131
      layout-styles "1",                                870131
      specific-logical-structure "1",                   850131
      document-characteristics {                        A22B
            document-architecture-class processable,    810101
            content-architecture-classes{               A512
                  {2 8 2 6 1},                           060458020601
                  {2 8 2 7 0},                           060458020700
                  {2 8 2 8 0}},                          060458020800
            interchange-format-class if-a,              860101
            oda-version "ISO 8613 : 1988"}},            880F49534F..31393838
layout-object-class{                                    A122
      object-type document-layout-root,                 020100
      descriptor-body{                                  311D
            object-class-identifier "0",                410130
            user-visible-name "Letter",                 8E064C6574746572
            generator-for-subordinates{                 A010
                  sequence-construction                 A00E
                  required-construction-factor          A005
                        object-class-identifier "0 0",  4103302030
                  repetitive-construction-factor        A205
                        object-class-identifier "0 1"}}}, 4103302031
layout-object-class{                                    A14D
      object-type page,                                 020102
      descriptor-body{                                  3148
            object-class-identifier "0 0",              4103302030
            user-visible-name "Header",                 8E06486561646572
            dimensions{                                 A408
                  horizontal fixed 9920,                800226C0
                  vertical fixed 14030},                800236CE
            generator-for-subordinates{                 A02F
                  sequence-construction                 A02D
```

```
                    required-construction-factor                          A007
                        object-class-identifier "0 0 0",                 41053020302030
                    required-construction-factor                          A007
                        object-class-identifier "0 0 1",                 41053020302031
                    required-construction-factor                          A007
                        object-class-identifier "0 0 2",                 41053020302032
                    required-construction-factor                          A007
                        object-class-identifier "0 0 3",                 41053020302033
                    required-construction-factor                          A007
                        object-class-identifier "0 0 4"}}},              41053020302034
layout-object-class{                                                      A12F
     object-type frame,                                                  020103
     descriptor-body{                                                    312A
          object-class-identifier "0 0 0",                              41053020302030
          position{
               fixed-position{                                           A308
                    horizontal 710,                                      800202C6
                    vertical 730}},                                      800202DA
          dimensions{                                                    A408
               horizontal fixed 3685,                                    80020E65
               vertical fixed 2495},                                     800209BF
          generator-for-subordinates{                                    A00D
               single-term-construction                                  A30B
                    required-construction-factor                         A009
                         object-class-identifier
                              "0 0 0"}}},                                410730203020302030
layout-object-class{                                                     A121
     object-type block,                                                  020104
     descriptor-body{                                                    311C
          object-class-identifier{"0 0 0"},                             410730203020302030
          user-visible-name{"Logo"},                                    8E044C6F676F
          presentation-attributes{                                       A606
               content-architecture-class
               {2 8 2 7 0},                                             060458020700
          content-portions{"0"}}},                                      A103120130
layout-object-class{                                                     A126
     object-type frame,                                                  020103
     descriptor-body{                                                    3121
          object-class-identifier "0 0 1",                             41053020302031
          user-visible-name "Date",                                    8E0444617465
          position{
               fixed-position{                                           A308
                    horizontal 5045,                                     800213B5
                    vertical 565}},                                      80020235
          dimensions{                                                    A408
               horizontal fixed 3970,                                    80020F82
               vertical fixed 1615}}},                                  8002064F
layout-object-class{                                                     A12B
     object-type frame,                                                  020103
     descriptor-body{                                                    3126
          object-class-identifier "0 0 2",                             41053020302032
          user-visible-name "Addressee",                               8E09416464726573736565
          position{
               fixed-position{                                           A308
                    horizontal 1105,                                     80020451
                    vertical 4310}},                                     800210D6
          dimensions{                                                    A408
               horizontal fixed 5395,                                    80021513
               vertical fixed 1415}}},                                  80020587
layout-object-class{                                                     A129
     object-type frame,                                                  020103
     descriptor-body{                                                    3124
          object-class-identifier "0 0 3",                             41053020302033
```

```
            user-visible-name "Subject",          8E075375626A656374
            position{
                fixed-position{                    A308
                    horizontal 1105,               80020451
                    vertical 6660}},               80021A04
                dimensions{                        A408
                    horizontal fixed 7200,         80021C20
                    vertical fixed 1785}}},        800206F9
layout-object-class{                               A129
    object-type frame,                             020103
    descriptor-body{                               3124
        object-class-identifier "0 0 4",           41053020302034
        user-visible-name "Summary",               8E0753756D6D617279
        position{
            fixed-position{                        A308
                horizontal 2180,                   80020884
                vertical 9695}},                   800225DF
            dimensions{                            A408
                horizontal fixed 6290,             80021892
                vertical fixed 3570}}},            80020DF2
layout-object-class{                               A127
    object-type page,                              020102
    descriptor-body{                               3122
        object-class-identifier "0 1",             4103302031
        user-visible-name "Body",                  8E04426F6479
        dimensions{                                A408
            horizontal fixed 9920,                 800226C0
            vertical fixed 14030}},                800236CE
        generator-for-subordinates{                A00B
            single-term-construction               A309
                required-construction-factor       A007
                    object-class-identifier "0 1 0"}}},   41053020312030
layout-object-class{                               A126
    object-type frame,                             020103
    descriptor-body{                               3121
        object-class-identifier "0 1 0",           41053020312030
        user-visible-name "Body",                  8E04426F6479
        position{
            fixed-position{                        A308
                horizontal 565,                    80020235
                vertical 565}},                    80020235
            dimensions{                            A408
                horizontal fixed 8815,             8002226F
                vertical fixed 12870}}},           80023246
logical-object-class{                              A522
    object-type document-logical-root,             020100
    descriptor-body{                               311D
        object-class-identifier "2",               410132
        user-visible-name "Letter",                8E064C6574746572
        generator-for-subordinates{                A010
            sequence-construction                  A00E
                required-construction-factor       A005
                object-class-identifier "2 0",     4103322030
                required-construction-factor       A005
                object-class-identifier "2 1"}}},  4103322031
logical-object-class{                              A53A
    object-type composite-logical,                 020101
    descriptor-body{                               3135
        object-class-identifier "2 0",             4103322030
        user-visible-name "Header",                8E06486561646572
        generator-for-subordinates{                A026
            sequence-construction                  A024
                required-construction-factor       A007
                    object-class-identifier "2 0 0",   41053220302030
```

```
        required-construction-factor                              A007
            object-class-identifier "2 0 1",                      41053220302031
        required-construction-factor                              A007
            object-class-identifier "2 0 2",                      41053220302032
        required-construction-factor                              A007
            object-class-identifier "2 0 3"}}},                   41053220302033
logical-object-class{                                             A51F
    object-type basic-logical,                                    020102
    descriptor-body{                                              311A
        object-class-identifier "2 0 0",                          41053220302030
        user-visible-name "Date",                                 8E0444617465
        layout-style "4 0",                                       9303342030
        presentation-attributes{                                  A606
            content-architecture-class {2 8 2 6 1}}},             060458020601
logical-object-class{                                             A524
    object-type basic-logical,                                    020102
    descriptor-body{                                              311F
        object-class-identifier "2 0 1",                          41053220302031
        user-visible-name "Addressee",                            8E09416464726573736565
        layout-style "4 1",                                       9303342031
        presentation-attributes{                                  A606
            content-architecture-class {2 8 2 6 1}}},             060458020601
logical-object-class{                                             A527
    object-type basic-logical,                                    020102
    descriptor-body{                                              3122
        object-class-identifier "2 0 2",                          41053220302032
        user-visible-name "Subject",                              8E075375626A656374
        layout-style "4 2",                                       9303342032
        presentation-style "5 0",                                 9103352030
        presentation-attributes{                                  A606
            content-architecture-class {2 8 2 6 1}}}},            060458020601
logical-object-class{                                             A529
    object-type composite-logical,                                020101
    descriptor-body{                                              3124
        object-class-identifier "2 0 3",                          41053220302033
        user-visible-name "Summary",                              8E0753756D6D617279
        layout-style "4 3",                                       9303342033
        generator-for-subordinates{                               A00D
            single-term-construction                              A30B
                repetitive-construction-factor                    A309
                    object-class-identifier
                        ("2 0 3 1"}}},                            410732203020332031
logical-object-class{                                             A533
    object-type basic-logical,                                    020102
    descriptor-body{                                              312E
        object-class-identifier "2 0 3 1",                        410732203020332031
        user-visible-name "Summary-paragraph",                    8E1153756D6D6172792D706172612
                                                                  6772617068
        layout-style "4 4",                                       9303342034
        presentation-style "5 1",                                 9103352031
        presentation-attributes{                                  A606
            content-architecture-class                            060458020601
            {2 8 2 6 1}}}},
logical-object-class{                                             A531
    object-type composite-logical,                                020101
    descriptor-body{                                              312C
        object-class-identifier "2 1",                            4103322031
        user-visible-name "Body",                                 8E04426F6479
        layout-style "4 5",                                       9303342035
            generator-for-subordinates{                           A02A
                sequence-construction                             A028
                    repetitive-construction-factor                A214
                        choice-construction                       A212
```

```
                      required-construction-factor          A007
                          object-class-identifier
                              "2 1 1",                      41053220312031
                      required-construction-factor          A007
                          object-class-identifier
                              "2 1 0",                      41053220312030
                      required-construction-factor          A007
                          object-class-identifier
                              "2 1 2",                      41053220312032
                      required-construction-factor          A007
                          object-class-identifier
                              "2 1 3"}}},                   41053220312033
logical-object-class{                                       A533
    object-type composite-logical,                         020101
    descriptor-body{                                       312E
        object-class-identifier "2 1 0",                   41053220312030
        user-visible-name "Figure",                        8E06466967757265
        layout-style "4 6",                                9303342036
        generator-for-subordinates{                        A018
            sequence-construction                          A016
                required-construction-factor               A009
                    object-class-identifier
                        "2 1 0 0",                          410732203120302030
                required-construction-factor               A009
                    object-class-identifier
                        "2 1 0 1"}}},                       410732203120302031
logical-object-class{                                       A524
    object-type basic-logical,                             020102
    descriptor-body{                                       311F
        object-class-identifier "2 1 0 0",                 410732203120302030
        user-visible-name "Drawing",                       8E0744726177686E67
        presentation-attributes{                           A606
            content-architecture-class                     060458020800
            {2 8 2 8 0},
        layout-style "4 7"}},                              9303342037
logical-object-class{                                       A524
    object-type basic-logical,                             020102
    descriptor-body{                                       311F
        object-class-identifier "2 1 0 1",                 410732203120302031
        user-visible-name "Caption",                       8E0743617074696F6E
        layout-style "4 8",                                9303342038
        presentation-attributes{                           A606
            content-architecture-class                     060458020601
            {2 8 2 6 1}}}},
logical-object-class{                                       A52E
    object-type basic-logical,                             020102
    descriptor-body{                                       3129
        object-class-identifier "2 1 1",                   41053220312031
        user-visible-name "Body-paragraph",                8E0E426F64792D706172616772617068
        layout-style "4 9",                                9303342039
        presentation-style "5 3",                          9103352033
        presentation-attributes{                           A606
            content-architecture-class                     060458020601
            {2 8 2 6 1}}}},
logical-object-class{                                       A52C
    object-type basic-logical,                             020102
    descriptor-body{                                       3127
        object-class-identifier "2 1 2",                   41053220312032
        user-visible-name "Ending",                        8E06456E64696E67
        layout-style "4 10",                               930434203130
        presentation-style "5 4",                          9103352034
        content-portions{"0"},                             A103120130
        presentation-attributes{                           A606
```

```
                content-architecture-class{            060458020601
                    {2 8 2 6 1}}}},
logical-object-class{                                  A538
        object-type composite-logical,                020101
        descriptor-body{                              3133
                object-class-identifier "2 1 3",      41053220312033
                user-visible-name "Signature-and-Name", 8E125369676E61747572652D616E
                                                      642D4E616D65
                generator-for-subordinates{           A018
                        sequence-construction         A016
                                required-construction-factor  A009
                                        object-class-identifier
                                                "2 1 3 0",    410732203120332030
                                required-construction-factor  A009
                                        object-class-identifier
                                                "2 1 3 1"}}},  410732203120332031
logical-object-class{                                  A527
        object-type basic-logical,                    020102
        descriptor-body{                              3122
                object-class-identifier "2 1 3 0",    410732203120332030
                user-visible-name "Signature",        8E095369676E6174757265
                presentation-attributes{              A606
                        content-architecture-class    060458020701
                        {2 8 2 7 1}},
                layout-style "4 11"}},                 930434203131
logical-object-class{                                  A527
        object-type basic-logical,                    020101
        descriptor-body{                              3122
                object-class-identifier "2 1 3 1",    410732203120332031
                user-visible-name "Name",             8E044E616D65
                layout-style "4 12",                  930434203132
                presentation-style "5 0",             9103352030
                presentation-attributes{              A606
                        content-architecture-class    060458020601
                        {2 8 2 6 1},
text-unit{                                             A3LL
        content-portion-attributes{                   3111
                content-identifier-layout "0 0 0 0"}, 4009400730203020302030
                raster-gr-coding-attributes{          A204
                        number-of-pels-per-line 737}}, 800202E1
        content-information{/*Array of
                        raster-graphics               04LL........
                        content elements for          ................
                        the logo*/ }},                ................
text-unit{                                             A3LL
        content-portion-attributes{                   3109
                content-identifier-logical "2 1 2 0"}, 400732203120322030
        content-information{/*Unformatted string       04LL........
                        of ending*/}},                ................
presentation-style{                                    A70D
        style-identifier "5 0",                       4503352030
        presentation-attributes{                      A306
                character-attributes{                 A004
                        line-spacing 300}}},          8702012C
presentation-style{                                    A710
        style-identifier "5 1",                       4503352031
        presentation-attributes{                      A309
                character-attributes{                 A007
                        first-line-offset 1417,       97020589
                        alignment justified}}},       880103
presentation-style{                                    A714
        style-identifier "5 3",                       4503352033
        presentation-attributes{                      A30D
```

```
        character-attributes{
                first-line-offset 1417,
                alignment justified,
                line-spacing 300}}},
presentation-style{
        style-identifier "5 4",
        presentation-attributes{
                character-attributes{
                        first-line-offset 1020,
                        alignment justified,
                        line-spacing 300}}},
layout-style{
        style-identifier "4 0",
        layout-directives{
                layout-object-class "0 0 1",
                offset{
                        trailing 710,
                        right-hand 395}}},
layout-style{
        style-identifier "4 1",
        layout-directives{
                layout-object-class "0 0 2"}},
layout-style{
        style-identifier "4 2",
        layout-directives{
                layout-object-class "0 0 3"}},
layout-style{
        style-identifier "4 3",
        layout-directives{
                layout-object-class "0 0 4"}},
layout-style{
        style-identifier "4 4",
        layout-directives{
                offset{
                        left-hand 705}}},
layout-style{
        style-identifier "4 5",
        layout-directives{
                new-layout-object{
                        to-layout-object-class "0 1"}}},
layout-style{
        style-identifier "4 6",
        layout-directives{
                indivisibility{
                        to-layout-object-class "0 1 0"}}},
layout-style{
        style-identifier "4 7",
        layout-directives{
                offset{
                        right-hand 1615,
                        left-hand 2155},
                separation{
                        trailing 905"}}},
layout-style{
        style-identifier "4 8",
        layout-directives{
                offset{
                        right-hand 1985,
                        left-hand 2860},
                separation{
                        trailing 200}}},
layout-style{
        style-identifier "4 9",
        layout-directives{
```

```
A00B
97020589
880103
8702012C
A714
4503352034
A30D
A00B
970203FC
880103
8702012C
A818
4503342030
A411
8B053020302031
A408
820202C6
8002018B
A80E
4503342031
A407
8B053020302032
A80E
4503342032
A407
8B053020302033
A80E
4503342033
A407
8B053020302034
A80D
4503342034
A406
A404
810202C1
A80C
4503342035
A405

8703302031
A80E
4503342036
A407

80053020312030
A817
4503342037
A410
A408
8002064F
8102086B
A304
81020389
A817
4503342038
A410
A408
800207C1
81020B26
A304
810201C8
A81F
4503342039
A418
```

`        offset{`	A410
`            trailing 540,`	8202021C
`            leading 280,`	83020118
`            right-hand 540,`	8002021C
`            left-hand 340},`	81020154
`        separation{`	A304
`            trailing 880}}},`	81020370
`layout-style{`	A818
`    style-identifier "4 10",`	450434203130
`    layout-directives{`	A410
`        offset{`	A408
`            right-hand 1420,`	8002058C
`            left-hand 535},`	81020217
`        separation{`	A304
`            trailing 880}}},`	81020370
`layout-style{`	A818
`    style-identifier "4 11",`	450434203131
`    layout-directives{`	A410
`        offset{`	A408
`            right-hand 2695,`	80020A87
`            left-hand 535},`	81020217
`        separation{`	A304
`            trailing 765}}},`	810202FD
`layout-style{`	A812
`    style-identifier "4 12",`	450434203132
`    layout-directives{`	A40A
`        offset{`	A408
`            right-hand 5385,`	80021509
`            left-hand 910}}},`	8102038E
`logical-object{`	A61B
`    object-type document-logical-root,`	020100
`    descriptor-body{`	3118
`        object-identifier "3",`	410103
`        object-class "2",`	820132
`        user-visible-name "Letter",`	8E064C6574746572
`        subordinates{"0","1"}}},`	A006120130120131
`logical-object{`	A625
`    object-type composite-logical}`	020101
`    descriptor-body{`	3120
`        object-identifier "3 0",`	4103302030
`        object-class "2 0",`	8203322030
`        user-visible-name "Header",`	8E06486561646572
`        subordinates{"0","1","2","3"}}},`	A00C120130120131120132120133
`logical-object{`	A61E
`    object-type basic-logical,`	020102
`    descriptor-body{`	3119
`        object-identifier "3 0 0",`	41053320302030
`        object-class "2 0 0",`	82053220302030
`        user-visible-name "Date",`	8E0444617465
`        content-portions{"0"}}},`	A103120130
`logical-object{`	A623
`    object-type basic-logical,`	020102
`    descriptor-body{`	311E
`        object-identifier "3 0 1",`	41053320302031
`        object-class "2 0 1",`	82053220302031
`        user-visible-name "Addressee",`	8E09416464726573736565
`        content-portions{"0"}}},`	A103120130
`logical-object{`	A621
`    object-type basic-logical,`	020102
`    descriptor-body{`	311C
`        object-identifier "3 0 2",`	41053320302032
`        object-class "2 0 2",`	82053220302032
`        user-visible-name "Subject",`	8E075375626A656374
`        content-portions{"0"}}},`	A103120130

```
logical-object{                                    A621
    object-type composite-logical,                 020101
    descriptor-body{                               311C
        object-identifier "3 0 3",                 41053320302033
        object-class "2 0 3",                      82053320302033
        user-visible-name "Summary",               8E0753756D6D617279
        subordinates{"0"}}},                       A003120130
logical-object{                                    A62F
    object-type basic-logical,                     020102
    descriptor-body{                               312A
        object-identifier "3 0 3 0",               410733203020332030
        object-class "2 0 3 1",                    820732203020332031
        user-visible-name "Summary-paragraph",     8E1153756D6D617279
                                                   2D706172616772617068
        content-portions{"0"}}},                   A103120130
logical-object{                                    A62C
    object-type composite-logical,                 020101
    descriptor-body{                               3127
        object-identifier "3 1",                   4103332031
        object-class "2 1",                        8203322031
        user-visible-name "Body",                  8E04426F6479
        subordinates{"0","1","2","3","4","5","6"}}},  A015120130120131120132120133
                                                   120134120135120136
logical-object{                                    A625
    object-type basic-logical,                     020102
    descriptor-body{                               3120
        object-identifier "3 1 0",                 41053320312030
        object-class "2 1 1",                      82053220312031
        user-visible-name "Paragraph A",           8E0B5061726167726170682041
    content-portions{"0"}}},                       A103120130
logical-object{                                    A62D
    object-type basic-logical,                     020102
    descriptor-body{                               3128
    object-identifier "3 1 1",                     41053320312031
    object-class "2 1 1",                          82053220312031
    user-visible-name "Paragraph B",               8E0B5061726167726170682042
    presentation-attributes{                       A606
        character-attributes{                      A004
            line-spacing 400}},                    87020190
    content-portions{"0"}}},          ·            A103120130
logical-object{                                    A623
    object-type composite-logical"}                020101
    descriptor-body{                               311E
        object-identifier "3 1 2",                 41053320312032
        object-class "2 1 0",                      82053220312030
        user-visible-name "Figure",                8E06466967757265
        subordinates{"0","1"}}},                   A006120130120131
logical-object{                                    A625
    object-type basic-logical,                     020102
    descriptor-body{                               3120
        object-identifier "3 1 2 0",               410733203120322030
    object-class "2 1 0 0",                        820732203120302030
        user-visible-name "Drawing",               8E0744726177686E67
        content-portions{"0"}}},                   A103120130
logical-object{                                    A625
    object-type basic-logical,                     020102
    descriptor-body{                               3120
        object-identifier "3 1 2 1",               410733203120322031
        object-class "2 1 0 1",                    820732203120302031
        user-visible-name "Caption",               8E0743617074696F6E
        content-portions{"0"}}},                   A103120130
logical-object{                                    A625
    object-type basic-logical,                     020102
    descriptor-body{                               3120
```

```
        object-identifier "3 1 3",
        object-class "2 1 1",
        user-visible-name "Paragraph C",
        content-portions{"0"}}},
logical-object{
    object-type basic-logical,
    descriptor-body{
        object-identifier "3 1 4",
        object-class "2 1 1",
        user-visible-name "Paragraph D",
        content-portions{"0"}}},
logical-object{
    object-type basic-logical,
    descriptor-body{
        object-identifier "3 1 5",
        object-class "2 1 2",
        user-visible-name "Ending"}},
logical-object{
    object-type composite-logical,
    descriptor-body{
        object-identifier "3 1 6",
        object-class "2 1 3",
        user-visible-name "Signature and Name",

        subordinates{"0","1"}}},
logical-object{
    object-type basic-logical,
    descriptor-body{
        object-identifier "3 1 6 0",
        object-class "2 1 3 0",
        user-visible-name "Signature",
        content-portions{"0"}}},
logical-object{
    object-type basic-logical,
    descriptor-body{
        object-identifier "3 1 6 1",
        object-class "2 1 3 1",
        user-visible-name "Name",
        content-portions{"0"}}},
text-unit{
    content-portion-attributes{
        content-identifier-logical "3 0 0 0"},
    content-information{"CESSON, 26 JUNE 1985"}},
text-unit{
    content-portion-attributes{
        content-identifier-logical "3 0 1 0"},
    content-information{"To members of TC97
                            /SC18 /WG3"}},
text-unit{
    content-portion-attributes{
        content-identifier-logical "3 0 2 0"},
    content-information{"SUBJECT: PROPOSED
                        EXAMPLE TO CLARIFY
                        THE DOCUMENT
                        ARCHITECTURE MODEL"}},
text-unit{
    content-portion-attributes{
        content-identifier-logical "3 0 3 0 0"},
    content-information{/*Unformatted string
                        of SUMMARY-*/}},
text-unit{
    content-portion-attributes{
        content-identifier-logical "3 1 0 0"},
    content-information{/*Unformatted string of
                                A's*/}},
```

```
41053320312033
82053220312031
8E0B506172616772617068 2043
A103120130
A625
020102
3120
41053320312034
82053220312031
8E0B5061726172617070682043
A103120130
A61B
020102
3116
41053320312035
82053220312032
8E06456E64696E67
A62F
020101
312A
41053320312036
82053220312033
8E125369676E61747572652652D
4E616D65
A006120130120131
A627
020102
3122
410733203120362030
820732203120332030
8E095369676E6174757265
A103120130
A622
020102
311D
410733203120362031
820732203120332031
8E044E616D65
A103120130
A321
3109
840733203020312030
0414434553534F4E2C....31393835
A32C
3109
840730203020312030
041F546F206D656D62657273........
..4733
A3LL
3109
840733203020322030

04LL5456424A4543553A........

...........4D4F44454C
A3LL
310B
840933203020332030302030
04LL53554D4D415259.....

A3LL
3109
840733203120302030

04LL414141............
```

text-unit{	A3LL
content-portion-attributes{	3109
content-identifier-logical "3 1 1 0"},	840733203120312030
content-information{/*Unformatted string	
of B's*/}},	04LL4242424242......
text-unit{	A3LL
content-portion-attributes{	310B
content-identifier-logical "3 1 2 0 0"},	8409332031203220302030
content-information{/*Ordered set of	
geometric-graphics content	04LL4242424242............
elements for the diagram*/}},	....
text-unit{	A3LL
content-portion-attributes{	310B
content-identifier-logical "3 1 2 1 0"},	8409332031203220312030
content-information{/*Unformatted string	
for the caption*/}},	04LL63617074696F6E......
text-unit{	A3LL
content-portion-attributes{	3109
content-identifier-logical "3 1 3 0"},	840733203120332030
content-information{/*Unformatted string of	
C's*/}},	04LL4343434343.......
text-unit{	A3LL
content-portion-attributes{	3109
content-identifier-logical "3 1 4 0"},	840733203120342030
content-information{/*Unformatted string of	
D's*/}},	04LL44444444.........
text-unit{	A3LL
content-portion-attributes{	310B
content-identifier-logical "3 1 6 0 0",	4109332031203620302030
raster-gr-coding-attributes{	A104
number-of-pels-per-line 1117}},	8002045D
content-information{"/*Array of	
raster-graphics content elements	04LLZZZZZZZZZZZZ.....
for the signature*/}},	....
text-unit{	A3LL
content-portion-attributes{	310B
content-identifier-logical "3 1 6 1 0"},	4109332031203620312030
content-information{"Miss Aude HEA Document	04LL4D697373............
Architect"}},	....

D.4   *Example 4:*   *Sample document from Annex B of Recommendation T.412; specific layout structure only*

document-profile{	A030
specific-layout-structure{"1"}	810131
document-characteristics {	A22B
document-architecture-class formatted,	810100
content-architecture-classes{	A512
{2 8 2 6 0},	060458020600
{2 8 2 7 0},	060458020700
{2 8 2 8 0}},	060458020800
interchange-format-class if-b,	860101
oda-version{"ISO 8613 : 1988"}}},	880F49534F..31393838
layout-object{	A21E
object-type document-layout-root,	020100
descriptor-body{	3119
object-identifier "1",	4101331
object-class "0",	820130
user-visible-name "Letter",	8E064C6574746572
subordinates{"0","1","2"}}},	A0091201301201311120132
layout-object{	A228
object-type page,	020102
descriptor-body{	3123

```
        object-identifier "1 0",                              4103312030
        object-class "0 0",                                   8203302030
        user-visible-name "Header",                           8E06486561646572
        subordinates{"0","1","2","3","4"}}},                  A00F1201301201311201321201133
                                                              120134
layout-object{                                                A218
        object-type frame,                                    020103
        descriptor-body{                                      3113
            object-identifier "1 0 0",                        41053120302030
            object-class "0 0 0",                             82053020302030
            subordinates{"0"}}},                              A003120130
layout-object{                                                A229
        object-type block,                                    020104
        descriptor-body{                                      3124
            object-identifier "1 0 0 0",                      410531203020302030
            object-class "0 0 0 0",                           820730203020302030
            user-visible-name "Logo",                         8E044C6F676F
            presentation-attributes{                          A60C
                content-architecture-class                    060458020700
                    {2 8 2 7 0}},
                raster-graphics-attributes{                   A104
                    pel-transmission-density 5}}}},           820105
layout-object{                                                A21E
        object-type frame,                                    020103
        descriptor-body{                                      3119
            object-identifier "1 0 1",                        410531203020331
            object-class "0 0 1",                             820530203020331
            user-visible-name "Date",                         8E0444617465
            subordinates{"0"}}},                              A003120130
layout-object{                                                A22F
        object-type block,                                    020104
        descriptor-body{                                      312A
            object-identifier "1 0 1 0",                      410731203020312030
            position{                                         A308
                horizontal 395,                               8002018B
                vertical 710,                                 800202C6
            dimensions{                                       A408
                horizontal 3060,                              80020BF4
                vertical fixed 540},                          8002021C
            content-portions{"0"},                            A103180130
            presentation-attributes{                          A606
                content-architecture-class                    060458020602
                {2 8 2 6 2}}}}},
layout-object{                                                A223
        object-type frame,                                    020103
        descriptor-body{                                      311E
            object-identifier "1 0 2",                        41053120302032
            object-class "0 0 2",                             82053020302032
            user-visible-name "Addressee",                    8E0941646472657373656565
            subordinates{"0"}}},                              A003120130
layout-object{                                                A225
        object-type block,                                    020104
        descriptor-body{                                      3120
            object-identifier "1 0 2 0",                      410731203020322030
            dimensions{                                       A408
                horizontal 4505,                              80021199
                vertical fixed 540},                          8002021C
            content-portions{"0"},                            A103180130
            presentation-attributes{                          A606
                content-architecture-class                    060458020602
                {2 8 2 6 2}}}}},
layout-object{                                                A221
        object-type frame,                                    020103
        descriptor-body{                                      311C
```

```
        object-identifier "1 0 3",                        41053120302033
        object-class "0 0 3",                             82053020302033
        user-visible-name "Subject",                      8E075375626A656374
        subordinates{"0"}}},                              A003120130
layout-object{                                            A22A
        object-type block,                                020104
        descriptor-body{                                  3125
                object-identifier "1 0 3 0",              410731203020332030
                dimensions{                               A408
                        horizontal 7200,                  80021C20
                        vertical fixed 905},              80020389
                presentation-style "5 0",                 9103352030
                content-portions{"0"},                    A103120130
                presentation-attributes{                  A606
                        content-architecture-class        060458020602
                        {2 8 2 6 2}}}},
layout-object{                                            A221
        object-type frame,                                020103
        descriptor-body{                                  311C
                object-identifier "1 0 4",                41053120302034
                object-class "0 0 4",                     82053020302034
                user-visible-name "Summary",              8E0753756D6D617279
                subordinates{"0"},                        A003120130
layout-object{                                            A22A
        object-type block,                                020104
        descriptor-body{                                  3125
                object-identifier "1 0 4 0",              410731203020342030
                dimensions{                               A408
                        horizontal 5585,                  800215D1
                        vertical fixed 2325},             80020915
                presentation-style "5 1",                 91033520331
                content-portions{"0"},                    A103120130
                presentation-attributes{                  A606
                        content-architecture-class        060458020602
                        {2 8 2 6 2}}}},
layout-object{                                            A21A
        object-type page,                                 020102
        descriptor-body{                                  3115
                object-identifier "1 1",                  41033120331
                object-class "0 1",                       82033020331
                user-visible-name "Body",                 8E04426F6479
                subordinates{"0"},                        A00310130
layout-object{                                            A22A
        object-type frame,                                020103
        descriptor-body{                                  3125
                object-identifier "1 1 0",                41053120312030
                object-class "0 1 0",                     82053020312030
                user-visible-name "Body",                 8E04426F6479
                subordinates{"0","1","2","3","4"},        A00F120130120131120132120
                                                          33120134
layout-object{                                            A234
        object-type block,                                020104
        descriptor-body{                                  312F
                object-identifier "1 1 0 0",              410731203120302030
                position{                                 A308
                        horizontal 540,                   8002021C
                        vertical 540,                     8002021C
                dimensions{                               A408
                        horizontal 7935,                  80021EFF
                        vertical fixed 1785},             800206F9
                presentation-style "5 3",                 9103352033
                content-portions{"0"},                    A103120130
                presentation-attributes{                  A606
                        content-architecture-class        060458020602
                        {2 8 2 6 2}}}},
```

```
layout-object{                                          A238
        object-type block,                              020104
        descriptor-body{                                3133
                object-identifier "1 1 0 1",            4107312031203020331
                position{                               A308
                        horizontal 540,                 8002021C
                        vertical 3205,                  8020C83
                dimensions{                             A408
                        horizontal 7935,                80021EFF
                        vertical fixed 1785},           800206F9
                presentation-style "5 3",               9103352033
                content-portions{"0"},                  A103120130
                presentation-attributes{                A60A
                        character-attributes{
                                line-spacing, 400},     87020190
                        content-architecture-class      060458020602
                        {2 8 2 6 2}}}},
layout-object{                                          A22D
        object-type block,                              020104
        descriptor-body{                                3128
                object-identifier "1 1 0 2",            410531203120302032
                position{                               A308
                        horizontal 1615,                8002064F
                        vertical 6460}                  8002193C
                dimensions{                             A408
                        horizontal 5045,                800213B5
                        vertical fixed 4140},           8002102C
                presentation-attributes{                A606
                        content-architecture-class      060458020800
                        {2 8 2 8 0}},
                content-portions{"0"}}},                A103120130
layout-object{                                          A22D
        object-type block,                              020104
        descriptor-body{                                3128
                object-identifier "1 1 0 3",            410531203120302033
                position{                               A308
                        horizontal 1985,                800207C1
                        vertical 10235},                800227FB
                dimensions{                             A408
                        horizontal 3970,                80020782
                        vertical fixed 370},            80020172
                content-portions{"0"},                  A103120130
                presentation-attributes{                A606
                        content-architecture-class      060458020602
                        {2 8 2 6 2}}}},
layout-object{                                          A232
        object-type block,                              020104
        descriptor-body{                                312D
                object-identifier "1 1 0 4",            410531203120302034
                position{                               A308
                        horizontal 540,                 8002021C
                        vertical 11485},                80022CDD
                dimensions{                             A408
                        horizontal 7935,                80021EFF
                        vertical fixed 1075},           80020433
                presentation-style "5 3",               9103352033
                content-portions{"0"},                  A103120130
                presentation-attributes{                A606
                        content-architecture-class      060458020602
                        {2 8 2 6 2}}}},
layout-object{                                          A21A
        object-type page,                               020102
        descriptor-body{                                3115
```

```
            object-identifier "1 2",                    4103312032
            object-class "0 1",                          82033020331
            user-visible-name "Body",                    8E04426F6479
            subordinates{"0"}}},                         A003120130
layout-object{                                           A22A
      object-type frame,                                 020103
      descriptor-body{                                   3125
            object-identifier "1 2 0",                   41053120322030
            object-class "0 1 0",                        82053020312030
            user-visible-name "Body",                    8E04426F6479
            subordinates{"0","1","2","3","4"}}},         A00F120130120131120132120133
                                                         120134
layout-object{                                           A234
      object-type block,                                 020104
      descriptor-body{                                   312F
            object-identifier "1 2 0 0",                 410731203220302030
            position{                                    A308
                  horizontal 540,                        8002021C
                  vertical 540,                          8002021C
            dimensions{                                  A408
                  horizontal 7935,                       80021EFF
                  vertical fixed 1275},                  800206F9
            presentation-style "5 3",                    9103352033
            content-portions{"0"},                       A103120130
            presentation-attributes{                     A606
                  content-architecture-class             060458020602
                  {2 8 2 6 2}}}},
layout-object{                                           A234
      object-type block,                                 020104
      descriptor-body{                                   312F
            object-identifier "1 2 0 1",                 410731203220303020331
            position{                                    A308
                  horizontal 540,                        8002021C
                  vertical 2695,                          80020A87
            dimensions{                                  A408
                  horizontal 7935,                       80021EFF
                  vertical fixed 1615},                  8002064F
            presentation-style "5 3",                    9103352033
            content-portions{"0"},                       A103120130
            presentation-attributes{                     A606
                  content-architecture-class             060458020602
                  {2 8 2 6 2}}}},
layout-object{                                           A234
      object-type block,                                 020104
      descriptor-body{                                   312F
            object-identifier "1 2 0 2",                 410731203220302032
            position{                                    A308
                  horizontal 1820,                       8002071C
                  vertical 5190,                          80021446
            dimensions{                                  A408
                  horizontal 6860,                       80021ACC
                  vertical fixed 2155},                  8002086B
            presentation-style "5 4",                    9103352034
            content-portions{"0"},                       A103120130
            presentation-attributes{                     A606
                  content-architecture-class             060458020602
                  {2 8 2 6 2}}}},
layout-object{                                           A22F
      object-type block,                                 020104
      descriptor-body{                                   312A
            object-identifier "1 2 0 3",                 410731203220302033
            position{                                    A308
                  horizontal 2695,                       80020A87
```

```
                    vertical 8110,
                dimensions{
                        horizontal 5585,
                        vertical fixed 2495},
                presentation-attributes{
                        content-architecture-class
                                {2 8 2 7 0},
                content-portions{"0"},
layout-object{
        object-type block,
        descriptor-body{
                object-identifier "1 2 0 4",
                position{
                        horizontal 5385,
                        vertical 10605,
                        dimensions{
                                horizontal 2520,
                                vertical fixed 905},
presentation-style "5 0",
        content-portions{"0"},
        presentation-attributes{
                content-architecture-class
                        {2 8 2 6 2}}}},
text-unit{
        content-portion-attributes{
                content-identifier-layout "1 0 1 0 0",
                content-identifier-logical "3 0 0 0"},
        content-information{"CESSON, 26 JUNE 1985"}},

text-unit{
        content-portion-attributes{
                content-identifier-layout "1 0 2 0 0",
                content-identifier-logical "3 0 1 0"},
        content-information{"To members of ISO/
                        TC97/SC18/WG3"}},
text-unit{
        content-portion-attributes{
                content-identifier-layout "1 0 3 0 0",
                content-identifier-logical "3 0 2 0"},
        content-information{"SUBJECT: PROPOSED
                        EXAMPLE TO CLARIFY
                        THE DOCUMENT\n
                        ARCHITECTURE MODEL"}},
text-unit{
        content-portion-attributes{
                content-identifier-layout "1 0 4 0 0",
                content-identifier-logical "3 0 3 0 0"},
        content-information{/*Formatted string of
                        SUMMARY-*/}},
text-unit{
        content-portion-attributes{
                content-identifier-layout "1 1 0 0 0",
                content-identifier-logical "3 1 0 0"},
        content-information{/*Formatted string of
                                A's*/}},
text-unit{
        content-portion-attributes{
                content-identifier-layout "1 1 0 1 0",
                content-identifier-logical "3 1 1 0"},
        content-information{/*Formatted string of
                                B's*/}},
text-unit{
        content-portion-attributes{
                content-identifier-layout "1 1 0 2 0",
```

80021FAE
A408
800215D1
800209BF
A606
060458020700

A103120130
A234
020104
312F
410731203220302034
A308
80021509
8002296F
A408
800209DA
80020389
9103352030
A103120130
A606
060458020602

A32E
312C
400B400931203020312030 302030
840733203020302030
0414434553534F4E2C..
..31393835
A339
3116
400B400931203020322030 302030
840733203020312030
041F546F206D656D62657273..
..574733
A3LL
3116
400B400931203020332030 302030
840733203020322030
04LL5456424A4543553A..
..4D4F44454C

A3LL
3118
400B4009312030203420302030
840933203020333020302030
04LL53554D4D415259.....

A3LL
3116
400B400931203120302030 302030
840733203120302030
04LL414141.............

A3LL
3116
400B400931203120302031 2030
840733203120312030
04LL424242.............

A3LL
3118
400B4009312031203020322030

```
        content-identifier-logical "3 1 2 0 0"},          840933203120322030302030
    content-information{/*Ordered set of                  04LL........
                    geometric-graphics
                    content elements
                    for the diagram*/}},
text-unit{                                                A3LL
    content-portion-attributes{                           3118
        content-identifier-layout "1 1 0 3 0",            400B4009312031203020332030
        content-identifier-logical "3 1 2 1 0"},          840933203120322030312030
    content-information{/*Formatted string for             04LL63617074696F6E..
                    the caption*/}},
text-unit{                                                A3LL
    content-portion-attributes{                           3116
        content-identifier-layout "1 1 0 4 0",            400B4009312031203020342030
        content-identifier-logical "3 1 3 0"},            840733203120332030
    content-information{/*Formatted string                04LL434343........
                    of C's*/}},
text-unit{                                                A3LL
    content-portion-attributes{                           3116
        content-identifier-layout "1 2 0 0 0" ,           400B4009312032203020302030
        content-identifier-logical "3 1 3 1"} ,           840733203120333 0331
    content-information{/*Formatted string                04LL434343........
                    of C's*/}},
text-unit{                                                A3LL
    content-portion-attributes{                           3116
        content-identifier-layout "1 2 0 1 0",            400B4009312032203020312030
        content-identifier-logical "3 1 4 0"},            840733203120342030
    content-information{/*Formatted string                04LL444444........
                    of D's*/}},
text-unit{                                                A3LL
    content-portion-attributes{                           310D
        content-identifier-layout "1 2 0 2 0"},           400B4009312032203020322030
    content-information{/*Formatted string                04LL454E44494E47........
                    of ENDING*/}},
text-unit{                                                A3LL
    content-portion-attributes{                           311E
        content-identifier-layout "1 2 0 3 0",            400B4009312032203020332030
        content-identifier-logical "3 1 6 0 0"},          840933203120362030302030
        raster-gr-coding-attributes{                      A204
            number-of-pels-per-line 1117}},              8002045D
    content-information{/*Array of                        04LL........
                    raster-graphics
                    content elements for
                    the signature*/}},
text-unit{                                                A3LL
    content-portion-attributes{                           3118
        content-identifier-layout "1 2 0 4 0",            400B4009312032203020342030
        content-identifier-logical "3 1 6 1 0"},          840933203120362030312030
content-information{"Miss Aude HEA\n                       04LL4D697373....
Document Architect"}},
```

D.5   *Example 5:*   *Sample document profile from Annex C of Recommendation T.414; document profile only*

```
document-profile {                                        A082LLLL
    generic-layout-structure "0",                         800130
    generic-layout-structure "1",                         810131
    generic-layout-structure "1",                         850131
    resource-document                                     AA41
```

```
        descriptive-reference "Finance Master",          433F46696E616E636520
        Widget Inc, 4511 McKenzie,                       4D61737465722C576964
        Atlanta, Georgia, USA",                          67657420496E632E2C34
                                                         353131204D634B656E7A
                                                         69652C41746C616E7461
                                                         2C2047656F726769612C
                                                         205553412E
document-characteristics {                               A2LL
        document-application-profile {        },         800103
        doc-app-profile-defaults {                       AA0F
                document-architecture-defaults {         A00D
                        page-dimensions {                A208
                                horizontal 10200,        800227D8
                                vertical 13200},         80023390
                        transparency opaque (1)},        830101
                document-architecture-class              810102
                        formatted-processable (2),_
                content-architecture-classes {           A50706058808020602
                        {        }},
                interchange-format-class if-a (0),       860100
                oda-version "ISO 8613 : 1987-12-15",     A816
                                                         430849534F2038363133
                                                         440A4408313938373132
                                                         3135
non-basic-doc-characteristics {                          A22C
        page-dimensions {                                A241
                {horizontal 10200,                       3008800227D8
                vertical 13200},                         80023390
                {horizontal 13200,                       300880023390
                vertical 10200}},                        800227D8
        medium types {                                   A80F
                {nominal-page-size {                     300D3008
                        horizontal 10200,                800227D8
                        vertical 13200},                 80023390
                side-of-sheet recto (1)}},               020101
        protections {                                    B703
                protected (1)}},                         020101
additional-doc-characteristics {                         A9LL
        unit-scaling {12, 10},                           A30602010C02010A
        fonts-list {                                     A2LL
                {font-identifier 0,                      020100
                font-reference {        }},
                                                         30LL..............
                {font-identifier 1,                      020101
                font-reference {        }}}}},
                                                         30LL..............
                                                         ......................
document-management-attributes {                         A382036B
        document-description {                           A7E9
                title "May finance report",              80124D61792066696E61
                                                         6E6365207265706F7274
                subject "May results",                   810B4D61792072657375
                                                         6C7473
                document-reference
                        descriptive-reference
                        "May financial prelim."          A51743154D6179206669
                                                         6E616E6369616C207072
                                                         656C696D2E
document-type "Report",                                  82065265706F7274
        abstract "The current figures show              83795468652063757272
                an improvement in return                 656E7420666967757265
                on assets but still show                 732073686F7720616E20
```

an undercapitalization of
production capacity.",

            keywords {
                "Finance", "Financial",
                "May", "Return on assets"}}

    dates-and-times{
        document-date-and-time "1988-06-05",
        creation-date-and-time
            "1988-06-05T16:29:57",
        local-filing-date-and-time
            {"1988-06-05T11:51:03"},

        expiry-date-and-time "1989",
        purge-date-and-time "1989-12-31",
        release-date-and-time "1988-06-05"},
originators {
        organizations {
            "Widget Inc., Finance and control"},

            preparers {
                {personal-name {
                    surname "Maltby",
                    givenname "Reginald",
                    initials "P.", ·
                    title "Accountant"}}},

    owners {,
        {organization
            "Widget Inc., 4511 McKenzie,
            Atlanta, Georgia, USA"}},

    authors,
        {organization
            "Dewey, Cheatam & Howe CPA"}}},

    other-use-information {
        copyright {
            copyright-information {"Widget Inc."},

            copyright-dates {"1988"}},
        [... as before ...]
        status "May financial report",

distribution-list {
        {personal-name {
            surname "Marks",
            initials "D",
            title "Accountant"}},

696D70726F76656D656E
7420696E207265747572
6E206F6E206173736574
732062757420737469 6C
6C2073686F7720616E20
756E64657263
61706974616C697A6174
696F6E206F662070726F
64756374696F6E206361
7061636974792E
A42A
430746696E616E6365
430846696E6369616C
43034D6179
431052657475726E206F
6E20617373657473
A048
80083139383830363035
810F3139383830363035
54313632393537
A211
440F3139383830363035
54313135313033
830431393839
85083139383931323331
86083139383830363035
A181A4
A022
43205769646765742049
6E632E2C2046696E616E
636520616E6420436F6E
74726F6C
A1253123
A021
80064D616C746279
8108526567696E616C64
820150
830A4163636F756E7461
6E74
A2353133
81315769646765742049
6E632E2C203435313120
41746C616E74612C4765
6F726769612C20555341
2E
A320311E
80054465776579
810E4368656174616D20
2620486F7765
8303435041
A281B4
A0193117
A00D430B576964676574
20496E632E
A106440431393838

81104D61792066696E61
6C207265606F7274
A36C
3118A016
80054D61726B73
820144
830A4163636F756E7461
6E74

```
    {personal-name {
        surname "Bucks",
        initials "B.",
        title "VP },
        organization "Finance"},
    {personal-name {
        surname "Pencil",
        givenname "James",
        initials "K.",
        organization "Audits"}},
    {personal-name {
        surname "Duck",
        initials "D.",
        title "Controller"}}},
additional-information},


external-references {
    references-to-other-documents {
        descriptive-reference
            "April finance report",

        descriptive-reference
            "May balance",
        descriptive-reference
            "May accounting prelim."},

        superceded-documents {
            descriptive-reference
                "May financial A"},
        local-file-references{
            {file-name "mayfin",

                location "financial-previous"},

            {file-name "mayfin",

                location "financial-current"}},
content-attributes {
    document-size 40447,
    number-of-pages 16,
    languages ("US English")},


    security information {
        authorization
            organization "Widget Inc., Finance",

    security-classification
        "Company Financial",
    access-rights "Finance Group"}}}
```

```
3119A00E
80054275636B73
820142
83025650
810746696E616E6365
311CA012
800650656E63696C
81054A616D6573
82014B
8106417564697473
3117A015
80044475636B
820144
830A436F6E74726F6C6C
A519
43175369676E6E61747572
6520726563656970707420
7265712764
A34F
A03A
4314417072696C206669
6E616E6365207265706F
7274
430B4D617920626616C61
6E6365
43154D6179206163636F
756E74696E6720707265
6C696D
A111
430F4D617920666696E61
6E6369616C2041
A43B
311C
80066D617966696E
811266696E616E636961
6C5F70726576696F7573
311B
80066D617966696E
811166696E616E636961
6C5F63757272656E74
A515
81029DFF
820110
840C
430A555320456E676C69
7368
A632

840C5769646765742049
6E632E2C
8111436F6D70616E7920
46696E616E6369616C
A20F
430D46696E616E636520
47726F7470
```

OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT – CHARACTER CONTENT ARCHITECTURES[1]

CONTENTS

---

[1]   This text is aligned with the final text of the corresponding International Standard ISO 8613-6.

1 Scope

1.1 The purpose of the T.410 Series 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 graphic elements and raster graphic 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.

1.2 The T.410 Series of Recommendations 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.

1.3   This Recommendation:

-    defines character content architectures that can be used in conjunction with the document architecture defined in Recommendation T.412;

-    defines the internal structure of content conforming to these character content architectures;

-    defines those aspects of positioning and imaging applicable to presentation of these character content architectures in a basic layout object;

-    defines the presentation attributes and control functions applicable to these character content architectures;

-    describes a content layout process which, together with the document layout process described in Recommendation T.412, determines the layout of character content in basic layout objects and the dimensions of these basic layout objects.

2   **References**

The following Recommendations and Standards contain provisions which, through reference in this text, constitute provisions of the T.410 Series of Recommendations. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on the T.410 Series of Recommendations are encouraged to investigate the possibility of applying the most recent editions of the Recommendations and Standards listed below. Members of CCITT, ISO and IEC maintain registers of currently valid Recommendations and Standards.

-    ISO 2022 (1986): Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques.

-    ISO 6429 (1983): Information processing - ISO 7-bit and 8-bit coded character sets - Additional control functions for character imaging devices.

-    ISO 6937 (1983): Information processing - Coded character sets for text communication.

-    Rec. T.50 (1988): International alphabet No. 5.

-    Rec. X.208 (1988): Specification of abstract syntax notation one (ASN.1).

-    Rec. T.61 (1988): Character repertoire and coded character sets for the international Teletex service.

-    ISO 9541-5[2]: Information processing - Font and character information interchange - Part 5: Font attributes and character model.

_____

[2]   To be published.

3   Definitions

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

4   General principles

4.1   *Classes*

Three classes of character content architecture are distinguished:

- a character content architecture for formatted content which allows for document content to be presented (e.g. printed or displayed) as intended by the originator. Formatted content can be used in any basic component;

- a character content architecture for processable content which allows for document content to be processed (e.g. edited or formatted). Processable content can be used in any basic logical component;

- a character content architecture for formatted processable content which allows for document content to be processed and also to be presented as intended by the originator. Formatted processable content can be used in any basic component.

4.2   *Content*

The content of a basic component that conforms to a character content architecture is a character string. This character string is formed by concatenating the character strings in the content portions of the basic component.

The content character string consists of a combination of graphic characters, control functions and space characters.

4.3   *Presentation attributes*

Presentation attributes are applicable to basic logical and layout components. They contain information that specifies the initial conditions relating to the layout, the imaging and the selection of graphic characters of the content of these basic components. Some of these conditions can be changed by control functions contained within the content.

Presentation attributes are classified as follows:

- logical presentation attributes which can be associated with processable and formatted processable character content. These attributes take effect during the content layout process but are ignored during the content imaging process;

- layout presentation attributes which can be associated with formatted and formatted processable character content. These attributes take effect during the content imaging process. They are generated either by a content layout process or by a process that creates or edits the formatted or formatted processable content;

- shared presentation attributes which can be associated with all character content architecture classes. These attributes take effect during either or both the content layout and imaging processes.

Note - Presentation attributes can be applied to the content of a basic component in one of three ways. They can be specified directly in an object or object class description or they can be specified directly by means of a presentation style associated with the object or object class description. Alternatively, they can be indirectly applied to an object or object class by means of a default value list (see Recommendation T.412).

4.4 Control functions

Control functions with zero or more parameters may specify information relating to the layout or imaging of subsequent graphic characters. A control function can also be used to extend or replace the set of graphic characters being used. The scope of all control functions is limited to the basic component in which they occur.

Classification of control functions is similar to that of presentation attributes:

- logical control functions which can be used in processable and formatted processable character content. These control functions take effect during the content layout process but are ignored during the content imaging process;

- layout control functions which can be used in formatted and formatted processable character content. These control functions take effect during the content imaging process. They are generated by the content layout process. Alternatively, they may be inserted by a process (not described in the T.410 Series of Recommendations) that creates of edits the formatted or formatted processable content;

- shared control functions which can be used in all character content architecture classes. These control functions take effect during either or both the content layout and imaging processes.

In addition, formatted processable content may contain control functions known as delimiters. These delimiters are used to indicate a string of one or more graphic characters and/or control functions that have been inserted as the result of a content layout process (see § 12). The delimited graphic characters and/or control functions take effect only during the content imaging process. The delimiters take effect during the content layout process by deleting them and the enclosed character sequence.

4.5 Graphic characters

The set of graphic characters used in the content of a basic component, and their coded representation, are specified by presentation attributes and code extension control functions (see §§ 10 and 11.1.17).

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

Any non-spacing characters included in a graphic character set are not to be used in isolation but only in combination with spacing characters.

4.6 Space characters

The character SPACE (SP) is considered both as a logical control function and as a graphic character. As a graphic character, it has a graphical representation consisting of the absence of a graphic symbol. As a control function, it indicates a potential line break point (see § 12.2.1.3.2).

Note - NBSP (No Break SPace) and any fixed-width space characters, such as "digit space", "em space" and "en space" are regarded as graphic characters i.e. are not regarded as line break points.

4.7 Coding of content information

The coded representation of the content information within a content portion is in accordance with the rules specified in ISO 2022.

Coded representations of control functions are defined in ISO 6429 and are summarized in Annex C.

4.8 Internal structure

4.8.1 Formatted content

Formatted content is content for which all the necessary information relating to the layout and imaging of that content has been specified. Content in this form is intended to be imaged as specified and is not intended to be revised by an editing process or to be reformatted.

The content of a basic component conforming to a formatted character content architecture consists of one or more lines of characters. Each pair of successive lines is separated by a hard line terminator. The last (or only) line may or may not be terminated by a hard line terminator; the end of the content of a basic component implicitly terminates the last line.

4.8.2 Processable content

Processable content is content which has not been laid out. Content in this form is suitable for revision by an editing process.

Note – The editing process is implementation dependent and is not described in the T.410 Series of Recommendations.

In order to image content in this form, it is necessary to apply a content layout process (see § 12) to the content which converts the processable content into formatted content (see § 4.8.1) or into formatted processable content (see § 4.8.3).

To assist the processing (i.e. editing or layout processes) of processable content, a number of logical presentation attributes and control functions have been defined (see §§ 7 and 11). In addition, the character SPACE is regarded as both a graphic character and as a control function that indicates where a line break may occur when the content is laid out.

The content of a basic component conforming to a processable character content architecture consists of one or more sequences of characters. Each pair of successive character sequences is separated by a hard line terminator control function. The last (or only) character sequence may or may not be terminated by a hard line terminator.

If the hard line terminator is omitted at the end of the content of a basic logical component to which another basic logical component is concatenated (see Recommendation T.412), then the last character sequence continues into the content of the next basic logical component. In all other cases, the end of the content of the basic logical component implicitly terminates the last character sequence.

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

4.8.3 Formatted processable content

Formatted processable content is content that is structured such that it contains both the formatted content and the processable content as subsets. It is identical in structure to the processable content, except that it may contain additional control functions and graphic characters that have been added as a result of the content layout process. It is identical in structure to the formatted content, except that it may contain logical control functions and delimiters.

Thus, formatted processable content can be converted to processable content by deleting (or ignoring) all layout control functions, all occurrences of the delimiters and all control functions and characters within those delimiters.

Alternatively, formatted processable content can be converted to formatted content by deleting (or ignoring) all logical control functions and the delimiters but retaining the control functions and characters within the delimiters.

Note – The conversion of formatted processable content to processable content is a reversible process (providing the same layout constraints are applicable to the content layout process) but converting formatted processable content to formatted content is irreversible.

The formatted view of a basic component conforming to a formatted processable character content architecture consists of one or more lines of characters. Each pair of successive lines is separated by either a hard or soft line terminator. The last (or only) line may or may not be terminated by a hard line terminator; the end of the content of a basic layout component implicitly terminates the last line.

The processable view of a basic component conforming to a formatted processable character content architecture consists of one or more sequences of characters. Each pair of successive character sequences is separated by a hard line terminator. The last (or only) character sequence may or may not be terminated by a hard line terminator.

If the hard line terminator is omitted at the end of the content of a basic logical component to which another basic logical component is concatenated (see Recommendation T.412), then the last character sequence continues into the content of the next basic logical component. In all other cases, the end of the content of the basic logical component implicitly terminates the last character sequence.

Soft line terminators are used as separators between lines within a character sequence. The division into character sequences represents the internal structure of the content of a basic logical component. Each character sequence is anonymous, in that no name or identifier is associated with it, and no relationship exists among character sequences except that of sequence.

5 Character positioning

This section specifies how characters are to be positioned within a basic layout object. The intention is to aid understanding of the presentation attributes and control functions that relate to character positioning.

This section provides for the positioning of any font that is defined in accordance with ISO 9541-5. This section also caters for the positioning of characters pertaining to different fonts within the same basic layout object.

5.1 *Basic concepts*

5.1.1 *Character fonts*

In the context of this part of ISO 8613, the term *graphic character* is used in its abstract sense; that is, this term refers to a number of a set of graphic symbols used for the representation of information. The term *character image* is then used to refer to the rendition of a graphic character on a presentation medium.

A *font* is a set of character images, normally with a common design and size. A set of font attributes is associated with the font as a whole and a set of character attributes is associated with each individual character. These attributes are defined in ISO 9541-5.

The main purpose of the font attributes is for the recipient to identify the font used by the originator and, in case the specified font is not available, the font and character attributes serve as a guidance for the recipient to find an appropriate substitute font among those available.

Further information concerning the designation and invocation of different fonts within a basic object is given in § 6.

5.1.2 *Directions*

In the context of this Recommendation, all directions are expressed as counter-clockwise angles of rotation (in degrees) relative to a specified reference direction (an example is given in Figure 1/T.416).

The *character path* is the direction of progression of successive character images within a line box (defined in § 5.1.7) and is expressed as a direction relative to the horizontal direction of the layout object (see Figure 4/T.416).

The *line progression* is the direction of progression of successive line boxes within the basic layout object and is expressed as a direction relative to the character path (see Figure 11/T.416).

The *character orientation* is the direction of the character baseline (defined in § 5.1.3) relative to the character path.

Only one value for the character path, line progression and character orientation may be specified for a basic component.
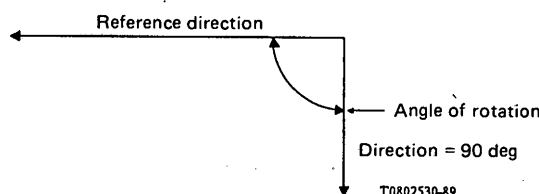
FIGURE 1/T.416

Example of direction

5.1.3 *Character image model*

The *position point* is a reference point associated with a character image (see Figure 2/T.416). It is used for the positioning of the character image within a line box. The *escapement point* is a reference point associated with a character image (see Figure 2/T.416). It is used for the positioning of the next character image.
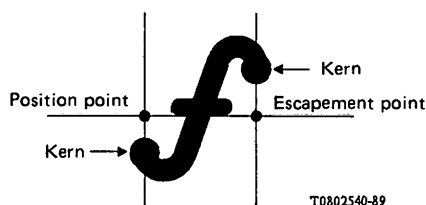
FIGURE 2/T.416

Illustration of kerns

The *character baseline* is an imaginary line across a character image, for the purpose of defining the character orientation. The character baseline is a horizontal lien with the character image is in its intended viewing orientation (see Figure 3/T.416).

Character path

Character baseline

PP   A   EP

Character orientation = 0 deg.

Character path

Character baseline

PP   ◁   EP

Character orientation = 90 deg.

PP   EP

Character baseline

Character orientation = 180 deg.

Character path

Character baseline

Character path

PP   ▷   EP

Character orientation = 270 deg.

T0802550-89

PP — Position point
EP — Escapement point
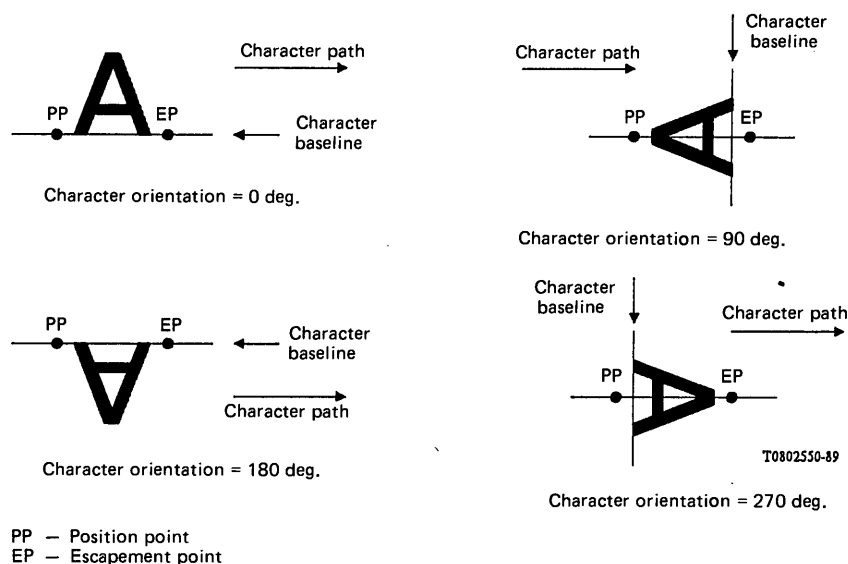
FIGURE 3/T.416

Reference points for character image positioning

A position point and escapement point must be defined for each character orientation which is intended to be used (see Figure 3/T.416); i.e., "writing modes" corresponding to the required character orientation must be defined in the font description, or fall-backs must be defined in document application profiles.

A *kern* is that part of the character image that extends beyond its position and escapement points (see Figure 2/T.416).

5.1.4   Character spacing

The concept of character spacing is only applicable when a constant spacing font is selected. It is used (in conjunction with the inter-character space) to determine the distance between character images within a line box as defined in § 5.2.1.

The character spacing is equal to the distance between the position points of successive character images when the inter-character space is zero.

The character spacing is independent of the distance between the position point and escapement point of character images.

5.1.5   Active position

The *active position* is an abstraction of an imaging device concept such as a *cursor*. This concept is used in the definitions of control functions (see § 11) where a sequential method of processing a character string is assumed.

The active position indicates the point, within the positioning area of a basic layout object, at which the action specified by the next character (graphic character or control function) is to be effected.

If the next character is a graphic character, its character image is positioned with the position point at the active position and the active position is advanced in the direction of the character path by the amount of spacing defined in § 5.2.1. If the next character is a control function, this may cause the active position to move to another point within the positioning area.

5.1.6   Positioning area

A *positioning area* is a rectangular area, wholly contained within a basic layout object, within which position points and escapement points are to be positioned (see Figure 4/T.416). Kerns of character images are permitted to extend beyond the positioning area but are not permitted to extend beyond the edges of the basic layout object.

Two of the edges of the positioning area are referred to as the *start edge* and *end edge* (see Figure 4/T.416). The start edge and the end edge are defined such that the direction from the start edge to the end edge is in the direction of the character path.

The other two edges of the positioning area are referred to as the *top edge* and *bottom edge* (see Figure 11/T.416). The top edge and the bottom edge are defined such that direction from the top edge to the bottom edge is in the direction of line progression.

The start and end edges are indented from the corresponding edges of the basic layout object by a distance referred to as the *kerning offset* (see Figure 4/T.416). The kerning offset specified depends upon the fonts of the characters to be positioned in the positioning area. Its value is chosen such that no part of any character image with kerns will extend beyond the boundary of the basic layout object when sequences of character images are positioned within it.
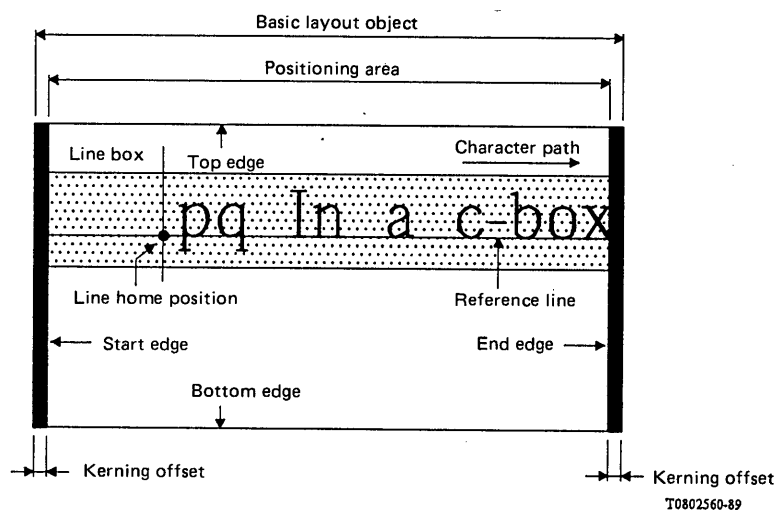
FIGURE 4/T.416

Illustration of character position concepts

5.1.7   *Line boxes*

Within the positioning area, a sequence of character images is positioned within an area, a sequence of character image is positioned within an area called a *line box* (see Figure 11/T.416). Each line box is a rectangular area that extends from the start edge to the end edge of the positioning area.

Each line box contains a reference point called the *line home position* (see Figure 4/T.416). This point is used for positioning the line box within the basic layout object. It also serves as the active position for the first graphic character or control function in each line.

Each line box contains an imaginary line called a *reference line* (see Figure 4/T.416). The reference line passes through the line home position in the direction of the character path. It extends from the start edge to the end edge within the line box and is used for the alignment of character images.

The length of the line box is equal to the distance between the start and end edges. The width (or height) of a line box is equal to the sum of the line box forward extent and the line box backward extent (see Figure 4/T.416). The line box backward extent is the distance between the reference line and the edge of the lien box in the direction opposite to the direction of line progression. The line box forward extent is the distance between the reference line and the edge of the line box in the direction of line progression.

The values of the forward and backward extents depend on the maximum extents (measured perpendicular to the reference line) of the character fonts used in the line box. Determination of the forward and backward extents take into account any displacements of character images perpendicular to the character path, e.g. for subscripts, superscripts and parallel annotation.

Note - As an example, for a Latin font with character path 0 degrees, line progression 270 degrees and when a single font is used in the line box, then the forward and backward extents are equal to the maximum right and maximum left extents for that font as defined in ISO 9541-5.

5.2 *Positioning of character images within a line box*

Successive character images are positioned within a line box in the direction of the character path.

The position points of the character images are lined on the reference line unless the characters are imaged as subscripts, superscripts or parallel annotation.

There are several factors which affect the positioning of character images along the reference line:

- spacing between characters;

- alignment;

- tabulation;

- character ordering;

- parallel annotation;

- subscript/superscript;

- pairwise kerning;

- first line offset;

- itemization.

5.2.1 *Spacing between characters*

The *inter-character space* is an additional amount of spacing between the position points of successive character images, in the direction of the character path (see § 5.1.4). A negative value indicates a reduction in the spacing between successive character images.

The distance between the position points of successive character images may be constant or variable depending upon the font as follows:

- For fonts with *constant spacing*, the distance between the position points of successive character images is independent of the characters and is the sum of the character spacing (as specified by presentation attributes and control functions) and the inter-character space (see Figure 5/T.416).

- For fonts with *variable spacing*, the distance between the position points of successive character images is dependent upon the character i.e. normally the distance between the position point and the escapement point of a character, and is the sum of the net escapement of the character (as specified by the font) and the inter-character space (see Figure 6/T.416).
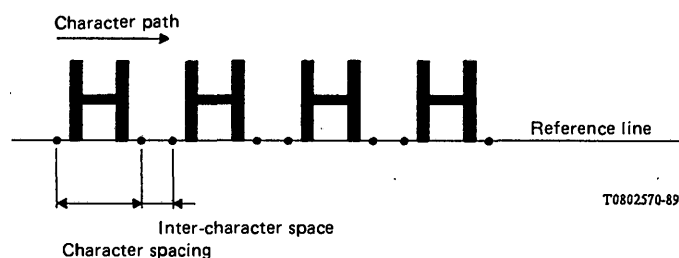
FIGURE 5/T.416

Spacing for a constant spacing font
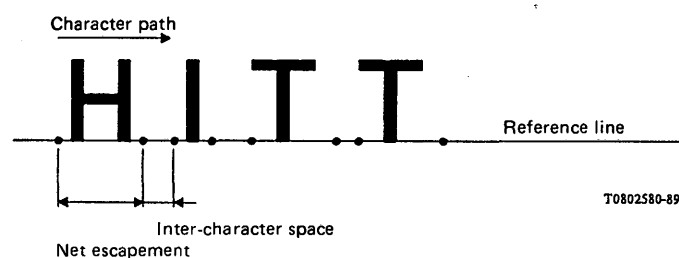
FIGURE 6/T.416

Spacing for a variable spacing font

The space width, i.e. the width of the SP (space) character image, is determined as follows:

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

- in a constant spacing font, the default width equals the character spacing;

- in a variable spacing font, the width is implicitly defined by the font;

- for all fonts, the width may be specified by a control function.

5.2.2 *Alignment*

The character images are positioned within a line box in accordance with the alignment attribute as follows:

- *start-aligned:* the position point of the first character image is placed at the line home position;

- *end-aligned:* the escapement point of the last character image is placed at the end edge of the positioning area;

- *centred:* the distance from the line home position to the position point of the first character image is approximately equal to the distance from the escapement point of the last character image to the end edge of the positioning area;

- *justified:* the position point of the first character image is placed at the line home position and the escapement point of the last character image is coincident with the end edge of the positioning area by appropriately setting the space width and/or the inter-character space.

5.2.3 *Tabulation*

The position of character images along a reference line can be controlled by means of a set of *tabulation stops*. Each tabulation stop specifies a point along a reference line relative to the start edge the positioning area.

A string of character images can be placed at a tabulation stop by means of a control function embedded in the text. The string may be start-aligned, end-aligned, centred or aligned around one or more specified characters within that string as follows (see Figure 7/T.416).

- *start-aligned:* the position point of the first character image of the string is placed at the tabulation stop;

- *end-aligned:* the escapement point of the last character image of the string is placed at the tabulation stop;

- *centred:* the string is placed such that the position point of the first character image and the escapement point of the last character image of the string are approximately equidistant from the tabulation stop;

- *aligned around:* the position point of the first character image of the first instance of the specified group of characters in that string is positioned at the tabulation stop. If the specified group or characters does not appear in the text associated with that tabulation stop, then the alignment defaults to end-aligned as defined above.

Tabulation stops

start-aligned	end-aligned	centred	aligned around '.'

This	This	This	124.56
is	is	is	23.5
start	end	centred	1254.76
aligned	aligned	text	32.465
text	text		342

T0802590-89
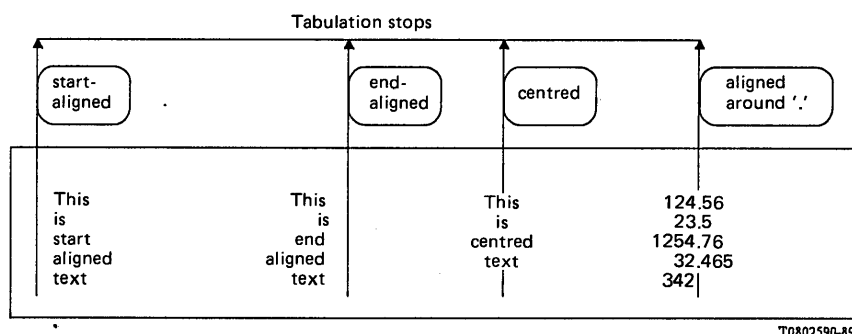
FIGURE 7/T.416

Tabulation

5.2.4 *Character ordering*

Within the content of a basic object, the interchange order is always the reading order of the language used.

In the case of certain languages, e.g. Arabic and Hebrew, where the alphanumeric text is read from right to left and the numeric text is read from left to right, the interchanged stream must indicate the change in presentation direction at the appropriate point(s).

This is necessary since control functions in character content architectures are defined to operate sequentially according to their position in the character stream.

When a string of characters with reversed presentation direction is embedded in the text with normal presentation direction, the image of the last character of the string with reversed presentation direction is positioned adjacent to the image of the last character of the preceding string with normal presentation direction (see Figure 8/T.416).

Note - In Figure 8/T.416 the terms "first" and "last" are used in relation to the interchange order and the terms "normal" and "reversed" in relation to the direction of the character path.

```
┌─────────────────────────────────────────┐
│ ░░░░░░░ Interchange order = ABCDE■123■FGH ░░░ │
└─────────────────────────────────────────┘
■  Control function to reverse presentation direction
P/D  Presentation direction
```

```
┌─────────────────────────────┐
│  HGF123EDCBA                │
└─────────────────────────────┘
              ▲ ▲  ▲  ▲
Character path │ │  │  └ Last character of normal P/D
◄─────────     │ │  └── Last character of reversed P/D
               │ └───── First character of reversed P/D
               └─────── First character of next normal P/D
                                        T0802600-89
```
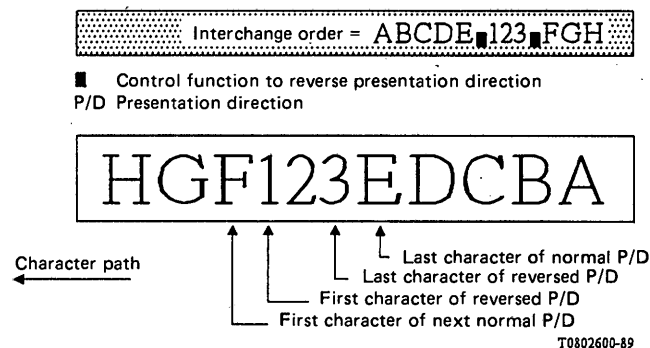
FIGURE 8/T.416

Character ordering

5.2.5 *Parallel annotation*

Two sequential character strings may be delimited such that the second string is used to indicate the pronunciation and interpretation of the first. It is intended to be used in the Japanese language to provide pronunciation and interpretation information (Ruby) in the form of *Kana character(s)* for one or more *Kanji characters*. This is indicated in the formatted text by the Kana character(s) being centered either above or to the right of the Kanji character(s), for character image orientations orthogonal and parallel to the character path respectively (see Figure 9/T.416).

Where centring would result in Kana characters being positioned outside the positioning area, then the Kana character string is positioned such that it is start-aligned or end-aligned with the edge of the available area.

5.2.6 *Subscript/superscript*

Subscript rendition allows for the active position to be displaced from the reference line in the direction of line progression.

Superscript rendition allows for the active position to be displaced from the reference line in the direction opposite to that of line progression.

The combined effect of all subscript/superscript renditions within a line box must be such that the active position is returned to the reference line before the occurrence of a hard or soft line terminator.

```
P       P       P
T 漢字 T カ ン ジ T        Processable
X       X       X          content
1       3       0
```

⇩ Content layout process

```
P       P S H   V   S       P S H     V   S    Formatted
T 漢字 T O P xx P yy T カ ン ジ T O P x'x' P yy T   processable
X       X S B   B           X S R     R        content
1       3                   0
```

⇩ Imaging process

Imaged content
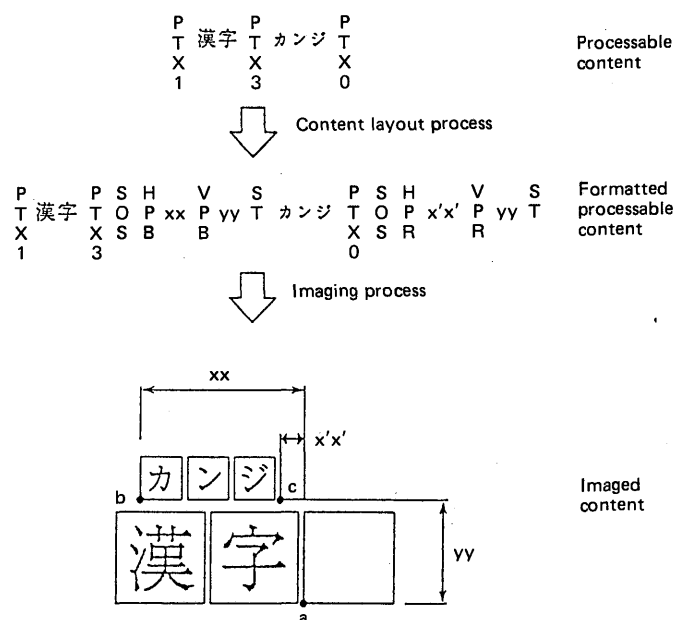
Where:

HPB	Character position backward;
HPR	Character position relative;
VPB	Line position backward;
VPR	Line position relative;
xx,x'x',yy	User specified.

The current reference point moves from "a", "b", "c" and returns to "a".

FIGURE 9/T.416

Illustration of parallel annotation

5.2.7 *Pairwise kerning*

Pairwise kerning allows for the moving of the active position from that defined by the preceding character. The distance and direction depend both on the character being imaged and the preceding character.

In the case of a constant spacing font, pairwise kerning is ignored.

In the case of a variable spacing font, the actual distance between the escapement point of one character image and the position point of the next character image is modified by the kerning information as defined in the character attributes of the font.

5.2.8 *First line offset*

First line offset allows for character imaging of the first line of a basic component to start at a position displaced from the line home position.

The offset is either in the direction of the character path (producing first line indentation) or in the direction opposite to the character path (producing overhang) as illustrated in Figure 10/T.416.
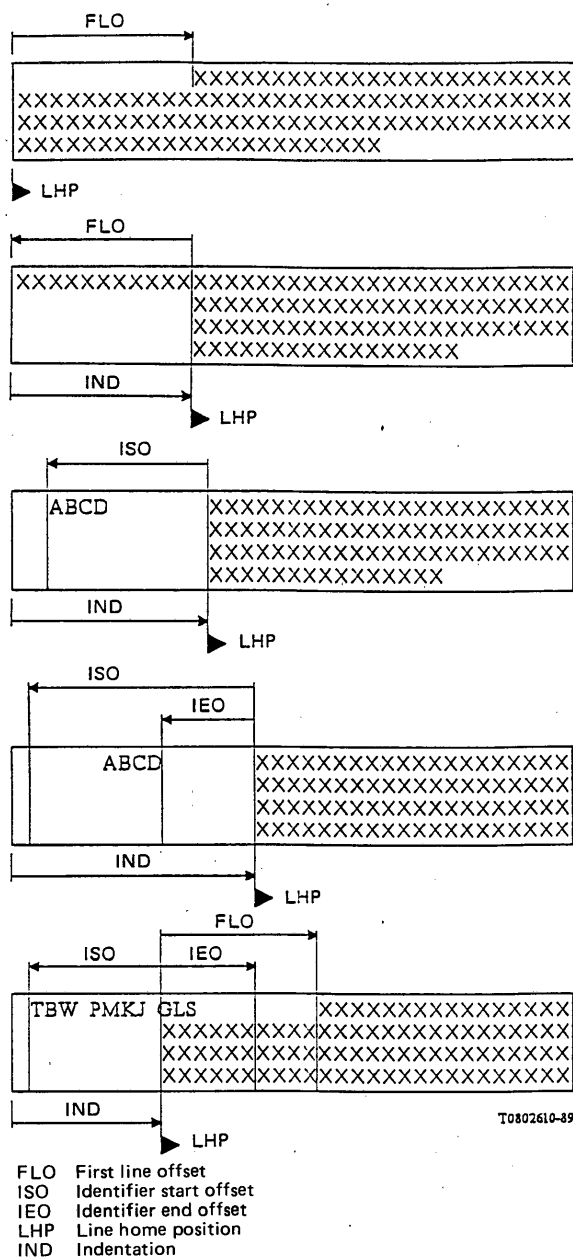
Example 10.1

first line offset:	positive
identifier alignment:	no itemisation
identifier start offset:	zero
identifier end offset:	zero
indentation:	zero

Example 10.2

first line offset:	negative
identifier alignment:	no itemisation
identifier start offset:	zero
identifier end offset:	zero
indentation:	positive

Example 10.3

first line offset:	zero
identifier alignment:	start aligned
identifier start offset:	negative
identifier end offset:	zero
indentation:	positive

Example 10.4

first line offset:	zero
identifier alignment:	end aligned
identifier start offset:	negative
identifier end offset:	negative
indentation:	positive

Example 10.5

first line offset:	positive
identifier alignment:	start aligned
identifier start offset:	negative
identifier end offset:	positive
indentation:	positive

T0802610-89

FLO First line offset
ISO Identifier start offset
IEO Identifier end offset
LHP Line home position
IND Indentation

FIGURE 10/T.416

Illustration of itemization and first line offset

5.2.9 Itemization

Itemization allows for imaging of an *item identifier* on the first line of a basic component in positions which are not constrained by the line home position and the first line offset (see Figure 10/T.416).

An item identifier is a string of characters that precedes and is separated from the remainder of the first line of a basic component.

A start offset and an end offset are defined relative to the line home position. These offsets determine the location of the item identifier and the separation between the item identifier and the line home position.

The item identifier may be positioned on the first line in accordance with the item identifier alignment attribute as follows:

-   *start-aligned:* the position point of the first character image of the item identifier is place at the start offset;

-   *end-aligned:* the escapement point of the last character image of the item identifier is placed at the end offset.

5.3   *Positioning of line boxes within a basic layout object*

For positioning of line boxes in a basic layout object, the area of that object is independent of any adjoining areas. No part of the image is permitted to extend beyond the boundaries of the basic layout object.

The *initial point* is the point relative to which all line boxes are positioned within the basic layout object (see § 7.2.2 and Figure 11/T.416).

The line home position of the first line box is at the initial point of the basic layout object. Subsequent line home positions are located on a line through the initial point in the direction of line progression.

The distance between the line home positions of two successive line boxes is determined as follows:

-   when proportional line spacing is to be performed, the distance between the reference lines of two successive line boxes is evaluated by an implementation dependent algorithm not defined in this Recommendation;

-   when proportional line spacing is not to be performed, the distance between the reference lines of two successive line boxes is equal to the current line spacing as specified by presentation attributes and control functions.
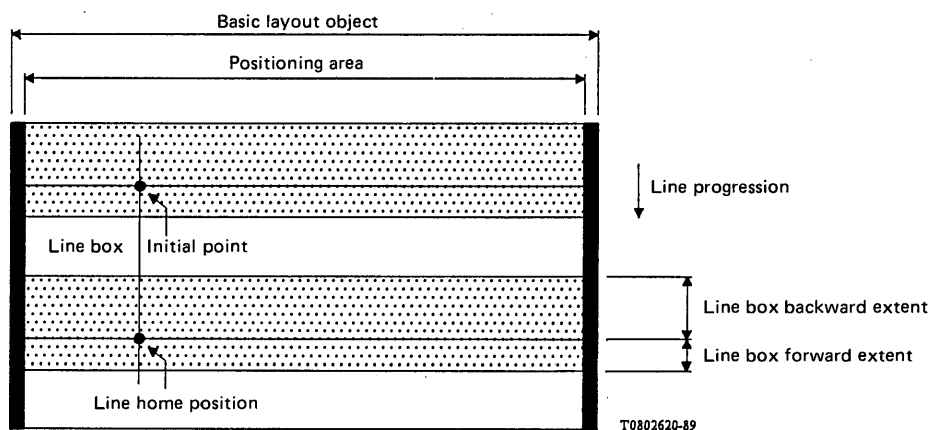
FIGURE 11/T.416

Illustration of line box positioning concepts

6   Character imaging

Four groups of specifications may apply to the imaging of graphic character elements in a basic object, namely those relating to:

-   emphasis;

-   font selection;

-   subscript and superscript;

-   character combinations.

These groups of specifications are defined below.

Note 1 - Emphasis (by "weight", "posture" etc.) and subscript/superscript rendition may be achieved by font selection.

Note 2 - Document application profiles may define additional restrictions on the use of the character features defined in this section. In addition, an implementation may replace an imaging feature by an alternative fall-back feature. However, this Recommendation does not define preferred fall-back features. It should also be noted that use of a fall-back feature may cause incorrect or misleading information to be conveyed to the user. It is particularly recommended that a fall-back is not used when the feature "crossed-out" is specified (see § 6.1.6).

6.1 *Emphasis*

Portions of text may be visually differentiated or emphasized in character imaging. Six methods of emphasis are provided:

- weight;

- posture;

- underlining;

- blinking;

- image inversion;

- crossing-out.

All of these may be controlled by means of rendition selection using the presentation attribute "graphic rendition" and the control function SGR (Select Graphic Rendition).

The emphasis takes effect at the active position of the line box where the control function initiates it and ends at the active position of the line box where the control function specifies an end to the emphasis or at the end of the object.

Note - Not all forms of emphasis allowed by this Recommendation are applicable to all presentation devices. For example, in printed text, 'blinking' may not be able to be represented, a second colour may be an available alternative to 'faint' (decreased intensity), but 'image inversion' may only be practicable by the use of appropriately designed cameo fonts.

6.1.1 *Weight*

This provides emphasis or de-emphasis by means of varying the contrast or intensity of the character image.

Using renditions selection, three weights are provided:

- faint (decreased intensity);

- normal intensity (neither faint nor bold);

- bold (increased intensity).

Only one of these may be in effect at any one point in the character stream, so that invoking one resets the others.

6.1.2 *Posture*

This causes a change of posture between an upright font and an italic font.

Using rendition selection, two renditions are provided:

- not italicized;

- italicized.

Only one of these renditions may be in effect at any one point in the character stream, so that invoking one resets the other.

6.1.3 *Underlining*

This provides for underlining character images. Three renditions are provided:

- not underlined;

- underlined;

- doubly underlined.

Only one of these renditions may in effect at any one point in the character stream, so that invoking one resets the others.

Note - For writing systems other than those that use a horizontal writing direction, underlining may be replaced by a suitable emphasis.

·6.1.4   *Blinking*

This provides for flashing of the graphic symbol in the line box ON and OFF. Three renditions are provided:

-    steady (not blinking);

-    slowly blinking;

-    rapidly blinking.

Only one of these renditions may by in effect at any one point in the character stream, so that invoking one resets the other.

Note - Slow blinking should be taken as less than 150 ON/OFF cycles per minute and rapid blinking should be taken as more than 150 per minute.

6.1.5   *Image inversion*

This causes the colours of the graphic symbol and the line box to be exchanged. Two renditions are provided:

-    positive image;

-    negative image.

Only one of these renditions may by in effect at any one point in the character stream, so that invoking one resets the other.

Note - Simple inversion of the pels within the line box area may not be sufficient for inverting the character image.

6.1.6   *Crossing-out*

This allows characters to be marked for deletion. The method of crossing out is not defined but the characters should be legible. Two renditions are provided:

-    not crossed-out;

-    crossed-out.

Only one of these renditions may by in effect at any one point in the character stream, so that invoking one resets the other.

6.2   *Font selection*

The capability for font selection is based upon:

-    font specification: the required fonts and their characteristics must be specified in the document profile (see Recommendation T.414). There is no limit to the number of fonts that can be specified;

-    font designation: a subset of these fonts can be designated for use within a basic component by means of the presentation attribute "character fonts". This subset is limited to a maximum of 10 fonts;

-    font invocation: any of the fonts within the subset can be invoked by means of the presentation attribute "graphic rendition" and/or control function SGR (Select Graphic Rendition).

6.2.1   *Font specification*

Within the document profile, there is an attribute "fonts list". This specifies, for each font used in the document:

-    a font identifier, represented by an integer, unique within the document;

-    a font description consisting of a set of font attributes specifying properties that are common to all characters of the font, e.g. weight, posture, constant or variable spacing, as well as properties that are unique to the individual characters.

The font description includes a structured font name as defined in ISO 9541. The other font attributes allow a recipient to select a suitable substitute font when the font specified by the structured font name is not available.

Note - It is necessary to consider that font designs are typically proprietary, and that, in addition, the recipient may have a limited repertoire. Thus, there can be no assurance that a particular font selected by the originator is actually available at the recipient's location, unless the originator has prior knowledge of the facilities available to the recipient. Since this would not generally be true in open interchange, it is necessary to specify the font and character characteristics in a manner that facilitates the selection of a near-equivalent alternative, from the repertoire of fonts available to the recipient.

6.2.2 *Font designation*

The presentation attribute "character fonts" enables a subset of up to 10 fonts from the set of fonts specified in the attribute "fonts list" in the document profile to be designated for use within the basic component.

Each font and its size is specified together with the method of invoking the font, i.e. as the primary or one of the alternative fonts.

If no font is designated for the primary font or for a particular alternative font, it is left to the recipient to select a suitable font for the primary or alternative font concerned.

Note - The selection of a substitution font may depend upon the characteristics and capabilities of the presentation device.

6.2.3 *Font invocation*

The control function SGR (Select Graphic Rendition) is used to invoke the primary font or one of the 9 alternative fonts. Such an invocation can be performed anywhere within the content of a basic component.

For the initial conditions at the beginning of the basic component, the font indicated by the presentation attribute "graphic rendition" is invoked. In the absence of this attribute, the primary font is assumed to be invoked. If no fonts have been specified, the font used is implementation dependent.

When a designated font is invoked, any specification of weight or posture by an SGR parameter value, or by the presentation attribute "graphic rendition", is ignored.

Invocation of a character set by way of code extension dose not imply invocation of a font for that character set.

6.3 *Subscript and superscript*

The control functions PLD (Partial Line Down) and PLU (Partial Line Up) provide for characters to be imaged as subscript and superscript respectively.

Finer control of subscript and superscript rendition is provided by the control functions VPB (Line Position Backward) and VPR (Line Position Relative).

6.4 *Character combinations*

Two or more characters can be imaged as if they were a single symbol.

The control function GCC (Graphic Character Composition) within the content identifies the string of characters to be imaged as a single symbol.

7 Definition of character presentation attributes

Presentation attributes are applicable to basic logical and layout components. They specify the initial conditions at the start of the rendition of the content of that basic component. The content architecture associated with each basic component can have the means for changing certain of these presentation attributes by means of control functions embedded in the content; such capabilities are indicated in the definition of each presentation attribute.

The following categories of presentation attributes are defined:

- shared attributes which are available to all character content architecture classes;

- layout attributes which are available to formatted and formatted processable form character content architecture classes;

- logical attributes which are available to processable and formatted processable form character content architecture classes.

These attributes are listed in Table 1/T.416.

For each presentation attribute, a default value is defined. This value is used in the defaulting mechanism as defined in Recommendation T.412.

In addition, values specific to character content architectures are defined for the content architecture class attributes. These attributes are defined in Recommendation T.412.

TABLE 1/T.416

Character presentation attributes

Shared attributes	Layout attributes
Alignment Character fonts Character orientation Character path Character spacing Code extension announcers First line offset Graphic character sets Graphic character subrepertoire Graphic rendition Itemization Kerning offset Line layout table Line progression Line spacing Pairwise kerning	Formatting indicator Initial offset
	Logical attributes
	Indentation Proportional line spacing Orphan size Widow size

7.1   *Shared presentation attributes*

7.1.1   **alignment**

CATEGORY: Shared

PERMISSIBLE   VALUES: 'start-aligned'
                                           'end-aligned'
                                           'centred'
                                           'justified'

DEFAULT   VALUE: 'start-aligned'

DEFINITION:

This attribute specifies the method of character alignment (see § 5.2.2).

REMARKS:

The character alignment cannot be altered within the content of a basic component. Application of the value 'justified' may be suppressed by an occurrence of the control function JFY (No Justify).

This attribute can be overridden by the presentation attribute "line layout table" when any tabulation stops are specified (see § 7.1.13).

7.1.2 character fonts

CATEGORY: Shared

STRUCTURE: 10 pairs of two parameters: font size
font identifier

PERMISSIBLE VALUES: font size: any positive integer
font identifier: any positive integer

DEFAULT VALUES: font size: {none is defined}
font identifier: {none is defined}

DEFINITION:

This attribute designates up to 10 fonts which may be used within the basic component (see §§ 5.1.2 and 6.2). These fonts are referred to as the primary font, the first alternative font, the second alternative font etc. The fonts designated must be chosen from the fonts listed in the document profile (see Recommendation T.414).

The parameter "font size" specifies the size of the font i.e. the height of the character image; its value is an integer representing the size of the font in SMUs.

The parameter "font identifier" is an integer equal to the font identifier associated with the font in the document profile attribute "fonts list".

REMARKS:

One of the designated fonts may be invoked at the start of the presentation of the content associated with a basic component by means of the presentation attribute "graphic rendition", otherwise the primary font is assumed to be invoked. Also, fonts may be invoked within the content by means of the control function SGR (Select Graphic Rendition).

7.1.3 character orientation

CATEGORY: Shared

PERMISSIBLE VALUES: 0, 90, 180, 270 degrees

DEFAULT VALUE: 0 degrees

DEFINITION:

This attribute specifies the character orientation (see § 5.1.3).

REMARKS:

The character orientation cannot be altered within the content of a basic component.

7.1.4 character path

CATEGORY: Shared

PERMISSIBLE VALUES: 0, 90, 180, 270 degrees

DEFAULT VALUE: 0 degrees

DEFINITION:

This attribute specifies the character path (see § 5.1.2).

REMARKS:

The character path cannot be altered within the content of a basic component. However, local changes of the relationship between the imaging order of the characters and the interchange order can be specified by the control function SRS (Start Reverse String).

7.1.5 character spacing

CATEGORY: Shared

PERMISSIBLE VALUES: Any positive integer

DEFAULT VALUE: - The equivalent of 120 BMUs

DEFINITION:

This attribute specifies the character spacing which applies at the beginning of the basic component (see § 5.1.4).

The value of this attribute is an integer specifying the distance in SMUs.

REMARKS:

The character spacing value is only used while a constant spacing font is in use; it has no effect while a variable spacing font is in use.

It can be altered within the content of a basic component by means of the control functions SHS (Select Character Spacing) or SCS (Set Character Spacing).

7.1.6 code extension announcers

CATEGORY:	Shared
PERMISSIBLE VALUES:	The value of this attribute consists of the string of escape sequences, in accordance with ISO 2022, to announce the use of code extension features.
DEFAULT VALUE:	The escape sequences announcing the use of the G0 and G2 sets, the G2 set being invoked in columns 10 to 15.

DEFINITION:

This attribute announces the code extension features used in the basic component.

REMARKS:

The set of code extension features announced by this attribute cannot be altered within the content of a basic component.

7.1.7 first line offset

CATEGORY:	Shared
PERMISSIBLE VALUES:	Any integer
DEFAULT VALUE:	0

DEFINITION:

This attribute specifies an offset along the character path from the line home position, measured in SMUs (see § 5.2.8). The offset shall be positive (in the direction of the character path), negative (in the direction opposite to the character path) or zero.

The position identifier by application of this offset to the line home position is used instead of the line home position for the purposes of formatting and imaging the first line of the basic layout object in which the content of the basic logical component is laid out.

REMARKS:

The value of the presentation attribute "indentation" (see § 7.3.1) must be set such that the the resulting displacement of the line home position from the start edge of the positioning area is sufficient to enable overhanging characters to be imaged within the positioning area.

The first line offset cannot be altered within the content of a basic component.

7.1.8 graphic character sets

CATEGORY:	Shared
PERMISSIBLE VALUES:	The value of this attribute consists of the string of escape sequences, in accordance with ISO 2022 and the register of ISO 2375, to designate one or more graphic character sets, and any locking shift functions needed to invoke these character sets.
DEFAULT VALUE:	The escape sequences and shift functions designating and invoking the primary character set of ISO 6937/2 as the G0 set and the supplementary character set of ISO 6937/2 as the G2 set in columns 10 to 15.

DEFINITION:

This attribute specifies the graphic character set(s) designated and/or invoked at the beginning of the basic component.

REMARKS:

Other graphic character sets can be designated and/or invoked within the content of a basic component by means of the appropriate code extension escape sequences and shift functions.

7.1.9 graphic character subrepertoire

CATEGORY:	Shared
PERMISSIBLE VALUES:	The value of this attribute is either 0 or the identifier of a subrepertoire assigned in the register of ISO 7350. The value 0 identifiers the full repertoire of the graphic character sets that are designated at the beginning of the basic component.
DEFAULT VALUE:	0

DEFINITION:

This attribute identifies the subrepertoire of the graphic character repertoire of ISO 6937 used at the beginning of the basic component.

REMARKS:

This attribute is only applicable if the graphic character sets of ISO 6937 are used.

Other graphic character subrepertoires can be invoked within the content of a basic component by means of the control function IGS (Identify Graphic Subrepertoire).

7.1.10 graphic rendition

CATEGORY:	Shared
PERMISSIBLE VALUE:	A sequence of one or more integers corresponding to parameter values of the control function SGR (Select Graphic Rendition).
DEFAULT VALUE:	0

DEFINITION:

This attribute specifies the rendition parameters for font, underlining, etc., which apply at the beginning of the basic component (see § 6.1).

REMARKS:

The graphic rendition can be altered within the content of a basic component by means of the control function SGR (Select Graphic Rendition).

Note – If more than one graphic rendition parameter is encoded, then it is the user's responsibility to ensure that they are consistent.

7.1.11 itemization

CATEGORY:	Shared
STRUCTURE:	Three parameters: - identifier alignment; - identifier start offset; - identifier end offset.
PERMISSIBLE VALUES:	Identifier alignment: - 'no itemization'; - 'start-aligned'; - 'end-aligned'.

Identifier start offset: any integer value
Identifier end offset: any integer value

DEFAULT VALUES: Identifier alignment: 'no itemization'
Identifier start offset: the distance from the line home position to the start edge of the positioning area

Identifier end offset: 0

DEFINITION:

This attribute specifies the placement of an item identifier which shall begin the basic component (see § 5.2.9).

If the value of the parameter ''identifier alignment'' is 'no itemization', then no item identifier is present.

For other values of the parameter "identifier alignment", the parameters "identifier start offset" and "identifier end offset" specify offsets, in SMUs, from the line home position along the character path which identify, respectively, the start edge and end edge of a portion of the line box in which the item identifier will be formatted. These offsets may be positive (in the direction of the character path), negative (in the direction opposite to the character path) or zero.

The item identifier consists of all graphic characters preceding the first occurrence of the control function CR (Carriage Return) in the basic component. The content of the basic component following that CR shall be formatted as specified by the presentation attribute "first line offset".

Values of the parameter ''identifier alignment'' other then 'no itemization' specify the method of character alignment for the item identifier.

REMARKS:

The value of the presentation attribute "indentation" (see § 7.3.1) must be set such that the resulting displacement of the line home position from the start edge of the positioning area is sufficient to enable the item identifier to be imaged within the positioning area.

Itemization cannot be altered within the content of a basic component.

7.1.12 **kerning offset**

CATEGORY: Shared

STRUCTURE: Two parameters: start edge offset, end edge offset

PERMISSIBLE VALUES: Start edge offset: any non-negative integer
End edge offset: any non-negative integer

DEFAULT VALUES: Start edge offset: 0
End edge offset: 0

DEFINITION:

This attribute specifies the kerning offset as a pair of integer values in SMUs (see § 5.1.6). The parameter "start edge offset" specifies in distance from the edge of the basic layout object to the start edge of the positioning area. The parameter "end edge offset" specifies the distance from the edge of the basic layout object to the end edge of the positioning area.

REMARKS:

The kerning offset cannot be altered within the content of a basic component.

7.1.13 **line layout table**

CATEGORY: Shared

STRUCTURE: Four parameters: tab reference
tab position
alignment
alignment string

PERMISSIBLE VALUES: Tab reference: a string of one to four decimal digits used as the reference parameter in the control function STAB (Selective Tabulation).

Tab position: any non-negative integer

Alignment: 'start-aligned'
                'end-aligned'
                'centred'
                'aligned-around'

Alignment string: graphic characters from the set of graphic elements specified by the presentation attributes "graphic character sets" and "graphic character subrepertoire".

DEFAULT VALUES: The default values are such that no tabulation stops are defined.

DEFINITION:

This attribute specifies the positions and types of a sequence of tabulation stops (see § 5.2.3). The value of the parameter "tab position" specifies the distance in SMUs, in the direction of the character path, from the start edge of the positioning area to the tabulation stop.

REMARKS:

When this presentation attribute specifies any tabulation stops, the presentation attribute ''alignment'' is assumed to have the value 'start-aligned' (see § 7.1.1).

The parameter ''alignment string'' is only present if the value of the parameter ''alignment'' is 'aligned-around'.

The tabulation stops cannot be altered within the content of a basic component.

7.1.14 line progression

CATEGORY: Shared

PERMISSIBLE VALUES: 90, 270 degrees

DEFAULT VALUE: 270 degrees

DEFINITION:

This attribute specifies the line progression (see § 5.1.2).

REMARKS:

The line progression cannot be altered within the content of a basic component.

7.1.15 line spacing

CATEGORY: Shared

PERMISSIBLE VALUES: Any positive integer

DEFAULT VALUE: The equivalent of 200 BMUs

DEFINITION:

This attribute only applies when proportional line spacing is not to be done by the content layout process. In this case, it specifies the line spacing which applies at the beginning of the basic component (see § 5.3).

The value of this attribute is an integer specifying the distance in SMUs.

REMARKS:

The line spacing can be altered within the content of a basic component by means of the control functions SVS (Select Line Spacing) or SLS (Set Line Spacing).

7.1.16 pairwise kerning

CATEGORY: Shared

PERMISSIBLE VALUES: 'Yes'
                              'No'

DEFAULT VALUE: 'No'

DEFINITION:

This attribute specifies whether pairwise kerning should be performed on the content during the formatting process (see § 5.2.7).

The value 'yes' indicates that the formatting process should perform, if possible, pairwise kerning to the content.

The value 'no' specifies that pairwise kerning should not be performed on the content.

REMARKS:

The specification of pairwise kerning cannot be altered within the content of a basic component.

7.2   *Layout presentation attributes*

7.2.1   **formatting indicator**

CATEGORY: Layout

PERMISSIBLE VALUES: 'Yes'
                              'No'

DEFAULT VALUE: 'No'

DEFINITION:

This attribute specifies whether the content of a basic component has been formatted by a content layout process or not.

The value 'yes' indicates that the content of the basic layout component concerned contains layout control functions representing the effects of any use of the control function STAB (Selective Tabulation) or of the presentation attributes ''alignment'', ''first line offset'', ''itemization'' and/or ''pairwise kerning'' (see § 12.2.1.3.1).

REMARKS:

The specification of alignment cannot be altered within the content of a basic component.

Note – In interchange, a recipient can take advantage of this attribute only if he has a character font that is similar to that of the sender, i.e. a font that has the same width for each character as the sender's font.

7.2.2   **initial offset**

CATEGORY: Layout

STRUCTURE: Two parameters: horizontal coordinate
                                                  vertical coordinate

PERMISSIBLE VALUES: Horizontal coordinate: any non-negative integer
                              Vertical coordinate: any non-negative integer

DEFAULT VALUES: The default values of this attribute depend on character path, line progression and line spacing as defined in Table 2/T.416.

Default values of the presentation attribute "initial offset"

Character path	Line progression	Horizontal coordinate	Vertical coordinate
0	270	0	S
	90	0	H-S
90	270	S	H
	90	W-S	H
180	270	W	H-S
	90	W	S
270	270	W-S	0
	90	S	0

W   Horizontal dimension of basic component
H   Vertical dimension of basic component
S   Backward extent of first line box

. DEFINITION:

This attribute specifies the position of the initial point (see § 5.3).

The values of the parameters "horizontal coordinate" and "vertical coordinate" specify the horizontal and vertical coordinates, in SMUs, of the initial point relative to the top left corner of the basic component. The horizontal coordinate is measured positively from the vertical axis to the right and the vertical coordinate is measured positively from the horizontal axis downwards.

REMARKS:

The position of the initial point cannot be altered within the content of a basic component.

7.3   *Logical presentation attributes*

7.3.1   **indentation**

CATEGORY: Logical

PERMISSIBLE VALUES: Any non-negative integer

DEFAULT VALUE: 0

DEFINITION:

This attribute specifies the distance, in the direction of character path from the start edge of the positioning area, to the initial point of the basic layout object in which the content of the basic logical component is laid out (see Figure 10/T.416). The distance is specified in SMUs.

REMARKS:

The indentation cannot be altered within the content of a basic component.

7.3.2   **orphan size**

CATEGORY: Logical

PERMISSIBLE VALUES: Any positive integer

DEFAULT VALUE: 1

DEFINITION:

This attribute only applies when the content layout process would result in the basic logical object being laid out in two or more basic layout objects; for example, at a page or frame boundary.

The value of this attribute specifies the minimum number of lines of content that must be placed in the first basic layout object.

If the number of lines remaining in the first basic layout object is less than the value of this attribute, all the content laid out in subsequent layout objects.

REMARKS:

The orphan size cannot be altered within the content of a basic component.

7.3.3 proportional line spacing

CATEGORY:	Logical
PERMISSIBLE VALUES:	'Yes' 'No'
DEFAULT VALUE:	'No'

DEFINITION:

This attribute specifies how the content layout process is to determine the distance between the reference lines of two successive line boxes.

If the value is 'yes', the line spacing is variable and the content layout process takes into account the forward extent of the first line box and the backward extent of the second line box.

If the value is 'no', the line spacing does not depend on the content of the line boxes but is determined from values set by the presentation attribute ''line spacing'' or the control functions SLS (Set Line Spacing) and SVS (Select Line Spacing).

7.3.4 widow size

CATEGORY:	Logical
PERMISSIBLE VALUES:	Any positive integer
DEFAULT VALUE:	1

DEFINITION:

This attribute only applies when the content layout process would result in the basic logical object being laid out in two or more basic layout objects; for example, at a page or frame boundary.

The value of this attribute specifies the minimum number of lines of content that must be placed in the last basic layout object.

If the number of lines in the last basic layout object is less than the value of this attribute, sufficient lines shall be moved from the end of the previous basic layout object to meet this requirement.

7.4 *Content architecture class attributes*

7.4.1 *Content architecture class*

The value of the attribute "content architecture class" of a basic component description that conforms to this Recommendation is an ASN.1 object identifier with one of the following values:

{ 2 8 2 6 0 }   for the formatted content architecture class;

{ 2 8 2 6 1 }   for the processable content architecture class;

{ 2 8 2 6 2 }   for the formatted processable content architecture class.

7.4.2 *Content type*

The content architecture class attribute "content type" cannot be used to specify any of the content architecture classes defined in this Recommendation.

7.5 *Interactions between presentation attributes and layout directives*

The attribute "concatenation" of the document architecture (see Recommendation T.412) is applicable to content portions belonging to the same character content architecture class. When concatenation is in effect, as a result of the layout directive "concatenation", then, for the following presentation attributes:

- alignment,

- character fonts,

- character orientation,

- character path,

- first line offset,

- indentation,

- itemization,

- kerning offset,

- line layout table,

- line progression,

- orphan size,

- pairwise kerning,

- widow size,

any value that is specified for the first component of the concatenated sequence applies to all components in the sequence. Thus, in the case of presentation attributes, the values specified for the first component override the values for the other components in the sequence.

8 Character content portion attributes

8.1 *Common coding attributes*

The value of the attribute "type of coding" of a content portion description that conforms to this Recommendation is an ASN.1 object identifier with the value { 2 8 3 6 0 }.

8.2 *Other coding attributes*

No other coding attributes are defined for content portions conforming to this Recommendation.

9 Formal definitions of character content architecture dependent data types

9.1 *Introduction*

This paragraph contains the formal definitions, in ASN.1 notation (defined in ISO 8824), of the data types that are character content architecture dependent.

These data types are:

- the data type to represent the character content architecture specific presentation attributes in basic components, presentation styles and default value lists;

- the data type to represent the character content architecture specific coding attributes in content portions;

- the data type to represent the non-basic values of the character content architecture presentation attributes and control function parameters in the document profile;

- the data type to represent the non-basic values of the character content architecture coding attributes in the document profile;

- the data type to represent the non-standard default values of the character content architecture presentation and coding attributes in the document profile.

9.2 *Representation of presentation attributes*

The data type "Character-Attributes" contains a set of subordinate data types that specify the character presentation attributes. Some of these subordinate data types are elementary but others are structured and are themselves made up of subordinate data types. The format of these data types is given below.

The subset of subordinate data types that may occur within a particular instance of the data type "Character-Attributes" depends upon the particular character content architecture level that is specified.

Character-Presentation-Attributes { 2 8 1 6 2 }

DEFINITIONS   ::= BEGIN

EXPORTS   Character-Attributes,
           One-Of-Four-Angles,
           One-Of-Two-Angles,
           Alignment,
           Graphic-Rendition,
           Kerning-Offset,
           Proportional-Line-Spacing,
           Pairwise-Kerning;

Character-Attributes   ::= SET {
    character-path   [0]   IMPLICIT One-Of-Four-Angles OPTIONAL,
    line-progression   [1]   IMPLICIT One-Of-Two-Angles OPTIONAL,
    character-orientation   [2]   IMPLICIT One-Of-Four-Angles OPTIONAL,
    initial-offset   [3]   IMPLICIT Measure-Pair OPTIONAL,
    character-spacing   [6]   IMPLICIT INTEGER OPTIONAL,
    line-spacing   [7]   IMPLICIT INTEGER OPTIONAL,
    alignment   [8]   IMPLICIT Alignment OPTIONAL,
    line-layout-table   [9]   IMPLICIT Layout-Table OPTIONAL,
    graphic-rendition   [10]   IMPLICIT Graphic-Rendition OPTIONAL,
    formatting-indicator   [11]   IMPLICIT Formatting-Indicator OPTIONAL,
    character-fonts   [12]   IMPLICIT Character-Fonts OPTIONAL,
    graphic-char-subrepertoire [13]   IMPLICIT INTEGER OPTIONAL,
    itemization   [14]   IMPLICIT Itemization OPTIONAL,
    widow-size   [15]   IMPLICIT INTEGER OPTIONAL,
    orphan-size   [16]   IMPLICIT INTEGER OPTIONAL,

graphic-character-sets	[17]	IMPLICIT OCTET STRING OPTIONAL,
indentation	[19]	IMPLICIT INTEGER OPTIONAL,
kerning-offset	[20]	IMPLICIT Kerning-Offset OPTIONAL,
proportional-line-spacing	[21]	IMPLICIT Proportional-Line-spacing OPTIONAL,
pairwise-kerning	[22]	IMPLICIT Pairwise-Kerning OPTIONAL,
first-line-offset	[23]	IMPLICIT INTEGER OPTIONAL,
coding extension announcers	[24]	IMPLICIT OCTET STRING OPTIONAL }

```
One-Of-Four-Angles    ::= INTEGER { d0      (0),
                                     d90     (1),
                                     d180    (2),
                                     d270    (3) }


One-Of-Two-Angles     ::= INTEGER { d90     (1),
                                     d270    (3) }


Measure-Pair          ::= SEQUENCE {
     horizontal                    [0]   IMPLICIT INTEGER,
     vertical                      [0]   IMPLICIT INTEGER }

Alignment             ::= INTEGER { start-aligned    (0),
                                     end-aligned      (1),
                                     centred          (2),
                                     justified        (3) }

Layout-Table          ::= SET OF Tabulation-Stop

Tabulation-Stop       ::= SET {
     tabulation-reference          [0]   IMPLICIT NumericString,
     tabulation-position           [1]   IMPLICIT INTEGER,
     alignment                     [2]   IMPLICIT INTEGER {
                                         start-aligned    (0),
                                         end-aligned      (1),
                                         centred          (2),
                                         aligned-around   (3) }
     alignment-character-string    [3]   IMPLICIT OCTET STRING }

                                         -- string of graphic characters from the set of
                                         -- graphic elements specified by the presentation
                                         -- attributes "graphic character sets" and
                                         -- "graphic character subrepertoire"

Graphic-Rendition     ::= SET OF Graphic-Rendition-Aspect

Character-Fonts       ::= SET {
     primary-font                  [0]   IMPLICIT Font-Type OPTIONAL,
     first-alternative-font        [1]   IMPLICIT Font-Type OPTIONAL,
     second-alternative-font       [2]   IMPLICIT Font-Type OPTIONAL,
     third-alternative-font        [3]   IMPLICIT Font-Type OPTIONAL,
     fourth-alternative-font       [4]   IMPLICIT Font-Type OPTIONAL,
     fifth-alternative-font        [5]   IMPLICIT Font-Type OPTIONAL,
     sixth-alternative-font        [6]   IMPLICIT Font-Type OPTIONAL,
     seventh-alternative-font      [7]   IMPLICIT Font-Type OPTIONAL,
     eighth-alternative-font       [8]   IMPLICIT Font-Type OPTIONAL,
     ninth-alternative-font        [9]   IMPLICIT Font-Type OPTIONAL }

Font-Type             ::= SET {
     font-size                     [0]   IMPLICIT INTEGER,
     font-identifier               [1]   IMPLICIT INTEGER }

Graphic-Rendition-Aspect ::= INTEGER {
                                 cancel                (0),
                                 increased-intensity   (1),
                                 decreased-intensity   (2),
                                 italicized            (3),
                                 underlined            (4),
```

slowly-blinking	(5),
rapidly-blinking	(6),
negative-image	(7),
crossed-out	(9),
primary-font	(10),
first-alternative-font	(11),
second-alternative-font	(12),
third-alternative-font	(13),
fourth-alternative-font	(14),
fifth-alternative-font	(15),
sixth-alternative-font	(16),
seventh-alternative-font	(17),
eighth-alternative-font	(18),
ninth-alternative-font	(19),
doubly-underlined	(21)
normal-intensity	(22),
not-italicized	(23),
not-underlined	(24),
steady	(25),
variable-spacing	(26),
positive-image	(27),
not-crossed-out	(29),
not-variable-spacing	(50) }

```
Formatting-Indicator    ::= INTEGER { no (0), yes (1) }

Itemization             ::= SET {
    identifier-alignment        [0]  IMPLICIT INTEGER {
                                     no-itemization  (0),
                                     start-aligned   (1),
                                     end-aligned     (2) }
    identifier-start-offset     [1]  IMPLICIT INTEGER OPTIONAL,
    identifier-end-offset       [2]  IMPLICIT INTEGER OPTIONAL }

Kerning-Offset          ::= SET {
    start-offset                [0]  IMPLICIT INTEGER,
    end-offset                  [1]  IMPLICIT INTEGER }

Proportional-Line-Spacing  ::= INTEGER { no (0), yes (1) }
Pairwise-Kerning           ::= INTEGER { no (0), yes (1) }

    END
```

9.3 *Representation of coding attributes*

```
Character-Coding-Attributes    ::= { 2 8 1 6 3 }

    DEFINITIONS               ::= BEGIN

        EXPORTS Character-Coding-Attributes;
            Character-Coding-Attributes  ::= SET { }

                        -- no character coding attributes are defined
                        -- in this part of ISO 8613

    END
```

9.4 *Representation of non-basic features and non-standard defaults*

```
Character-Profile-Attributes  { 2 8 1 6 4 }

    DEFINITIONS               ::= BEGIN

        EXPORTS Char-Presentation-Feature,
                Character-Coding-Attribute,
                Character-Content-Defaults;

        IMPORTS One-Of-Four-Angles,
                One-Of-Two-Angles,
              . Alignment,
                Graphic-Rendition,
```

```
                    Kerning-Offset,
                    Proportional-Line-Spacing,
                    Pairwise-Kerning;

            FROM Character-Presentation-Attributes;

                                -- see § 9.2

Char-Presentation-Feature    ::= CHOICE {
    character-path                  [0]   IMPLICIT One-Of-Four-Angles,
    line-progression                [1]   IMPLICIT One-Of-Two-Angles,
    character-orientation           [2]   IMPLICIT One-Of-Four-Angles,
    character-spacing               [6]   IMPLICIT INTEGER,
    line-spacing                    [7]   IMPLICIT INTEGER,
    alignment                       [8]   IMPLICIT Alignment,
    graphic-rendition               [10]  IMPLICIT Graphic-Rendition OPTIONAL,
    graphic-char-subrepertoire      [13]  IMPLICIT INTEGER,
    graphic-character-sets          [17]  IMPLICIT OCTET STRING }


Character-Coding-Attribute   ::= NULL


                            -- no character coding attributes are defined
                            -- in this Recommendation.


Character-Content-Defaults   ::= SET {
    character-path                  [0]   IMPLICIT One-Of-Four-Angles OPTIONAL,
    line-progression                [1]   IMPLICIT One-Of-Two-Angles OPTIONAL,
    character-orientation           [2]   IMPLICIT One-Of-Four-Angles OPTIONAL,
    character-spacing               [6]   IMPLICIT INTEGER OPTIONAL,
    line-spacing                    [7]   IMPLICIT INTEGER OPTIONAL,
    alignment                       [8]   IMPLICIT Alignment OPTIONAL,
    graphic-rendition               [10]  IMPLICIT Graphic-Rendition OPTIONAL,
    graphic-char-subrepertoire      [13]  IMPLICIT INTEGER OPTIONAL,
    widow-size                      [15]  IMPLICIT INTEGER OPTIONAL,
    orphan-size                     [16]  IMPLICIT INTEGER OPTIONAL,
    graphic-character-sets          [17]  IMPLICIT OCTET STRING OPTIONAL,
    indentation                     [19]  IMPLICIT INTEGER OPTIONAL,
    kerning-offset                  [20]  IMPLICIT Kerning-Offset OPTIONAL,
    proportional-line-spacing       [21]  IMPLICIT Proportional-Line-spacing OPTIONAL,
    pairwise-kerning                [22]  IMPLICIT Pairwise-Kerning OPTIONAL }

    END
```

10   Graphic characters

The set of graphic characters used in the content of a basic component, and their coded representations, are specified by the presentation attributes "graphic character sets" and "graphic character subrepertoire" and the associated control functions, viz. the code extension control functions and the control function IGS (Identify Graphic Subrepertoire) as described below:

- graphic character sets: this presentation attribute specifies the graphic character sets designated and/or invoked at the beginning of the basic component. The specification of the graphic character sets by this attribute implies the definition of a repertoire of graphic characters and of a unique coded representation for each character of that repertoire;

- graphic character subrepertoire: this presentation attribute can be used, when the graphic character sets designated are those of ISO 6937, to restrict the repertoire of graphic characters to a subset of the repertoire implied by the specification of the graphic character sets. Use of this attribute does not affect the coded representations of the graphic characters;

- code extension control functions: these control functions can be used within the content of a basic component to alter the designations and/or invocations of graphic character sets from those specified by the presentation attribute "graphic character sets";

- IGS (Identify Graphic Subrepertoire): this control function can be used within the content of a basic component, when the graphic character sets designated are those of ISO 6937, to alter the subrepertoire from that specified by the presentation attribute "graphic character subrepertoire".

When the presentation attributes "graphic character sets" and "graphic character subrepertoire" are not specified for a basic component, their values are determined using the defaulting mechanism defined in Recommendation T.412, as for any other presentation attribute.

11   Definition of control functions and the character SPACE

Control functions are classified in the following four categories:

- shared control functions, including code extension control functions, which are available to all classes of character content architecture;

- layout control functions which are available only to formatted form and formatted processable form character content architectures;

- logical control functions which are available only to processable form and formatted processable form character content architectures;

- delimiters which are used to delimit graphic characters and/or shared control functions introduced as a result of a formatting process. The delimiters are available only to formatted processable form character content architectures.

The control functions in these four categories are defined in §§ 11.1 to 11.4 respectively, § 11.5 defines the character SPACE. The allocation of control functions to categories is summarized in Table 3/T.416.

Control functions that have not been included explicitly in this section are:

- the sequence introducers ESC (Escape) and CSI (Control Sequence Introducer) that are permitted to be used in the character content architecture when required to represent a graphic character or a control function;

- line terminators:

a)   a soft line terminator is represented by the control function CR (Carriage Return) then LF (Line Feed), in that order, where the CR and LF are enclosed between a pair of delimiters SOS (Start Of String) and ST (String Terminator);

b)   a hard line terminator is represented by a control function LF (Line Feed) that immediately follows either a control function CR (Carriage Return) or another LF where the entire sequence of CR and LF(s) is NOT enclosed between a pair of delimiters SOS (Start Of String) and ST (String Terminator).

11.1   *Shared control functions*

11.1.1   *CR - Carriage return*

A control function which causes the active position to be moved to the line home position but not to be moved in the direction of lien progression.

Note - CR is used in conjunction with the control function LF (Line Feed) to move the active position to the line home position at the beginning of a new line of text. It is also used to move the active position to the line home position, for example, after an item identifier (see § 7.1.11). CR should not be used to cause character images to be superimposed.

TABLE 3/T.416

Control functions

```
Shared control functions

CR    Carriage return
GCC   Graphic character composition
IGS   Identify graphic subrepertoire
LF    Line feed
PLD   Partial line down
PLU   Partial line up
SCS   Set character spacing
SGR   Select graphic rendition
SHS   Select character spacing
SLS   Set line spacing
SRS   Start reverse string
STAB  Selective Tabulation
SUB   Substitute Character
SVS   Select line spacing
VPB   Line position backward
VPR   Line position relative
      Code extension control functions

Layout control functions

BS    Backspace
HPB   Character position backward
HPR   Character position relative
JFY   No justify
SACS  Set additional character separation
SRCS  Set reduced character separation
SSW   Set SPACE width

Logical control functions

BPH   Break permitted here
NBH   No break here
PTX   Parallel texts

Delimiters

SOS   Start of string
ST    String terminator
```

11.1.2 *GCC - Graphic character composition*

A control function with one parameter which specifies that two or more graphic characters are to be combined into one graphic symbol.

The value of the parameter is 0, 1 or 2.

GCC with the parameter value 0 indicates that the following two graphic characters are to be presented as a single symbol.

GCC with the parameter value 1 indicates the start and GCC with the parameter value 2 indicates the end of a string of graphic characters that are to be presented as a single symbol.

The default value of the parameter is 0.

11.1.3 *IGS - Identify graphic subrepertoire*

A control function with a selective parameter which is used to indicate that a subrepertoire of the graphic character repertoire of ISO 6937 is used in the text. All graphic character sets that

are used to represent the indicated graphic character subrepertoire must be explicitly or implicitly designated, but need not be invoked, prior to the occurrence of IGS.

The identification f the graphic character subrepertoire may be changed at any point within a document and becomes effective immediately. No graphic characters other than those of the specified subrepertoire shall be used in the text following the occurrence of IGS. The effect of a graphic character subrepertoire identification ceases upon the next occurrence of:

- another IGS;

- the end of the current basic object;

- the designation of any graphic character set.

The default value of the parameter is zero.

A non-zero parameter value is the identifier assigned to a subrepertoire if the repertoire of ISO 6937 in accordance with the registration procedure specified in ISO 7350.

The parameter value zero identifiers the entire repertoire of the currently designated graphic character sets.

In the absence of IGS, the subrepertoire identified by the presentation attribute "graphic character subrepertoire" applies or the entire repertoire of the currently designated graphic character sets otherwise.

Note – The use of IGS in document application profiles based on the T.410 Series is deprecated. IGS is included in this part of ISO 8613 only for compatibility with some existing applications such as those based upon Recommendation T.61 (1984).

11.1.4 *LF - Line feed*

A control function which causes the active position to be advanced in the direction of line progression but not to be moved in the direction of the character path. The amount of movement is that specified by the most recent occurrence either of the control functions SLS (Set Line Spacing) or SVS (Select Line Spacing) if any, or otherwise by the presentation attribute "line spacing".

LF is restricted to be used in the following cases:

- at the beginning of the content of a basic layout component;

- immediately following a control function CR (Carriage Return);

- immediately following another LF.

11.1.5 *PLD - Partial line down*

A control function which causes either the start of subscript rendition or the end of superscript rendition of graphic characters.

When superscript rendition is in effect, it is terminated by PLD; otherwise, subscript rendition is initiated by PLD.

Any occurrence of PLD to start subscript rendition shall be followed by a control function PLU (Partial Line Up) in the same line before another PLD or control function LF (Line Feed) is used.

PLD does not affect the position of any (graphic) lines used to implement the graphic rendition 'underlined', 'crossed-out' or 'doubly underlined' when such a graphic rendition is in effect prior to the occurrence of PLD.

Note 1 – The implementation of the subscript rendition initiated by PLD may be accomplished with special character fonts and/or movement of the active position not exceeding a half line space.

Note 2 – The graphic rendition 'underlined' may have been effected, prior to the occurrence of PLD, either by the control function SGR (Select Graphic Rendition) or by the non-spacing underline character (see ISO 6937).

11.1.6 *PLU - Partial line Up*

A control function which causes either the start of superscript rendition of graphic characters.

When subscript rendition is in effect, it is terminated by PLU; otherwise, superscript rendition is initiated by PLU.

Any occurrence of PLU to start superscript rendition shall be followed by a control function PLD (Partial Line Down) in the same line before another PLU or control function LF (Line Feed) is used.

PLU does not affect the position of any (graphic) lines used to implement the graphic rendition 'underlined', 'crossed-out' or 'doubly underlined' when such a graphic rendition is in effect prior to the occurrence of PLU.

Note 1 - The implementation of the subscript rendition initiated by PLU may be accomplished with special character fonts and/or movement of the active position not exceeding a half line space.

Note 2 - The graphic rendition 'underlined' may have been effected, prior to the occurrence of PLU, either by the control function SGR (Select Graphic Rendition) or by the non-spacing underline character (see ISO 6937).

11.1.7 *SCS - Set character spacing*

A control function with one numeric parameter which specifies the character spacing to be applied to constant spacing fonts in subsequent text.

The specified character spacing takes effect immediately and remains in effect until it is changed by a subsequent occurrence of either SCS or the control function SHS (Select Character Spacing) in the current basic component.

The character spacing is expressed as an integral multiple of SMUs.

The default value of the parameter is the equivalent of 120 BMUs.

11.1.8 *SGR - Select graphic rendition*

A control function with one or more selective parameters which specify one or more graphic rendition aspects for graphic characters and space characters in the subsequent text. The specified graphic rendition(s) takes effect immediately and remains in effect until a subsequent occurrence of SGR in the basic object.

The meaning of the parameter value is shown in Table 4/T.416.

The default value of the parameter is 0.

Any graphic rendition aspect specified by an occurrence of SGR, with the exceptions as noted in Notes 2 and 3 below, is combined with the graphic rendition aspects that are in effect prior to that occurrence of SGR as a result of either an earlier occurrence of SGR or the presentation attribute "graphic rendition" of the current basic object.

When SGR is used to start underlining (singly or doubly), or crossing out, within the scope of subscript or superscript image - see control functions PLD (Partial Line Down) and PLU (Partial Line Up), any lines used to implement such a graphic rendition are lowered or raised in order that the graphic rendition concerned applies to the subscript or superscript characters.

Note 1 - Several parameter values can be used in combination, in order to obtain, for example, underlined italics.

Note 2 - The default parameter value cannot be used in combination with any other parameter value.

Note 3 - When a designated font is invoked, any specification of weight or posture by an SGR parameter value is ignored. This applies to the parameter values 1, 2, 3, 22 and 23. These parameter values are enabled when a primary or alternative font is invoked for which no designation has been made.

Note 4 - The use of parameter values 26 and 50 in document application profiles based on the T.410 Series is deprecated. These values are included in this part of ISO 8613 only for compatibility with some existing applications such as those based upon Recommendation T.61 (1984).

TABLE 4/T.416

Parameter values for SGR

Value	Meaning
0	default rendition (implementation-defined); cancels the effect of any preceding occurrence of SGR and cancels the effect of the presentation attribute "graphic rendition"; invokes the primary font
1	bold or increased intensity
2	faint or decreased intensity
3	italicize
4	underlined
5	slowly blinking
6	rapidly blinking
7	negative image
9	crossed-out (characters still legible but marked as to be deleted)
10	primary (default) font
11	first alternative font
12	second alternative font
13	third alternative font
14	fourth alternative font
15	fifth alternative font
16	sixth alternative font
17	seventh alternative font
18	eighth alternative font
19	ninth alternative font
21	doubly underlined
22	normal intensity (neither bold nor faint)
23	not italicize
24	not underlined (neither singly nor doubly)
25	steady (not blinking)
26	variable spacing
27	positive image
29	not crossed-out
50	not variable spacing

11.1.9   *SHS - Select character spacing*

A control function with a selective parameter which specifies the character spacing to be applied to constant spacing fonts in subsequent text.

The specified character spacing remains until it is changed by a subsequent occurrence of either SHS or the control function SCS (Set Character Spacing) in the current basic object.

The meaning of the parameter value is:

0   120 BMUs
1   100 MBUs
2   80 BMUs
3   200 BMUs
4   400 BMUs

The default value of the parameter is 0.

11.1.10   *SLS - Set line spacing*

A control function with one numeric parameter which specifies the line spacing for subsequent text.

The specified line spacing takes effect immediately and remains in effect until it is changed by a subsequent occurrence of either SLS or the control function SVS (Select Line Spacing) in the current basic component.

The line spacing is expressed as an integral multiple of SMUs.

The default value of the parameter is the equivalent of 200 BMUs.

11.1.11   *SRS - Start reverse string*

A control function with one parameter which is used to indicate either the start or end of a string graphic characters that are to be imaged in the direction opposite to that of the immediately preceding text (see § 5.2.4).

SRS with parameter value 1 indicates the start of the string.

SRS with parameter value 0 indicates the end of the string.

Hard and soft line terminators shall not be used between SRS 1 and SRS 0.

Any occurrence of the control functions PLD (Partial line Down), PLU (Partial Line Up), VPB (Line Position Backward) or VPR (Line Position Relative) within the string of characters delimited by SRS 1 and SRS 0 must be matched by an occurrence of the opposite control function within the string. Strings delimited by SRS may be nested.

11.1.12   *STAB - Selective tabulation*

A control function with a selective parameter which references a tabulation stop position in an associated "line layout table" (see § 7.1.13).

This control function specifies the positioning of the subsequent text, until either the occurrence of another STAB or the end of the current line.

This text is to be positioned at the referenced tabulation stop and aligned in accordance with the properties specified for that tabulation stop.

No default value of the parameter is specified.

11.1.13   *SUB - Substitute character*

A control function which is used in the place of a character that has been found invalid or in error.

11.1.14   *SVS - Select line spacing*

A control function with a selective parameter which specifies the line spacing for subsequent text. The specified line spacing takes effect immediately and remains in effect until it is changed by a subsequent occurrence of either SVS or the control function SLS (Set Line Spacing) in the current basic object.

The meaning of the parameter value is:

0   200 BMUs
1   300 MBUs
2   400 BMUs
3   100 BMUs
4   150 BMUs
9   600 BMUs

The default value of the parameter is 0.

11.1.15   *VPB - Line position backward*

A control function with one numeric parameter which causes the active position to be moved in the opposite direction to the line progression the number of SMUs specified by the parameter.

The combined effect of all occurrences of the control functions VPB and VPR within a given line must be such that the active position is returned to the reference line before the occurrence of a hard or soft line terminator.

The default value of the parameter is the equivalent of 100 BMUs.

Note - The main purposes of VPB are to provide for positioning of parallel annotation and for explicit control for the positioning of superscripts.

11.1.16   *VPR - Line position relative*

A control function with one numeric parameter which causes the active position to be moved in the direction to line progression the number of SMUs specified by the parameter.

The combined effect of all occurrences of the control functions VPB and VPR within a given line must be such that the active position is returned to the reference line before the occurrence of a hard or soft line terminator.

The default value of the parameter is the equivalent of 100 BMUs.

Note - The main purposes of VPR are to provide for positioning of parallel annotation and for explicit control for the positioning of subscripts.

11.1.17   *Code extension control functions*

This is a category of control functions used for the designation and invocation of graphic character sets. They are defined in ISO 6429 and their use is defined in ISO 2022.

11.2   *Layout control functions*

11.2.1   *BS - Backspace*

A control function that causes the active position to be moved, in the direction opposite to the character path, a distance specified by the most recent occurrence either of the control functions SHS (Select Character Spacing) or SCS (Set Character Spacing), if any, or otherwise by the presentation attribute "character spacing".

Note 1 - BS allows for the positioning of item identifiers (see § 5.2.9) on systems which do not implement the control function HPB (Character Position Backward). It is included in this Recommendation only for compatibility with Recommendation T.61 (1984). Its use in other document application profiles based upon the T.410 Series is deprecated. It shall not be used to cause character images to be superimposed.

Note 2 - The control function SSW (Set SPACE Width) has no effect on BS.

11.2.2   *HPB - Character position backward*

A control function with one numeric parameter which causes the active position to be moved in the opposite direction to the character path the number of SMUs specified by the parameter.

The default value of the parameter is the equivalent of 120 BMUs.

Note - The main purposes of HPB are to move the active position backwards from the line home position, and to provide for the positioning of parallel annotation.

11.2.3 HPR - *Character position relative*

A control function with one numeric parameter which causes the active position to be moved in the direction to the character path the number of SMUs specified by the parameter.

The default value of the parameter is the equivalent of 120 BMUs.

Note 1 - Although HPR has a control effect similar to that of one or more space characters, it does not have the graphic equivalence of space characters. Therefore, HPR does not cause spaces to be imaged in accordance with the current graphic rendition, such as underlined, possibly specified by a preceding occurrence of the control function SGR (Select Graphic Rendition).

Note 2 - HPR also provides for the positioning of parallel annotation.

11.2.4 JFY - *No justify*

A control function with a selective parameter which is used at the beginning of a line to indicate that the line must not be justified. It has no effect on subsequent lines.

The only parameter value that is permitted to be used is zero, which is also the default parameter value.

11.2.5 SACS - *Set additional character separation*

A control function with one numeric parameter which specifies increased escapement between graphic characters in subsequent text i.e. a positive value for the inter-character space (see § 5.2.1).

The specified value applies after the first subsequent graphic character and remains in effect until the next occurrence of SACS or a control function SRCS (Set Reduced Character Separation) or until it is reset to 0 by a subsequent occurrence of a hard or soft line terminator.

The parameter value is expressed in SMUs.

The default value of the parameter is 0.

11.2.6 SRCS - *Set reduced character separation*

A control function with one numeric parameter which specifies reduced escapement between graphic characters in subsequent text i.e. a negative value for the inter-character space (see § 5.2.1).

The specified value applies after the first subsequent graphic character and remains in effect until the next occurrence of SRCS or a control function SACS (Set Additional Character Separation) or until it is reset to 0 by a subsequent occurrence of a hard or soft line terminator.

The parameter value is expressed in SMUs.

The default value of the parameter is 0.

11.2.7 SSW - *Set SPACE width*

A control function with one numeric parameter which specifies the characters escapement associated with the character SPACE for subsequent text. The specified value takes effect immediately and remains in effect until it is changed by a subsequent occurrence of SSW or reset to the default value by a subsequent occurrence of a hard or soft line terminator.

The parameter value is expressed in SMUs.

No default parameter is specified. The default width of SPACE is equal to the character spacing if the current font has constant spacing and is otherwise determined by the font concerned.

11.3 *Logical control functions*

11.3.1 BPH - *Break permitted here*

A control function which indicates a point where a line break must occur when text is formatted (see § 12.2.1.3.2).

11.3.2 *NBH - No break here*

A control function which indicates a point where no line break must occur when text is formatted (see § 12.2.1.3.2).

Note - The graphic character NBSP (No Break SPace) should be taken to be equivalent to SPACE followed by NBH. (

11.3.3 *PTX - Parallel texts*

A control function which delimits passages of text which are interchanged one after the other, but intended to be presented in parallel with one another (see § 5.2.5).

The only parameter values permitted are:

0
1
3

The default value of the parameter is 0.

PTX with parameter value 1 is the opening delimiter of the first (principal) of two passages of text intended to be presented in parallel with one another.

PTX with parameter value 3 is the closing delimiter of the first passage of text and the opening delimiter of the second (supplementary) passage intended to be presented in parallel with the first.

PTX with parameter value 0 indicates the end of the supplementary passage of text.

Note - Japanese Ruby permits the specification of exactly one supplementary passage of text.

11.4 *Delimiters*

11.4.1 *SOS - Start of string*

A control function that acts as the opening delimiter of a string of graphic characters and/or control functions that is marked to facilitate its removal by a subsequent content layout process. The string is closed by the terminating delimiter control function ST (String Terminator).

A string thus delimited may contain occurrences of graphic characters and control functions, in particular CR (Carriage Return), LF (Line Feed) and HYPHEN, introduced as a result of a formatting process (see § 12.2.1.3).

11.4.2 *ST - String terminator*

A control function that acts as the terminating delimiter of a string opened by the delimiter control function SOS (Start Of String).

11.5 *SP - Space*

A character with properties of both a graphic character and a logical control function.

As a control function, SP is significant to the content layout process. It acts as a word delimiter and indicates a potential line break point except when it is immediately followed by another SP or by an occurrence of the control function NBH (No Break Here) (see § 11.3.2).

As a graphic character, SP causes the active position to be advanced without a graphic symbol to be imaged. However, any graphic renditions that are in effect e.g. underlining, also apply to SP.

Any SP(s) that precede a line terminator, and follow the last graphic character of a line, are ignored by the imaging process.

12   Content layout process

This section describes a content layout process for basic logical objects associated with content architectures of type character.

Its purpose is to aid understanding of the semantics of the presentation attributes and control functions by describing the required results of such a process. However, it is not intended to specify any process that might be carried out in a particular implementation to achieve these results.

12.1 *Introduction*

12.1.1 *Purpose*

The content layout process describes a process of formatting and laying out character content into an allocated area. This area is referred to as the available area and is determined by the document layout process defined in Recommendation T.412.

The purpose of the content layout process is to convert content associated with basic logical components into content associated with basic layout objects. This might imply a transformation of the content from one form to another.

The content layout process results in the creation of a basic layout object(s) into which the content is to be positioned. The dimensions of each basic layout object is returned to the document layout process which determines the precise position of that basic layout object within the available area.

12.1.2 *Available area*

The content layout process is constrained by the available area. The maximum line length is constrained by the dimension of the available area in the direction of the character path.

During the layout of the content associated with a basic logical object into a basic layout object, the following cases can occur:

- The formatted or formatted processable content fits into the available area.

- The formatted or formatted processable content does not fit into the dimensions of the available area in the direction of line progression. In this case, an additional or a new available area is required depending upon any constraints imposed by the document layout process.

- The formatted content does not fit into the dimension of the available area in the direction of the character path. In this case, a large available are is required.

Note – This case is most likely to occur when laying out formatted content associated with a basic logical object.

12.1.3 *Presentation attributes*

The content layout process takes into account the presentation attributes applying to the basic logical object with which the content is associated. It also takes into account any control functions that are embedded in the content.

The presentation attributes applying to the content layout process can be specified in the generic layout structure and presentation styles. The values of these presentation attributes are determined according to the defaulting rules specified in Recommendation T.412.

12.1.4 *Character content architecture classes*

The content layout process is described for basic logical objects associated with content that conforms to any of the three character content architecture classes (see § 4.1) as follows:

- processable form character content in which the content layout process provides for formatting of the content. The content layout process results in the output of content in formatted or formatted processable form depending upon the desired form of document;

- formatted processable form character content in which the content layout process provides for reformatting of the content. This involves an initialization process which must be carried out on the content before the content layout process can be applied to that content. The content layout process results in the output of content in formatted or formatted processable form depending upon the desired form of document;

- formatted form character content in which the content layout process has no effect on the content itself but still determines the dimensions of the basic layout object into which that content is to be positioned.

12.1.5 *Use of delimiters*

When formatted processable form content is created as a result of the content layout process, all shared control functions and graphic characters inserted into the content as a result of the content layout process are enclosed between the delimiter control functions SOS (Start Of String) and ST (String Terminator).

12.1.6 *Layout of the content*

For each of the three character content architecture classes, three cases of laying out the content of basic logical objects into basic layout objects are possible:

-   single basic logical object to single basic layout object: the content of a single basic logical object can be laid out into a single basic layout object and is the only content associated with this basic layout object;

-   single basic logical object to multiple basic layout object: the content of a single basic logical object is split among two or more basic layout objects, i.e. the content portions associated with two or more basic layout objects are derived from a single basic logical object;

-   multiple basic logical object to single layout object: the content of two or more basic logical objects is laid out into a single basic layout object, i.e. the content portions of two or more basic logical objects are associated with a single basic layout object.

Multiple basic logical object to multiple basic layout object is also possible, but not described explicitly since this is a combination of the last two cases above.

12.1.7 *Layout sequence*

In all cases, the same sequence of steps for laying out content associated with a basic logical object is executed as follows:

-   initialization;
-   determination of initial point;
-   formatting of the content;
-   identification of content portions;
-   determination of basic layout object dimensions;
-   determination of the value of the presentation attribute "initial offset".

This also results in the creation of a basic layout object.

Although the sequence of steps is the same in all cases, the action performed at individual steps may vary.

12.2 *Content layout process for processable content*

12.2.1 *Single basic logical object to single basic layout object*

12.2.1.1 *Initialization*

In the case of processable form content, no initialization of the content is necessary.

12.2.1.2 *Determination of initial point*

The location of the initial point depends upon:

-   the presentation attributes "character path" and "line progression" (determining the start edge and top edge of the basic layout object);

-   the presentation attribute "kerning offset" (specifying the location of the start edge of the positioning area relative to the start edge of the basic layout object);

-   the presentation attribute "indentation" (specifying the distance between the initial point and the start edge of the positioning area);

- the invocation of a font by the presentation attribute "graphic rendition" and the presentation attribute "character orientation" (specifying the minimum backward extent of the first line box);

- the presence of control functions PLU (Partial Line Up), PTX (Parallel Texts), VPB (Line Position Backward) and font invocation by SGR (Select Graphic Rendition) in the first line of characters to be imaged (modifying the backward extent of the first line box).

The position of the initial point relative to the start edge and top edge of the positioning area is determined such that:

- its distance from the top edge of the positioning area is equal to the backward extent of the first line box;

- its distance from the start edge of the positioning area is equal to the value specified by the presentation attribute "indentation".

The position of the initial point relative to the upper left corner of the basic layout object can be determined only after the dimension of the basic layout object has been determined (see § 12.2.1.5). This value is assigned to the presentation attribute "initial offset" and should always be specified explicitly in order to achieve the desired result by the content imaging process.

12.2.1.3   *Formatting of the content*

Formatting of the content involves:

- the positioning of character images within a line box (see § 5.2);

- the determination of line breaks;

- the positioning of line boxes within the basic layout (see § 5.3).

It may involve the insertion of control functions and the assignment of presentation attribute values to the basic layout object.

As a general rule, the available area is utilized as much as possible in the direction of the character path.

12.2.1.3.1   *Position of character images within a line box*

There are two sets of operations that have an effect on the positioning of character images within a line box. One of these sets is related to the presentation attribute "formatting indicator", the other is not.

The operations related to the attribute "formatting indicator" are:

- alignment (presentation attribute "alignment");

- tabulation (presentation attribute "line layout table" and the control function STAB);

- first line offset (presentation attribute "first line offset");

- itemization (presentation attribute "itemization");

- pairwise kerning (presentation attribute "pairwise kerning").

Alignment and tabulation are mutually exclusive but itemization and first line offset may be applied in combination with either of them.

These operations may be performed by either the content layout process or the content imaging process. However, they can be performed by the content layout process only if the presentation attribute "formatting indicator" and the control functions HPB (Character Position Backward), HPR (Character Position Relative), SACS (Set Additional Character Separation), SRCS (Set Reduced Character Separation) and SSW (Set SPACE Width) are available in the content architecture level concerned.

The result of performing these operations by the content layout process is the insertion of the above control functions in the content.

If the content layout process has performed all of the specified operations for a basic layout object and inserted all necessary control functions, the value of the presentation attribute ''reformatting indicator'' is set to 'yes', otherwise it is set to 'no'.

Whether or not the content layout process inserts these control functions, it always determines the allocation of characters to each line box and the dimensions of each line box.

Note - If the value of the presentation attribute ''formatting indicator'' is 'no' or if a font substitution has been made, then the content imaging process should perform these operations.

The other set of operations which are not related to the presentation attribute "formatting indicator" are:

- character ordering (control function SRS);

- parallel annotation (control function PTX);

- graphic character composition (control function GCC).

12.2.1.3.1.1 *Pairwise kerning*

If the presentation attribute ''pairwise kerning'' specifies 'yes' and the content layout process is capable of performing this function and the font used provides the necessary information, then certain combinations of character images are positioned closer to (or further apart from) each other than determined by their position and escapement points.

When pairwise kerning is performed by the content layout process, it will result in the insertion of control functions HPB (Character Position Backward) or HPR (Character Position Relative) between the two characters involved.

12.2.1.3.1.2 *First line offset*

The characters associated with the first line can be controlled to be laid out differently from the rest of the lines in this object.

The presentation attribute "first line offset" specifies if the first line has an overhang or indentation relative to the line home position.

When first line offset is performed by the content layout process, it will result in the insertion of a control function HPB (Character Position Backward) or HPR (Character Position Relative).

12.2.1.3.1.3 *Itemization*

The first line of a basic layout object may contain an item identifier. The position of the item identifier is controlled by the presentation attribute "itemization".

When itemization is performed by the content layout process, it will result in the insertion of a control function HPB (Character Position Backward) or HPR (Character Position Relative) before the text of the item identifier.

12.2.1.3.1.4 *Alignment*

None of the alignments except 'start-aligned' can be performed until the dimensions of the basic layout object have been determined (see § 12.2.1.4).

When alignment is performed by the content layout process, the line length for alignment is determined to be:

- for the first line, the distance between the line home position and the end edge of the positioning area minus the value of the presentation attribute "first line offset";

- for all other lines, the distance between the line home position and the end edge of the positioning area.

The various values of the presentation attribute "alignment" are treated as follows:

- 'start-aligned' does not result in the insertion of any control functions;

- 'end-aligned' and 'centred' result in the insertion of a control function HPR (Character Position Relative) either before the first graphic character of each line or after the CR (Carriage Return) delimiting the item identifier if the presentation attribute ''itemization'' specifies a value other than 'no itemization';

- 'justified' results in the insertion of zero, one or more control functions SSW (Set SPACE Width), SACS (Set Additional Character Separation) and/or SRCS (Set Reduced Character Separation) in each line which ends with a line terminator inserted by the content layout process. The precise usage of SACS SRCS and SSW is implementation dependent.

Note - The presentation attribute "Alignment" does not apply to the item identifier.

When the presentation attribute ''alignment'' has the value 'justified', irrespective of whether the alignment is performed by the content layout process or the content imaging process, the control function JFY (No Justify) is inserted at the beginning of the last line of a character sequence in order to avoid justification of this line by the content imaging process.

12.2.1.3.1.5   *Tabulation*

When tabulation is performed by the content layout process, it results in the insertion of a control function HPR (Character Position Relative) or HPB (Character Position Backward) between each occurrence of the control function STAB (Selective Tabulation) and the first graphic character following it.

12.2.1.3.1.6   *Parallel annotation*

The occurrence of the control function PTX (Parallel Texts) in the content specifies that a string of characters is to be laid out as a parallel annotation to another specified string of characters. The result of the content layout process is as described in § 5.2.5.

If the output of the content layout process is in formatted form, the positioning of the parallel annotation is achieved by the removal of all occurrences of PTX and the insertion of the appropriate control functions HPR (Character Position Relative), HPB (Character Position Backward), VPR (Line Position Relative) and VPB (Line Position Backward).

If the output of the content layout process is in formatted processable form, the positioning of the parallel annotation is achieved by inserting the control functions HPR (Character Position Relative), HPB (Character Position Backward), VPR (Line Position Relative) and VPB (Line Position Backward) and enclosing them by the delimiter control functions SOS (Start Of String) and ST (String Terminator).

12.2.1.3.1.7   *Character ordering*

The occurrence of the control function SRS (Start Reverse String) in the content of a basic logical object controls the direction of imaging of the interchanged characters. The result of the content layout process is as described in § 5.2.4.

12.2.1.3.1.8   *Graphic character composition*

The control function GCC (Graphic Character Composition) is used to combine two or more graphic characters into a single symbol. The width of these graphic characters may be less than the sum of the widths of the images of the component characters.

12.2.1.3.2   *Insertion of line breaks*

The formatting process may cause the insertion of additional line breaks into the content. Existing hard line terminators (CR/LF combinations) that are already in the content remain in the content but must be taken into account during the formatting process.

If the output is in formatted processable form, the inserted line breaks consist of soft line terminators represented by the control function sequence SOS CR LF ST. Optionally, characters inserted by an implementation dependent hyphenation algorithm are included in the SOS-ST string. No characters are deleted from the content.

If the output is in formatted form, the inserted line breaks consist of hard line terminators represented by the control function sequence CR LF. All occurrences of the control functions BPH (Break Permitted Here) and NBH (No Break Here) are deleted.

It is the intention of the formatting process that the number of characters between the inserted line breaks is the maximum possible for each line. the exact algorithm for inserting line breaks is implementation dependent and is not defined in the T.410 Series. However, the line break algorithm shall conform to the following constraints:

- a soft line break may by inserted:

    - after a SP which is not immediately followed by another SP or the control function NBH (No Break Here);

    - after the control function BPH (Break Permitted Here);

    - at a point determined by an implementation or language dependent algorithm;

- a line break is not permitted:

    - when a subscribed rendition is active;

    - within a string with reversed presentation direction;

    - within a string of parallel annotation.

12.2.1.3.3   *Positioning of line boxes*

The first line box is positioned with its line home position at the initial point as described in § 12.2.1.2.

Each line box is positioned with its line home position on the line from the initial point in the direction of line progression.

When proportional line spacing is not to be performed, the distance between the line home positions of two successive line boxes is equal to the current line spacing. The initial value of the current line spacing is the value of the presentation attribute "line spacing". This value may be altered by occurrences of the control function SVS (Select Line Spacing) and SLS (Set Line Spacing).

When proportional line spacing is to be performed by the content layout process, the distance between the line home positions of two successive line boxes is evaluated by an implementation dependent algorithm. If the evaluated distance differs from the current value of line spacing, the control function SLS (Set Line Spacing) is inserted before the line terminator of the first line. The evaluated distance is inserted as the parameter of this control function and also becomes the current line spacing.

If the output of the content layout process is in formatted processable form, those occurrences of the control function SLS (Set Line Spacing) inserted by the content layout process are enclosed by the delimiters SOS (Start Of String) and ST (String Terminator).

12.2.1.4   *Identification of content portions*

The content layout process shall also provide a value for the attribute "Content portion identifier - layout" for each content portion associated with the layout structure.

12.2.1.5   *Determination of basic layout object dimensions*

The formatting action continues until all the available content has been allocated or all the available area has been filled.

The content layout process attempts to allocate the minimum portion of the available area sufficient to hold all of the available content. Hence, the dimensions of the basic layout object can only be determined once formatting is complete.

The mapping to horizontal and vertical dimensions of the basic layout object is dependent upon the character path as follows:

Character path	Dimension in direction of character path	Dimension in direction orthogonal to character path
0, 180	Horizontal dimension	Vertical dimension
90, 270	Vertical dimension	Horizontal dimension

12.2.2 *Single basic logical object to multiple basic layout objects*

If the formatted content does not fit into the available area in the direction of line progression then an additional available area has to be obtained from the document layout process e.g. in the case of a page boundary. In this case, the content of a single basic logical object is allocated to more than one basic layout object. The original content portion is split into several content portions, each corresponding to a different basic layout object.

There are two changes with respect to the description in § 12.2.1:

- the presentation attributes of the second and subsequent basic layout object are given values corresponding to the status of rendition (line spacing, character spacing, graphic renditions etc.) at the end of the previous basic layout object;

- the allocation of content to basic layout objects is to be performed such that the presentation attributes "orphan size" and "widow size" are fulfilled.

12.2.3 *Multiple basic logical objects to single basic layout object*

When concatenation results in the content associated with a more than one basic logical object being laid out in a single basic layout object (see § 7.5), it may be necessary for the content layout process to insert control functions at the beginning of the second and subsequent basic logical objects so that the values of certain presentation attributes associated with those basic logical objects are applied. These control functions are:

- SHS (Select Character Spacing) or SCS (Set character Spacing) for "character spacing";

- Designation and/or invocation sequences for "graphic character sets";

- IGS (Identify Graphic Subrepertoire) for "graphic character subrepertoire";

- SGR (Select Graphic Rendition) for "graphic rendition";

- SVS (Select Line Spacing) or SLS (Set Line Spacing) for "line spacing".

If the output of the content layout process is in formatted processable form, then the control functions inserted by the content layout process are enclosed by the delimiters SOS (Start Of String) and ST (String Terminator).

The presentation attribute "proportional line spacing" specified for the second or subsequent basic logical objects is interpreted by the content layout process as described in § 12.2.1.

The other presentation attributes specified for the second and subsequent basic logical objects are ignored (see § 7.5).

12.3 *Content layout process for formatted processable content*

For formatted processable form content, the initialization step of the content layout process shall:

- remove all SOS-ST control strings from the content;

- remove all layout control functions (BS, HPB, HPR, JFY, SACS, SRCS and SSW) from the content (see § 11.2);

- combine all content portions associated with the same basic logical object into a single content portion in order to prevent unnecessary fragmentation of the document content which could otherwise occur;

- delete the content portion attribute "content layout identifier - layout", if present.

After initialization, the content is in processable form. The remaining steps of the content layout process are as described for processable content (see § 12.2).

12.4 *Content layout process for formatted content*

For formatted content, the content layout process still has to determine the dimensions of the basic layout object to be allocated.

The same steps of the content layout process are used as for processable form content. In this case, however, formatting only involves:

- the positioning of character images within a line box as described in §§ 5.2 and 12.2.1.3.1;

- the positioning of line boxes within a basic layout object as described in § 5.3.

13   Content imaging process

This section describes a content imaging process for basic layout objects associated with content architecture of type character.

Its purpose is to aid understanding of the semantics of the presentation attributes and control functions by describing the required results of such a process. However, it is not intended to specify any process that might be carried out in a particular implementation to achieve these results.

13.1   *Introduction*

The content imaging process is only concerned with the logical structures, the presentation styles and the content of basic layout components conforming to this Recommendation.

All logical control functions, if any, are ignored.

The content imaging process is only applicable to the formatted and formatted processable form character content architecture classes.

13.2   *Content imaging process for formatted content*

This section describes how the various shared and layout presentation attributes and shared and layout control functions influence the image of the contents.

Most shared presentation attributes and shared control functions serve the purpose of positioning and orienting character images along reference lines and for positioning and orienting these reference lines within the basic layout object.

Thus, the effect of most shared presentation attributes and shared control functions have already been described in the content layout process.

Most layout presentation attributes and control functions are related to positioning and, thus, have already been described as the result of the content layout process.

The following sub-sections provide additional information relating to the content imaging process.

13.2.1   *Determination of initial point*

The active position for imaging is set on the initial point within the basic layout object. This information is derived from the presentation attribute "initial offset".

13.2.2   *Choosing character images*

The following presentation attributes and control functions determine the character images to be chosen for imaging:

- "graphic character sets" and code extension announcer, designation and invocation control functions;

- "character fonts" (together with the attribute "fonts list" in the document profile);

- "graphic rendition" and SGR (Select Graphic Rendition).

If the specified font is not available, the content imaging process may decide to provide a substitute for this font by making use of the font information available in the document profile attribute "fonts list".

13.2.3 *Formatting indicator*

The presentation attribute "formatting indicator" specifies whether first line offset, itemization, alignment, tabulation and pairwise kerning have already been performed by the content layout process or not.

If not, or if the result from the content layout process has been invalidated by a font substitution, then the content imaging process must perform the task in the same way as described in the content layout process (see § 12.2.1.3.1).

13.3 *Content imaging process for formatted processable content*

For content in formatted processable form, the only difference from the case of formatted form is that all logical control functions and the delimiter control functions SOS (Start Of String) and ST (String Terminator) are ignored.

The effect of shared and layout presentation attributes and shared and layout control functions is as described in § 13.2.

14 Interactions between presentation attributes and control functions

This section contains a summary of the interactions among and between presentation attributes and control functions defined in various places in this Recommendation as follows:

- LF is restricted to be used in the following cases:

  1) at the beginning of the content of a basic layout component;

  2) immediately following a control function CR (Carriage Return);

  3) immediately following another LF.

- BPH or CR is not permitted in the following cases:

  1) when PLU or PLD is active;

  2) after VPR and/or VPB have moved the active position away from the reference line;

  3) between the control functions SRS 1 and SRS 0;

  4) between the control functions PTX 1 and PTX 0.

- Rendition aspects defined by some presentation attributes can be overridden by control functions embedded in content portions as follows:

Presentation attribute	Control function
Character spacing	SHS, SCS
Line spacing	SVS, SLS
Graphic rendition	SGR
Graphic character subrepertoire	IGS
Graphic character sets	Code extension control functions

- When the presentation attribute "line layout table" specifies any tabulation stops, the presentation attribute "alignment" is assumed to have the value "start-aligned".

15   Definition of character content architecture classes

This section defines the three classes of character content architecture as described in § 5, namely:

- a formatted form character content architecture which allows for document content to be presented (e.g. printed or displayed) as intended by the originator. Formatted form can be used in any basic component;

- a processable form character content architecture which allows for document content to be processed (e.g. edited or formatted). Processable form can be used in any basic logical component;

- a formatted processable form character content architecture which allows for document content to be processed and also to be presented as intended by the originator. Formatted processable form can be used in any basic component.

Paragraphs 15.1, 15.2 and 15.3 specify the categories of presentation attributes and control functions that pertain to these content architecture classes. The individual presentation attributes and control functions are summarized in Tables 5 and 6/T.416. The permissible values and the default values of the presentation attributes and the control functions parameter values are defined in §§ 7 and 11 respectively.

TABLE 5/T.416

Summary of presentation attributes

Character content architecture class	Formatted processable		
		Processable	
Presentation attribute	Formatted		
Alignment	X	X	X
Character fonts	X	X	X
Character orientation	X	X	X
Character path	X	X	X
Character spacing	X	X	X
Code extension announcers	X	X	X
First line offset	X	X	X
Formatting indicator	X		X
Graphic character sets	X	X	X
Graphic character subrepertoire	X	X	X
Graphic rendition	X	X	X
Indentation		X	X
Initial offset	X		X
Itemization	X	X	X
Kerning offset	X	X	X
Line layout table	X	X	X
Line progression	X	X	X
Line spacing	X	X	X
Orphan size		X	X
Pairwise kerning	X	X	X
Proportional line spacing		X	X
Widow size		X	X

TABLE 6/T.416

Summary of control functions

Character content architecture class	Formatted processable		
		Processable	
Control function	Formatted	Processable	
BPH (Break Permitted Here)		X	X
BS (Backspace)	X		X
CR (Carriage Return)	X	X	X
GCC (Graphic Character Composition	X	X	X
HPB (Character Position Relative)	X		X
HPR (Character Position Backward)	X		X
IGS (Identify Graphic Subrepertoire	X	X	X
JFY (No Justify)	X		X
LF (Line Feed)	X	X	X
NBH (No Break Here)		X	X
PLD (Partial Line Down)	X	X	X
PLU (Partial Line Up)	X	X	X
PTX (Parallel Texts)		X	X
SACS (Set Additional Character Spacing)	X		X
SCS (Set Character Spacing)	X	X	X
SGR (Select Graphic Rendition)	X	X	X
SHS (Select Character Spacing)	X	X	X
SLS (Set Line Spacing)	X	X	X
SOS (Start Of String)			X
SP (Space)	X	X	X
SRCS (Set Reduced Character Spacing)	X		X
SRS (Start Reverse String)	X	X	X
SSW (Set SPACE Width)	X		X
ST (String Terminator)			X
STAB (Selective Tabulation)	X	X	X
SUB (Substitute)	X	X	X
SVS (Select Line Spacing)	X	X	X
VPB (Line Position Backward)	X	X	X
VPR (Line Position Relative)	X	X	X
Code extension control functions	X	X	X

In order to aid the definition of content architecture levels for use in document application profiles (see Recommendation T.411), the presentation attributes and the control functions that are applicable to each content architecture class are listed in Annex A, together with their permissible values and default values.

15.1 Formatted character content architecture class

The following categories of presentation attributes and control functions pertain to the formatted character content architecture class:

- shared presentation attributes (see § 7.1);

- layout presentation attributes (see § 7.2);

- shared control functions (see § 11.1);

- layout control functions (see § 11.2).

15.2 *Processable character content architecture class*

The following categories of presentation attributes and control functions pertain to the processable character content architecture class:

- shared presentation attributes (see § 7.1);

- logical presentation attributes (see § 7.3);

- shared control functions (see § 11.1);

- logical control functions (see § 11.3).

15.3 *Formatted processable character content architecture class*

The following categories of presentation attributes and control functions pertain to the formatted processable character content architecture class:

- shared presentation attributes (see § 7.1);

- layout presentation attributes (see § 7.2);

- logical presentation attributes (see § 7.3);

- shared control functions (see § 11.1);

- layout control functions (see § 11.2);

- logical control functions (see § 11.3);

- delimiters (see § 11.4).

ANNEX A

(to Recommendation T.416)

Summary of content architecture classes

(Informative)

This Annex summarizes the present attributes and control functions that apply to each of the three content architecture classes (formatted, processable and formatted processable) defined in section 14, together with their permissible values and default values.

The purpose of this Annex is to facilitate the definition of content architecture levels for use in document application profiles (see Recommendation T.411).

A.1   *Formatted character content architecture class*

A.1.1   *Presentation attributes*

Attribute	Permissible values	Default value
alignment	start aligned end aligned centred justified	start aligned
character fonts	1) any positive integer 2) any positive integer	none none
character orientation	0, 90, 180, 270 degrees	0 degrees
character path	0, 90, 180, 270 degrees	0 degrees
character spacing	any positive integer	equivalent of 120 BMUs
code extension announcers	any string of escape sequences in accordance with ISO 2022	escape sequences for G0 and G2 sets
first line offset	any integer	0
formatting indicator	no, yes	no
graphic character sets	the escape sequences to designate, and any required locking shift functions to invoke, one or more registered graphic character sets	the escape sequences to designate, and the locking shift functions to invoke, the graphic character sets of ISO 6937-2
graphic character subrepertoire	0 or the identifier of any registered subrepertoire of ISO 6937	0
graphic rendition	0, 1-7, 9, 10-19, 21-27, 29, 50	0
initial offset	1) any non-negative integer 2) any non-negative integer	see Table 2/T.416 see Table 2/T.416

itemization	1) no itemization
	   start-aligned
	   end-aligned
	2) any integer
	
	3) any integer
		no itemization
		
		the distance from the start edge of the positioning area to the line home position
		0

Attribute	Permissible values	Default value
kerning offset	1) any non-negative integer 2) any non-negative integer	0 0

line layout table	1) any
	2) any
	3) start-aligned
	   end-aligned
	   centred
	   aligned-around
	4) any
		No tabulation stops
		defined

line progression	90, 270 degrees	270 degrees
line spacing	any positive integer	equivalent of 200 BMUs

A.1.2 *Control functions*

Control function	Permissible value(s)	Default value(s)
BS	not applicable	not applicable
CR	not applicable	not applicable
GCC	0, 1, 2	0
HPB	any positive integer	equivalent of 120 BMUs
HPR	any positive integer	equivalent of 120 BMUs
IGS	0 or the identifier of any registered subrepertoire of ISO 6937	0
JFY	0	0
LF	not applicable	not applicable
PLD	not applicable	not applicable
PLU	not applicable	not applicable
SACS	any positive integer	0
SCS	any positive integer	equivalent of 120 MBUs
SGR	0, 1-7, 9, 10-19, 21-27, 29, 50	0
SHS	0, 1, 2, 3	0

Control function	Permissible value(s)	Default value(s)
SLS	any positive integer	equivalent of 200 MBUs
SP	not applicable	not applicable
SRCS	any positive integer	0
SRS	0, 1	0
SSW	any positive integer	none
STAB	any	none
SUB	not applicable	not applicable
SVS	0, 1, 2, 3, 4, 9	0
VPB	any positive integer	equivalent of 100 BMUs
VPR	any positive integer	equivalent of 100 BMUs

In addition, any code extension control function defined in ISO 2022, within the scope of the value of the attribute "code extension announcers", is permitted.

A.2   *Processable character content architecture class*

A.2.1   *Presentation attributes*

Attribute	Permissible values	Default value
alignment	start aligned end aligned centred justified	start aligned
character fonts	1) any positive integer 2) any positive integer	none none
character orientation	0, 90, 180, 270 degrees	0 degrees
character path	0, 90, 180, 270 degrees	0 degrees
character spacing	any positive integer	equivalent of 120 BMUs
code extension announcers	any string of escape sequences in accordance with ISO 2022	escape sequences for G0 and G2 sets
first line offset	any integer	0

Attribute	Permissible values	Default value
graphic character sets	the escape sequences to designate, and any required locking shift functions to invoke, one or more registered graphic character sets	the escape sequences to designate, and the locking shift functions to invoke, the graphic character sets of ISO 6937-2
graphic character subrepertoire	0 or the identifier of any registered subrepertoire of ISO 6937	0
graphic rendition	0, 1-7, 9, 10-19, 21-27, 29, 50	0
indentation	any non-negative integer	0

itemization	1) no itemization
	   start-aligned
	   end-aligned
	2) any integer
	
	3) any integer
		no itemization
		
		the distance from the start edge of the positioning area to the line home position
		0

kerning offset	1) any non-negative integer 2) any non-negative integer	0 0

line layout table	1) any
	2) any
	3) start-aligned
	   end-aligned
	   centred
	   aligned-around
	4) any
		none
		none
		start-aligned
		
		none

line progression	90, 270 degrees	270 degrees
line spacing	any positive integer	equivalent of 200 BMUs
orphan size	any positive integer	1
pairwise kerning	yes, no	no
proportional line spacing	yes, no	no
widow size	any positive integer	1

A.2.2 Control functions

Control function	Permissible value(s)	Default value(s)
BPH	not applicable	not applicable
CR	not applicable	not applicable
GCC	0, 1, 2	0
IGS	0, or the identifier of any registered subrepertoire of ISO 6937	0
LF	not applicable	not applicable
NBH	not applicable	not applicable
SCS	not positive integer	equivalent of 120 BMUs
SGR	0, 1-7, 9, 10-19, 21-27, 29, 50	0
SHS	0, 1, 2, 3	0
SLS	any positive integer	equivalent of 200 BMUs
SP	not applicable	not applicable
SRS	0, 1	0
STAB	any	none
SUB	not applicable	not applicable
SVS	0, 1, 2, 3, 4, 9	0
VPB	any positive integer	equivalent of 100 BMUs
VPR	any positive integer	equivalent of 100 BMUs

In addition, any code extension control function defined in ISO 2022, within the scope of the value of the attribute "code extension announcers", is permitted.

A.3 Formatted processable character content architecture class

A.3.1 Presentation attributes

Attribute	Permissible values	Default value
alignment	start aligned end aligned centred justified	start aligned
character fonts	1) any positive integer 2) any positive integer	none none

Attribute	Permissible values	Default value
character orientation	0, 90, 180, 270 degrees	0 degrees
character path	0, 90, 180, 270 degrees	0 degrees
character spacing	any positive integer	equivalent of 120 BMUs
code extension announcers	any string of escape sequences in accordance with ISO 2022	escape sequences for G0 and G2 sets
first line offset	any integer	0
formatting indicator	no, yes	no
graphic character sets	the escape sequences to designate, and any required locking shift functions to invoke, one or more registered graphic characters	the escape sequences to designate, and the locking shift functions to invoke, the graphic character sets of ISO 6937-2
graphic character subrepertoire	0 or the identifier of any registered subrepertoire of ISO 6937	0
graphic rendition	0, 1-7, 9, 10-19, 21-27, 29, 50	0
initial offset	1) any non-negative integer 2) any non-negative integer	see Table 2/T.416 see Table 2/T.416
indentation	any non-negative integer	0

itemization	1) no itemization
	   start-aligned
	   end-aligned
	2) any integer
	
	3) any integer
		no itemization
		
		the distance from the start edge of the positioning area to the line home position
		0

kerning offset	1) any non-negative integer 2) any non-negative integer	0 0

Attribute	Permissible values	Default value

line layout table	1) any
	2) any
	3) start-aligned
	   end-aligned
	   centred
	   aligned-around
	4) any
		none
		none
		start-aligned
		
		none

line progression	90, 270 degrees	270 degrees
line spacing	any positive integer	equivalent of 200 BMUs
orphan size	any positive integer	1
pairwise kerning	yes, no	no
proportional line spacing	yes, no	no
widow size	any positive integer	1

A.3.2 *Control functions*

Control function	Permissible value(s)	Default value(s)
BPH	not applicable	not applicable
BS	not applicable	not applicable
CR	not applicable	not applicable
GCC	0, 1, 2	0
HPB	any positive integer	equivalent of 120 BMUs
HPR	any positive integer	equivalent of 120 BMUs
IGS	0 or the identifier of any registered sub-repertoire of ISO 6937	0
JFY	0	0
LF	not applicable	not applicable
NBH	not applicable	not applicable
PLD	not applicable	not applicable

Control function	Permissible value(s)	Default value(s)
PLU	not applicable	not applicable
PTX	0, 1, 3	0
SACS	any positive integer	0
SCS	any positive integer	equivalent of 120 MBUs
SGR	0, 1-7, 9, 10-19, 21-27, 29, 50	0
SHS	0, 1, 2, 3	0
SLS	any positive integer	equivalent of 200 BMUs
SOS	not applicable	not applicable
SP	not applicable	not applicable
SRCS	any positive integer	0
SRS	0, 1	0
SSW	any positive integer	none
ST	not applicable	not applicable
STAB	any	none
SUB	not applicable	not applicable
SVS	0, 1, 2, 3, 4, 9	0
VPB	any positive integer	equivalent of 100 BMUs
VPR	any positive integer	equivalent of 100 BMUs

In addition, any code extension control function defined in ISO 2022, within the scope of the value of the attribute "code extension announcers", is permitted.

ANNEXE B

(to Recommendation T.416)

Character content architecture levels

(Informative)

The content of this Annex is for information only and may be included in a future register of character content architecture levels and will then be deleted from this Recommendation.

This Annex contains the definitions of nine character content architecture levels for use in document application profiles. These have been defined in accordance with the rules contained in Recommendation T.411.

The content architecture levels defined in this section are listed below. They are grouped into four formatted content architecture levels (CF), three processable content architecture levels (CP) and two formatted processable content architecture levels (CFP):

- CF-0: This content architecture is functionally equivalent to Telex (CCITT ITA2);

- CF-1: This content architecture is functionally equivalent to conventional implementations of Recommendation T.50;

- CF-2: This content architecture is functionally equivalent to Teletex (Recommendation T.61);

- CF-3: This content architecture is an enhanced formatted form architecture which does not correspond to any existing standard and which incorporates all the features defined for its class;

- CP-0: This content architecture is functional equivalent to Recommendation X.420 (1984) – Simple formattable documents;

- CP-2: This content architecture is a processable form architecture that functionally corresponds to the formatted content architecture CF-2;

- CP-3: This content architecture is a processable form content architecture which does not correspond to any existing standard and which incorporates all the features defined for its class;

- CFP-2: This content architecture is the formatted processable form architecture which corresponds to the formatted form content architecture CF-2 and the processable form content architecture CP-2;

- CFP-3: This content architecture is a formatted processable form content architecture which does not correspond to any existing standard and which incorporates all the features defined for its class.

CF-0 is a subset of CF-1, which is a subset of CF-2, which in turn is a subset of CF-3.

CP-0 is a subset of CP-2, which in turn is a subset of CP-3.

CF-2 and CP-2 are subsets of CFP-2, which is a subset of CFP-3.

CF-3 and CP-3 are subsets of CFP-3.

B.1   *Character content architecture level CF-0*

B.1.1   *Type*

CF-0 is a character content architecture level pertaining to the formatted character content architecture class.

B.1.2   *Presentation attributes*

None are permitted to be specified.

It is assumed that all presentation attributes that are applicable to formatted character content architectures have values equal to the default values specified in this Recommendation except:

- "graphic character set": default value is the primary character set of ISO 6937-2;

- "graphic character subrepertoire": default value is the minimum subrepertoire of ISO 6937-2.

B.1.3   *Control functions*

Control function	Basic value(s)	Non-basic value(s)
CR	not applicable	not applicable
LF	not applicable	not applicable
SP	not applicable	not applicable

B.2   *Character content architecture level CF-1*

B.2.1   *Type*

CF-1 is a character content architecture level pertaining to the formatted character content architecture class.

B.2.2   *Presentation attributes*

Presentation attribute	Basic value(s)	Non-basic value(s)
code extension announcers	any string of escape sequences in accordance with ISO 2022	none
graphic character sets	the escape sequences to designate, and the locking of shift functions to invoke, the primary graphic character set of ISO 6937-2	the escape sequences to designate, and any required locking shift functions to invoke, any other registered graphic character sets
graphic character subrepertoire	0	the identifier of any registered subrepertoire of ISO 6937

The default value of these presentation attributes is as specified in Recommendation T.416.

It is assumed that all presentation attributes that are applicable to formatted character content architectures, and that are not specified in the above, have values equal to the default values specified in this Recommendation.

B.2.3   *Control functions*

Control function	Basic value(s)	Non-basic value(s)
CR	not applicable	not applicable
LF	not applicable	not applicable
SP	not applicable	not applicable

Any code extension control function defined in ISO 2022, within the scope of the value of the attribute "code extension announcers", is permitted.

B.3 *Character content architecture level CF-2*

B.3.1 *Type*

CF-2 is a character content architecture level pertaining to the formatted character content architecture class.

B.3.3 *Presentation attributes*

Presentation attribute	Basic value(s)	Non-basic value(s)
alignment	start aligned	none
character orientation	0 degrees	90 degrees
character path	0, 90 degrees	270 degrees
character spacing	the equivalent of 120 BMUs	the equivalent of 80, 100, 120 BMUs
code extension announcers	any string of escape sequences in accordance with ISO 2022	none
graphic character sets	the escape sequences to designate, and the locking shift functions to invoke, the graphic character sets of ISO 6937-2	the escape sequences to designate, and any required locking shift functions to invoke, any other registered graphic character sets
graphic character subrepertoire	3	the identifier of any registered subrepertoire of ISO 6937
graphic rendition	0, 4	1, 3, 10-19, 22-24, 26, 50
initial offset	1) any non-negative integer 2) any non-negative integer	none none
line progression	270 degrees	90 degrees
line spacing	the equivalent of 100, 200, 300, 400 MBUs	the equivalent of 150 BMUs

The default value of these presentation attributes is as specified in this Recommendation except "graphic character subrepertoire".

It is assumed that all presentation attributes that are applicable to formatted character content architectures, and that are not specified in the above table, have values equal to the default values specified in this Recommendation.

B.3.3 Control functions

Control function	Basic value(s)	Non-basic value(s)
BS	not applicable	not applicable
CR	not applicable	not applicable
IGS	3	the identifier of any other registered subrepertoire of ISO 6937
LF	not applicable	not applicable
PLD	not applicable	not applicable
PLU	not applicable	not applicable
SGR	0, 4	1, 3, 10-19, 22-24, 26, 50
SHS	0	1, 2, 3
SP	not applicable	not applicable
SUB	not applicable	not applicable
SVS	0, 1, 2, 3	4

Any code extension control function defined in ISO 2022, within the scope of the value of the attribute "code extension announcers", is permitted.

The default value of these control function parameters is as specified in this Recommendation except IGS (default value = 3).

B.4 Character content architecture level CF-3

B.4.1 Type

CF-3 is a character content architecture level pertaining to the formatted character content architecture class.

B.4.2 Presentation attributes

Presentation attribute	Basic value(s)	Non-basic value(s)
alignment	start aligned end aligned centred justified	none
character fonts	1) none 2) none	any any
character orientation	0 degrees	90, 180, 270 degrees
character path	0, 90 degrees	180, 270 degrees
character spacing	the equivalent of 100, 120 BMUs	any other positive integer

Presentation attribute	Basic value(s)	Non-basic value(s)
code extension announcers	any string of escape sequences in accordance with ISO 2022	none
first line offset	any integer	none
formatting indicator	no, yes	none
graphic character sets	the escape sequences to designate, and the locking shift functions to invoke, the graphic character sets of ISO 6937-2	the escape sequences to designate, and any required locking shift functions to invoke, any other registered graphic character sets
graphic character subrepertoire	0	the identifier of any registered subrepertoire of ISO 6937
graphic rendition	0, 1, 3-4, 9, 10-19, 21-24, 29	2, 5-7, 25-27, 50
initial offset	1) any non-negative integer 2) any non-negative integer	none none
itemization	1) no itemization start-aligned end-aligned 2) any integer 3) any integer	none none none
kerning offset	1) any non-negative integer 2) any non-negative integer	none none
line layout table	1) any 2) any 3) start-aligned end-aligned centred aligned-around 4) any	none none none none
line progression	270 degrees	90 degrees
line spacing	the equivalent of 100, 200, 300, 400 BMUs	any other positive integer
pairwise kerning	yes, no	none

The default value of these presentation attributes is as specified in this Recommendation.

B.4.3 *Control functions*

Control function	Basic value(s)	Non-basic value(s)
BS	not applicable	not applicable
CR	not applicable	not applicable
GCC	0, 1, 2	none
HPB	any positive integer	none
HPR	any positive integer	none
IGS	0	the identifier of any registered subrepertoire of ISO 6937
JFY	0	none
LF	not applicable	not applicable
PLD	not applicable	not applicable
PLU	not applicable	not applicable
SACS	any non-negative integer	none
SCS	any positive integer	none
SGR	0, 1, 3-4, 10-19, 21-24, 29, 50	2, 5-7, 25-27, 50
SHS	0, 1	2, 3, 4
SLS	any positive integer	none
SP	not applicable	not applicable
SRCS	any non-negative integer	none
SRS	0, 1	none
SSW	any positive integer	none
STAB	any	none
SUB	not applicable	not applicable
SVS	0, 1, 2, 3	4, 9
VPB	any positive integer	none
VPR	any positive integer	none

Any code extension control function defined in ISO 2022, within the scope of the value of the attribute "code extension announcers", is permitted.

The default value of these control function parameters is as specified in this Recommendation.

B.5   *Character content architecture level CP-0*

B.5.1   *Type*

CP-3 is a character content architecture level pertaining to the processable character content architecture class.

B.5.2   *Presentation attributes*

Presentation attribute	Basic value(s)	Non-basic value(s)
alignment	start aligned centred justified	none
code extension announcers	any string of escape sequences in accordance with ISO 2022	none
graphic rendition	0, 4	none

The default value of these presentation attributes is as specified in this Recommendation.

It is assumed that all presentation attributes that are applicable to processable character content architectures, and that are not specified in the above table, have values equal to the default values specified in this Recommendation except:

- "graphic character subrepertoire": default value = 3.

B.5.3   *Control functions*

Control function	Basic value(s)	Non-basic value(s)
CR	not applicable	not applicable
LF	not applicable	not applicable
SUB	not applicable	not applicable

Any code extension control function defined in ISO 2022, within the scope of the value of the attribute "code extension announcers", is permitted.

B.6   *Character content architecture level CP-2*

B.6.1   *Type*

CP-2 is a character content architecture level pertaining to the processable character content architecture class.

B.6.2   *Presentation attributes*   ,

Presentation attribute	Basic value(s)	Non-basic value(s)
alignment	start aligned end aligned centred justified	none
character orientation	0 degrees	90 degrees
character path	0, 90 degrees	270 degrees

Presentation attribute	Basic value(s)	Non-basic value(s)
character spacing	the equivalent of 120 BMUs	the equivalent of 80, 100, 120 BMUs
code extension announcers	any string of escape sequences in accordance with ISO 2022	none
graphic character sets	the escape sequences to designate, and the locking shift functions to invoke, the graphic character sets of ISO 6937-2	the escape sequences to designate, and any required locking shift functions to invoke, any other registered graphic character sets
graphic character subrepertoire	3	the identifier of any registered subrepertoire of ISO 6937
graphic rendition	0, 4	1, 3, 10-19, 22-24, 26, 50
indentation	any non-negative integer	none

line layout table	1) any
	2) any
	3) start-aligned
	   end-aligned
	   centred
	   aligned-around
	4) any
		none
		none
		none
		
		none

line progression	270 degrees	none
line spacing	the equivalent of 100, 200, 300, 400 BMUs	the equivalent of 150 BMUs

The default value of these presentation attributes is as specified in this Recommendation except "graphic character subrepertoire".

It is assumed that all presentation attributes that are applicable to processable character content architectures, and that are not specified in the above table, have values equal to the default values specified in this Recommendation.

B.6.3 Control functions

Control function	Basic value(s)	Non-basic value(s)
BPH	not applicable	not applicable
CR	not applicable	not applicable
IGS	3	the identifier of any other registered subrepertoire of ISO 6937
LF	not applicable	not applicable
NBH	not applicable	not applicable
PLD	not applicable	not applicable
PLU	not applicable	not applicable
SGR	0, 4	1, 3, 10-19, 22-24, 26, 50
SHS	0	1, 2, 3
SP	not applicable	not applicable
STAB	any	none
SUB	not applicable	not applicable
SVS	0, 1, 2, 3	4, 9

Any code extension control function defined in ISO 2022, within the scope of the value of the attribute "code extension announcers", is permitted.

The default value of these control function parameters is as specified in this Recommendation except IGS (default value = 3).

B.7 Character content architecture level CP-3

B.7.1 Type

CP-3 is a character content architecture level pertaining to the processable character content architecture class.

B.7.2 Presentation attributes

Presentation attribute	Basic value(s)	Non-basic value(s)
alignment	start aligned end aligned centred justified	none
character fonts	1) none 2) none	any any
character orientation	0 degrees	90, 180, 270 degrees
character path	0, 90 degrees	180, 270 degrees

Presentation attribute	Basic value(s)	Non-basic value(s)
character spacing	the equivalent of 100, 120 BMUs	any other positive integer
code extension announcers	any string of escape sequences in accordance with ISO 2022	none
graphic character sets	the escape sequences to designate, and the locking shift functions to invoke, the graphic character sets of ISO 6937-2	the escape sequences to designate, and any required locking shift functions to invoke, any other registered graphic character sets
graphic character subrepertoire	0	the identifier of any registered subrepertoire of ISO 6937
graphic rendition	0, 1, 3-4, 9, 10-19, 21-24, 29	2, 5-7, 25-27, 50
first line offset	any integer	none
indentation	any non-negative integer	none

itemization	1) no itemization
	    start-aligned
	    end-aligned
	2) any integer
	3) any integer
		none
		
		none
		none

kerning offset	1) any non-negative integer 2) any non-negative integer	none none

line layout table	1) any
	2) any
	3) start-aligned
	    end-aligned
	    centred
	    aligned-around
	4) any
		none
		none
		none
		
		none

line progression	270 degrees	90 degrees
line spacing	the equivalent of 100, 200, 300, 400 BMUs	any other positive integer
orphan size	any positive integer	none
pairwise kerning	yes, no	none
proportional line spacing	yes, no	none
widow size	any positive integer	none

The default value of these presentation attributes is as specified in this Recommendation.

B.7.3 *Control functions*

Control function	Basic value(s)	Non-basic value(s)
BPH	not applicable	not applicable
CR	not applicable	not applicable
GCC	0, 1, 2	none
IGS	0	the identifier of any registered subrepertoire of ISO 6937
LF	not applicable	not applicable
NBH	not applicable	not applicable
PLD	not applicable	not applicable
PLU	not applicable	not applicable
PTX	0, 1, 3	none
SCS	any positive integer	none
SGR	0, 1, 3-4, 9, 10-19, 21-24, 29	2, 5-7, 25-27, 50
SHS	0, 1	2, 3, 4
SLS	any positive integer	none
SP	not applicable	not applicable
SRS	0, 1	none
STAB	any	none
SUB	not applicable	not applicable
SVS	0, 1, 2, 3	4, 9
VPB	any positive integer	none
VPR	any positive integer	none

Any code extension control function defined in ISO 2022, within the scope of the value of the attribute "code extension announcers", is permitted.

The default value of these control function parameters is as specified in this Recommendation.

B.8 *Character content architecture level CFP-2*

B.8.1 *Type*

CFP-2 is a character content architecture level pertaining to the formatted processable character content architecture class.

B.8.2 *Presentation attributes*

Presentation attribute	Basic value(s)	Non-basic value(s)
alignment	start aligned end aligned centred justified	none
character orientation	0 degrees	90 degrees
character path	0, 90 degrees	270 degrees
character spacing	the equivalent of 120 BMUs	the equivalent of 80, 100, 120 BMUs
code extension announcers	any string of escape sequences in accordance with ISO 2022	none
graphic character sets	the escape sequences to designate, and the locking shift functions to invoke, the graphic character sets of ISO 6937-2	the escape sequences to designate, and any required locking shift functions to invoke, any other registered graphic character sets
graphic character subrepertoire	3	the identifier of any registered subrepertoire of ISO 6937
graphic rendition	0, 4	1, 3, 10-19, 22-24, 26, 50
initial offset	1) any non-negative integer 2) any non-negative integer	none none
indentation	any non-negative integer	none

line layout table	1) any
	2) any
	3) start-aligned
	    end-aligned
	    centred
	    aligned-around
	4) any
		none
		none
		none
		
		none

line progression	270 degrees	none
line spacing	the equivalent of 100, 200, 300, 400 BMUs	the equivalent of 150 BMUs

The default value of these presentation attributes is as specified in this Recommendation except "graphic character subrepertoire".

It is assumed that all presentation attributes that are applicable to formatted processable character content architectures, and that are not specified in the above table, have values equal to the default values specified in this Recommendation except:

- graphic character subrepertoire: default value = 3.

B.8.3 *Control functions*

Control function	Basic value(s)	Non-basic value(s)
BPH	not applicable	not applicable
BS	not applicable	not applicable
CR	not applicable	not applicable
IGS	3	the identifier of any other registered subrepertoire of ISO 6937
LF	not applicable	not applicable
NBH	not applicable	not applicable
PLD	not applicable	not applicable
PLU	not applicable	not applicable
SGR	0, 4	1, 3, 10-19, 22-24, 26, 50
SHS	0	1, 2, 3
SOS	not applicable	not applicable
SP	not applicable	not applicable
ST	not applicable	not applicable
STAB	any	none
SUB	not applicable	not applicable
SVS	0, 1, 2, 3	4, 9

Any code extension control function defined in ISO 2022, within the scope of the value of the attribute "code extension announcers", is permitted.

The default value of these control function parameters are as specified in this Recommendation except IGS (default value = 3).

B.9 *Character content architecture level CFP-3*

B.9.1 *Type*

CFP-3 is a character content architecture level pertaining to the formatted processable character content architecture class.

B.9.2 *Presentation attributes*

Presentation attribute	Basic value(s)	Non-basic value(s)
alignment	start aligned end aligned centred justified	none
character fonts	1) none 2) none	any any
character orientation	0 degrees	90, 180, 270 degrees
character path	0, 90 degrees	180, 270 degrees
character spacing	the equivalent of 100, 120 BMUs	any other positive integer
code extension announcers	any string of escape sequences in accordance with ISO 2022	none
formatting indicator	no, yes	none
graphic character sets	the escape sequences to designate, and the locking shift functions to invoke, the graphic character sets of ISO 6937-2	the escape sequences to designate, and any required locking shift functions to invoke, any other registered graphic character sets
graphic character subrepertoire	0	the identifier of any registered subrepertoire of ISO 6937
graphic rendition	0, 1, 3-4, 9, 10-19, 21-24, 29	2, 5-7, 25-27, 50
first line offset	any integer	none
initial offset	1) any non-negative integer 2) any non-negative integer	none none
indentation	any non-negative integer	none
itemization	1) no itemization start-aligned end-aligned 2) any integer 3) any integer	none none none

Presentation attribute	Basic value(s)	Non-basic value(s)
kerning offset	1) any non-negative integer 2) any non-negative integer	none none

line layout table	1) any
	2) any
	3) start-aligned
	   end-aligned
	   centred
	   aligned-around
	4) any
		none
		none
		none
		
		none

line progression	270 degrees	90 degrees
line spacing	the equivalent of 100, 200, 300, 400 BMUs	any other positive integer
orphan size	any positive integer	none
pairwise kerning	yes, no	none
proportional line spacing	yes, no	none
widow size	any positive integer	none

B.9.3 *Control functions*

Control function	Basic value(s)	Non-basic value(s)
BPH	not applicable	not applicable
BS	not applicable	not applicable
CR	not applicable	not applicable
GCC	0, 1, 2	none
HPB	any positive integer	none
HPR	any positive integer	none
IGS	0	the identifier of any registered subrepertoire of ISO 6937
JFY	0	none
LF	not applicable	not applicable

Control function	Basic value(s)	Non-basic value(s)
NBH	not applicable	not applicable
PLD	not applicable	not applicable
PLU	not applicable	not applicable
PTX	0, 1, 3	none
SACS	any non-negative integer	none
SCS	any positive integer	none
SGR	0, 1, 3-4, 10-19, 21-24, 29	2, 5-7, 25-27, 50
SHS	0, 1	2, 3, 4
SLS	any positive integer	none
SOS	not applicable	not applicable
SP	not applicable	not applicable
SRCS	any non-negative integer	none
SRS	0, 1	none
SSW	any positive integer	none
ST	not applicable	not applicable
STAB	any	none
SUB	not applicable	not applicable
SVS	0, 1, 2, 3	4, 9
VPB	any positive integer	none
VPR	any positive integer	none

Any code extension control function defined in ISO 2022, within the scope of the value of the attribute "code extension announcers", is permitted.

ANNEX C

(to Recommendation T.416)

Coded representations of control functions

(Informative)

Coded representations of control functions are defined in ISO 6429. A summary of the coded representations of the control functions defined in this Recommendation is given below.

In this table, CSI denotes the Control Sequence Introducer represented by the bit-combination 09/11 and Pn and Ps denote respectively numeric and selective parameter values represented by one or more bit combinations in the range 03/00 to 03/09.

This coded representation of a control function with parameters but with Pn or Ps omitted represents that control function with the default value of the parameter.

Control function		Coded representation
BPH	(Break Permitted Here)	08/02
BS	(Backspace)	00/08
CR	(Carriage Return)	00/13
HPB	(Character Position Backward)	CSI Pn 06/10
HPR	(Character Position Relative)	CSI Pn 06/01
GCC	(Graphic Character Composition)	CSI Ps 02/00 05/15
IGS	(Identify Graphic Subrepertoire)	CSI Ps 02/00 04/13
JFY	(No Justify)	CSI 02/00 04/06
LF	(Line Feed)	00/10
NBH	(No Break Here)	08/03
PLD	(Partial Line Down)	08/11
PLU	(Partial Line Up)	08/12
PTX	(Parallel Texts)	CSI Ps 05/12
SCS	(Set Character Spacing)	CSI Pn 02/00 06/07
SGR	(Select Graphic Rendition)	CSI Ps... 06/13
SHS	(Select Character Spacing	CSI Ps 02/00 04/11
SACS	(Set Additional Character Spacing)	CSI Pn 02/00 05/12
SLS	(Set Line Spacing)	CSI Pn 02/00 06/08
SOS	(Start Of String)	09/08
SRCS	(Set Reduced Character Spacing)	CSI Pn 02/00 06/06
SSW	(Set SPACE Width)	CSI Pn 02/00 05/11
SP	(SPACE)	02/00
SRS	(Start Reverse String)	CSI Ps 05/11
ST	(String Terminator)	09/12
STAB	(Selective Tabulation)	CSI Ps 02/00 04/12
SUB	(Substitute)	01/10
SVS	(Select Line Spacing	CSI Ps 02/00 04/12
VPB	(Line Position Backward)	CSI Pn 06/11
VPR	(Line Position Relative)	CSI Pn 06/05

ANNEX D

(to Recommendation T.416)

Summary of object identifiers

(Informative)

Values of ASN.1 object identifiers are assigned in various sections in this Recommendation. These are summarised below:

Object identifier value	Meaning	Section
{ 2 8 1 6 2 }	Identifies module; character presentation attributes	9.2
{ 2 8 1 6 3 }	Identifies module; character coding attributes	9.3
{ 2 8 1 6 4 }	Identifies module; character profile attributes	9.4
{ 2 8 2 6 0 }	Identifies formatted character; content architecture class	7.4
{ 2 8 2 6 1 }	Identifies processable character; content architecture class	7.4
{ 2 8 2 6 2 }	Identifies formatted processable; character content architecture class	7.4
{ 2 8 3 6 0 }	Identifies ISO 2022 type of coding	8.1

OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT –

RASTER GRAPHICS CONTENT ARCHITECTURES[1)]

CONTENTS

[1)] This text is aligned with the final text of the corresponding International Standard ISO 8613-7.

1   Scope

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 graphic elements and raster graphic elements, all potentially within one document.

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

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.

1.3   This Recommendation defines:

-      the raster graphics content architectures that can be used in conjunction with the document architecture defined in Recommendation T.412;

-      the internal structure of content portions that are structured according to a raster graphics content architecture;

-      those aspects of positioning and imaging applicable to the presentation of raster graphics contents in a basic layout object;

- a content layout process which, together with the document layout process defined in Recommendation T.412, specifies the method for determining the dimensions of basic layout objects for raster graphics content portions;

- the presentation and content portion attributes applicable to raster graphics content architectures.

2   Normative references

The following Recommendations and International Standards contain provisions which, through references in this text, constitute provisions of the T.410-Series of Recommendations. At the time of publication the editions indicated were valid. All standards are subject to revision, and parties to agreements based upon the T.410-Series of Recommendations are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of ISO and IEC maintain registers of currently valid International Standards.

- Rec. T.4 (1988): Standardization of group 3 facsimile apparatus for document transmission.

- Rec. T.6 (1988): Facsimile coding schemes and coding control functions for group 4 facsimile apparatus.

- Rec. X.208 (1988): Specification of abstract syntax notation one (ASN.1).

3   Definitions

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

4   General principles

4.1   *Content architectures*

This Recommendation defines two classes of raster graphics content architectures:

- formatted raster graphics content architecture class, which allows for document content to be presented as intended by the originator. Formatted form content can only be associated with basic components;

- formatted processable raster graphics content architecture class, which allows for document content to be processed and also to be presented as intended by the originator. Formatted processable content can be associated with any basic logical component.

4.1.1   *Formatted content architecture class*

Formatted raster graphics content is intended to be laid out, or imaged by the recipient in accordance with the originator's intent. It is not intended to be reformatted. This form of content may only be used in formatted form documents.

For this form of content, all the necessary information for positioning of pels has been specified. The method of positioning is specified in Section 5.

A particular feature of this form of content is that the position of the pel array can be offset relative to the position of the basic layout object. As a result, it is possible that not all of the area of the basic layout object is utilized for positioning pels. Also a portion of the pel array may be positioned such that it is outside the basic layout object. Such a portion, if any, is not imaged.

4.1.2 Formatted processable content architecture class

Formatted processable raster graphics content is intended to be laid out, reformatted or imaged by the recipient in accordance with the originator's intent. This form of content may be used in formatted processable and formatted processable form documents.

The originator may, when using this form of content, specify the precise requirements for the layout and imaging of the pel array. Alternatively, the originator may specify various constraints concerning the intended layout and imaging of the pel array, i.e. the precise requirements are not specified and the layout is determined by the content layout process performed by the recipient.

When the precise requirements for the layout are specified, the fixed dimension layout method is used to layout and image the content. Otherwise, the content is laid out and imaged using the scalable dimension layout method. These layout methods are defined in Section 10.

A particular feature of these layout methods is that in both cases the content is laid out such that the entire basic layout object is utilized. In addition it is possible to specify that only a portion of the pel array is to be laid out.

4.2 Content

The content of a basic component that conforms to a raster graphics content architecture represents a two-dimensional pictorial image in the form of a rectangular two-dimensional array of *picture elements (pels)*.

Each element of a *pel array* comprises data used to determine the image of the corresponding pel.

Each basic component contains exactly one content portion.

The data which determines the image of a pel specifies one of two states, named "set" and "unset". The set state is used to identify the foreground colour and the unset to identify the background colour. The representation of foreground and background within an image is not defined by this Recommendation.

Note 1 - For reproduction on paper, the background colour will normally be the colour of the paper, for instance white, and the foreground colour a contrasting colour, for instance black.

Note 2 - A future version of this Recommendation may allow specification of more information for each pel, enabling the representation of multi-colour images.

4.3 Presentation attributes

Presentation attributes are applicable to basic components and specify information for laying out and imaging the content of the basic component, and are defined in Section 6. This information cannot be modified within the content of the basic component to which it applies.

4.4 Content portion attributes

Content attributes are applicable to content portions and specify information related to the identification and coding of the content. They are also used in laying out and imaging the content of the content portion. Content portion attributes are defined in Section 7.

4.5 Coding of content information

The methods of encoding the pel array in a content portion structured according to a raster graphics content architecture are specified in Section 9.

4.6 Picture element (pel) array

The picture elements in an array have a defined order. The array consists of an ordered sequence of rows of picture elements. Each row in the array contains the same number of picture elements and consists of an ordered sequence of picture elements that represents a line of the image.

5   Principles of positioning pels

The methods of positioning pels within a basic layout object are described in this section. One of these applies to content portions which pertain to the formatted form content architecture. The other applies to content portions which pertain to the formatted processable form content architecture.

The general principles of positioning that apply to both these methods are described in §§ 5.4.1, 5.4.2 and 5.4.3 then describes the specific principles that apply to the formatted and formatted processable forms of content.

A basic logical component with a formatted processable form content architecture class must undergo the content layout process before it can be positioned and imaged. The content layout process (defined in Section 10) determines the block size into which the content portion is to be imaged. The content is then positioned in accordance with the positioning rules for content pertaining to the formatted processable form content architecture class.

Any parts of a raster graphics content portion which extend beyond the boundaries of the basic layout object are not imaged.

5.1   *Basic concepts*

5.1.1   *Measurement units and directions*

For raster graphics content, the unit for positioning pels is the Scaled Measurement Unit (SMU).

The SMU is derived from the Basic Measurement Unit (BMU) by multiplying the BMU with a factor which is specified by the attribute "unit scaling" (defined in Recommendation T.414). The BMU and SMU are defined in Recommendation T.412 (§§ 3.3.4.1 and 3.3.4.2 respectively).

All directions are expressed as counter clockwise angles of rotation relative to some specified reference direction (as illustrated in Figure 1/T.417).
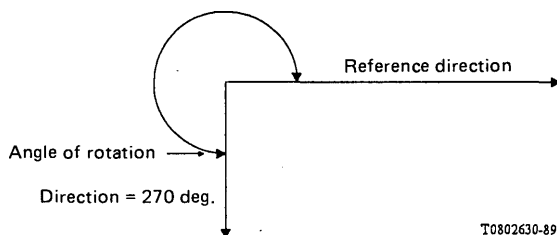
FIGURE 1/T.417

Example of direction

5.1.2   *Coordinate systems*

Two rectangular coordinate systems are used in the positioning of pels.

One system is a dimensionless coordinate system used to identify the pels that constitute a *clipped pel array* (defined in § 5.3.1). In this system, the origin of the coordinate system is positioned at the first pel in the pel array. One axis is in the direction of the pels in each row of pels. The second axis is in the direction of the columns of pels. This system uses non-negative dimensionless integer values and coordinate pairs are denoted using upper case letters.

The second system is used for the positioning of pels associated with basic layout objects. In this system, one axis is parallel to the horizontal axis of the page coordinate system (defined in Recommendation T.412) and the other axis is in a direction 270 degrees relative to the horizontal axis. This system uses rational values in scaled measurements units (SMUs) to identify points or specify lengths within a basic layout object. Coordinate pairs are indicated in lower case letters.

5.2 *Pel image model*

Each pel is associated with a *reference area*. The side of the reference are along the direction of the pel path equals the pel spacing and the side along the direction of line progression equals the line spacing.

Each reference area has a *reference point*, which is used for positioning the pel. The reference point is defined as the corner of the reference area situated in the opposite direction of both pel path and line progression. The position of a pel in a basic layout object is defined as the position of reference point of the reference area of that pel.

Note - The position of the image of the pel relative to the reference area is implementation dependent, but it is the intention that the main part of the image of pel is positioned within the reference area.

5.3 *Positioning of pels*

In general, when positioning (and subsequently imaging) the content of a content portion in relation to a basic layout object, only part of the content is considered. Two methods of selecting the required part of the content are provided:

- specification of a clipped pel array;

- discarding of pels.

5.3.1 *The clipped pel array*

The clipped pel array is a rectangular array of pels defined by two coordinate pairs in the dimensionless coordinate system. The diagonally opposite pairs of the clipped pel array are identified by the coordinate pairs (XI, Y1) and (X2, Y2) where X1 <= X2 and Y1 <= Y2. Figure 2/T.417 illustrates the clipping of a content portion.
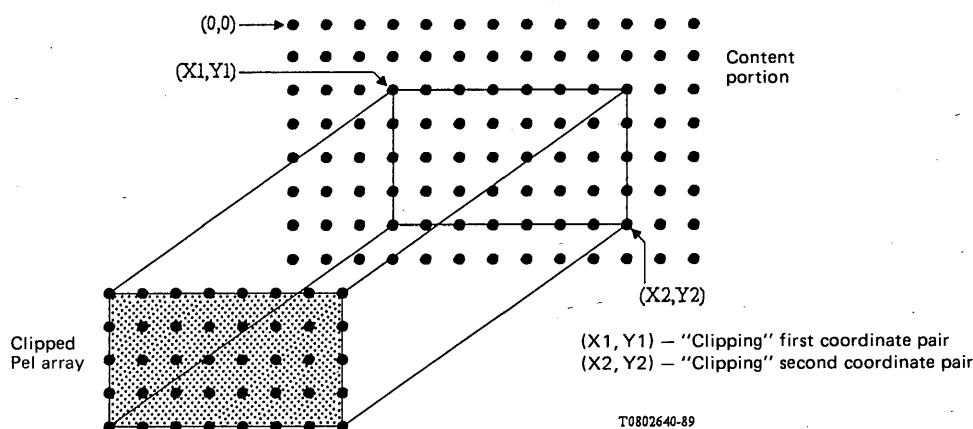
FIGURE 2/T.417

Example of clipping a content portion

5.3.2   *Discarded pels*

In the formatted raster graphics content architecture class, the number of pels to be discarded at the beginning and the end of each line of pels can be specified by a coding attribute.

5.4   *Positioning of pels in a basic layout object*

5.4.1   *Positioning parameters*

The positioning of pels within a basic layout object is determined by the following parameters (illustrated in Figure 3/T.417):

- initial point;

- pel path;

- line progression;

- pel spacing;

- line spacing.

The values of these parameters remain constant within the content associated with a particular basic layout object.

The general use of these parameters when positioning pels is described below and illustrated in Figure 3/T.417. The particular applicability of these parameters to formatted and formatted processable form content is described in §§ 5.4.2 and 5.4.3 respectively.
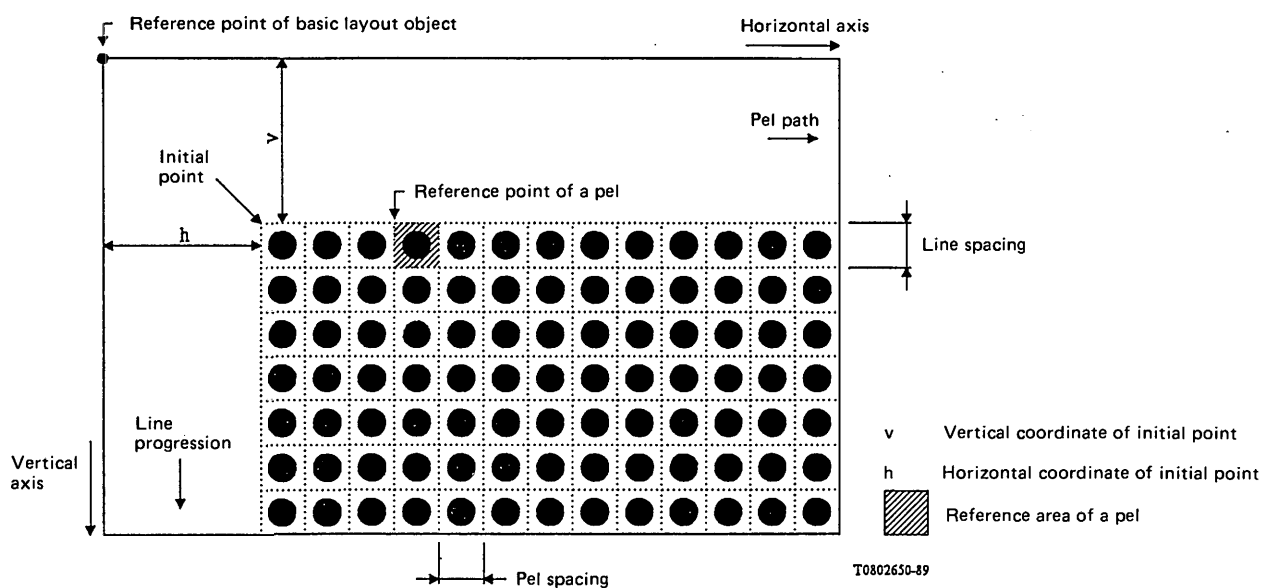
FIGURE 3/T.417

**Positioning of pels of the clipped pel
array within a basic layout object**

Note - In the following text, various sub-section titles have been omitted.

The *initial point* is the point relative to which all pels are positioned within a basic layout object.

The value of the initial point is a coordinate pair (x, y), where x and y are the horizontal and vertical distances respectively, of the initial point from the reference point of the basic layout object.

The *pel path* is the direction of progression of successive pels along a line and is expressed as a direction relative to the horizontal axis of the page coordinate system (as defined in Recommendation T.412).

Line progression is the direction of progression of successive lines and is expressed as a direction relative to the pel path.

Lines of pels are positioned such that the first pel to be positioned on each line falls on an imaginary line through the initial point in the direction of line progression.

The *pel spacing* is the distance between two adjacent pels along a line, in the direction of pel path.

The *line spacing* is the distance between two adjacent lines of pels. The line spacing may be less than, greater than or equal to the pel spacing.

The *spacing ratio* is defined as the ratio of the line spacing to pel spacing.

The *aspect ratio* of a clipped pel array that has been positioned in a basic layout object is defined as the ratio of the dimension of the pel array in the direction of pel path to the dimension in the direction of line progression.

The first pel of the clipped pel array is positioned at the initial point.

Each pel on first line is positioned along a line through the initial point in the direction of the pel path.

The first pel of each line is positioned along a line through the initial point in the direction of line progression.

5.4.2 *Positioning rules for formatted form content*

For this form of content, the positioning parameters are explicitly specified by applicable presentation attributes (see Section 6).

It is not possible to define a clipped pel array when using this form of content. However, a coding attribute can be used to indicate that a specified number of pels are to be discarded at the beginning and end of each line of the content portion. In this case, only the remaining pels in the content position are considered for positioning.

The line spacing and pel spacing are both specified by the same presentation attribute, and take the same value form the limited set of values specified in § 6.2.2.

The initial point can be positioned anywhere inside or outside of the basic layout object. Its default position (see § 6.2.1) is the corner of the basic layout object in the direction opposite of both pel path and line progression.

All the pels within a content portion are to be considered for positioning (apart from any pels that are to be discarded). However, pels that would be positioned outside of the basic layout object are not to be imaged by the imaging process.

5.4.3 *Positioning rules for formatted processable content*

For this form of content, the positioning parameters are determined from information specified in presentation and coding attributes and from the dimensions of the basic layout object.

The clipped pel array is specified by a presentation attribute, which selects the portion of the content portion to be positioned.

The pel path and line progression are explicitly specified by separate presentation attributes. The initial point is determined from the pel path and line progression specified, such that it is situated in the corner of the basic layout object in the opposite direction of both pel path and line progression (see § 6.2.1); other values for the initial point cannot be specified.

The pel spacing is set to be equal to the dimensions of the basic layout object in the direction of the pel path divided by the number of pels per line in the clipped pel array. Similarly, the line spacing is set to be equal to the dimensions of the basic layout object in the direction of line progression divided by the number of lines in the clipped pel array.

Thus the clipped pel array is positioned within the basic layout object such that the reference areas of all the pels completely fill the basic layout object. None of the pels in the clipped pel array can be positioned outside of the basic layout object.

6 Definition of raster graphics presentation attributes

Presentation attributes specify the initial conditions relating to the layout and imaging of the content of a basic component. They may be specified for basic logical and layout components and for presentation styles.

There are three categories of raster graphics presentation attributes:

- logical presentation attributes which take effect during the content layout process, but are ignored during the content imaging process;

- layout presentation attributes which take effect during the content imaging process. Their values are generated either by the content layout process, or by a process that creates or edits the content;

- shared presentation attributes which take effect during either or both the content layout and imaging process.

These attributes are listed in Table 1/T.417.

For each presentation attribute a default value is defined. This value is used in the defaulting mechanism as defined in Recommendation T.412.

This section also defines values for the content architecture attributes specific to raster graphics content architectures. These attributes are defined in Recommendation T.412.

TABLE 1/T.417

Raster graphics presentation attributes

Shared attributes	Layout attributes
Pel path Line progression Clipping	Pel transmission density Initial offset
	Logical attributes
	Pel spacing Spacing ratio Image dimensions

6.1   *Shared presentation attributes*

6.1.1   **clipping**

CATEGORY:	Shared	
APPLICABILITY:	Formatted processable content architecture class	
STRUCTURE:	First coordinate pair:	X coordinate, Y coordinate
	Second coordinate pair:	X coordinate, Y coordinate
PERMISSIBLE VALUES:	First coordinate pair:	non-negative integer, non-negative integer
	Second coordinate pair:	non-negative integer, non-negative integer
DEFAULT VALUES:	First coordinate:	(0,0)
	Second coordinate	(N-1, L-1) where:

N is the number of pels per line,
L is the number of lines

DEFINITION:

This attribute determines the subregion of the pel array, as described by the content portion, which is to be considered by the content layout process and the content imaging process.

This attribute consists of two coordinate pairs. The first pair specifies the first pel that is part of the selected array. The second pair specifies the last pel that is part of the selected array.

REMARKS:

Each coordinate of the first pair must be less than or equal to the corresponding coordinate of the second pair.

6.1.2   **line progression**

CATEGORY:	Shared
APPLICABILITY:	Formatted and formatted processable content architecture classes
PERMISSIBLE VALUES:	90, 270 degrees
DEFAULT VALUES:	270 degrees

DEFINITION:

This attribute specifies the direction of the progression of successive lines, relative to the pel path.

6.1.3   **pel path**

CATEGORY:	Shared
APPLICABILITY:	Formatted and formatted processable content architecture classes
PERMISSIBLE VALUES:	0, 90, 180, 270 degrees
DEFAULT VALUES:	0 degrees

DEFINITION:

This attribute specifies the direction of the progression of successive pels along a line, relative to the horizontal axis of the basic layout object.

6.2 *Layout presentation attributes*

6.2.1 **initial offset**

CATEGORY:	Layout
APPLICABILITY:	Formatted content architecture class
STRUCTURE:	Two parameters: horizontal coordinate, vertical coordinate
PERMISSIBLE VALUES:	Horizontal coordinate: any integer Vertical coordinate: any integer

DEFAULT VALUES: The default value of this attribute depends on the pel path and line progression as defined in Table 2/T.417.

DEFINITION:

This attribute specifies the position of the initial point relative to the basic layout object.

The parameters "horizontal coordinate" and "vertical coordinate" specify the horizontal and vertical coordinates, in SMUs, of the initial point relative to the reference point of the basic layout object. The value of each coordinate may be positive, zero or negative; if either or both coordinates are negative then the initial point will be outside the basic layout object.

Note - The facility to specify negative coordinate values for the initial point is intended only for use with content architectures based on CCITT Recommendation T.73, such as RF-1 (see Annex B), which provide no other clipping mechanism.

TABLE 2/T.417

Default values of the presentation attribute "initial offset"
(position of initial point)

Pel path	Line progression	Horizontal coordinate	Vertical coordinate
0	270	0	0
	90	0	BDV
270	270	BDH	0
	90	0	0
180	270	BDH	BDV
	90	BDH	0
90	270	0	BDV
	90	BDH	BDV

Note - The notation used in this table is:
BDV = vertical dimension of block;
BDH = horizontal dimension of block.

6.2.2 **pel transmission density**

CATEGORY:	Layout
APPLICABILITY:	Formatted content architecture class
PERMISSIBLE VALUES:	1, 2, 3, 4, 5, 6, BMU
DEFAULT VALUES:	6 BMU

DEFINITION:

This attribute specifies a single value for both the pel spacing and line spacing.

Note - The correspondence between pel spacing, line spacing and resolution is:

Pel spacing and line spacing in BMU	Resolution in number of pels per 1200 BMU
6	200
5	240
4	300
3	400
2	600
1	1200

6.3 *Logical presentation attributes*

6.3.1 image dimensions

CATEGORY:	Logical
APPLICABILITY:	Formatted processable content architecture class
STRUCTURE:	One of four parameters:

a) "width controlled" with sub-parameters:

-   "minimum width",
-   "preferred height"

b) "height controlled" with sub-parameters:

-   "minimum height:,
-   "preferred height"

c) "area controlled" with sub-parameters:

-   "minimum width",
-   "preferred width",
-   "minimum height",
-   "preferred height",
-   "aspect ratio flag"

d) "automatic" with no sub-parameters.

PERMISSIBLE VALUES:	"minimum width":	non-negative integer,
	"preferred width":	non-negative integer, .
	"minimum height":	non-negative integer,
	"preferred height":	non-negative integer,
	"aspect ratio flag":	'fixed', 'variable'
	"automatic":	'null'
DEFAULT VALUES:	"automatic".	

DEFINITION:

This attribute specifies the intended dimensions of the basic layout object that is to contain the clipped pel array.

The values of "minimum width" and preferred width" specify, respectively, the lower limit and the upper limit of the allowed dimensions of the basic layout object in the direction of the pel path. The value of the "minimum width" shall not to be greater than the value of the "preferred width".

The values of "minimum height" and "preferred height" specify, respectively, the lower limit and the upper limit of the dimensions of the basic layout object in the direction of the line progression. The value of the "minimum height" shall not to be greater than the value of the "preferred height".

If either or both of the values of the preferred parameters are specified, the corresponding dimensions of the basic layout object are required to be as close to these values as possible.

If only the width is specified (case a), this attribute specifies that the height shall be such that the aspect ratio of the clipped pel array is maintained.

If only the range of allowed heights is specified (case b), this attribute specifies that the width shall be such that the aspect ratio of the clipped pel array is maintained.

If both the ranges of allowed widths and heights are specified (case c) the value of "aspect ratio flag" determines whether or not the aspect ratio of the clipped pel array shall be maintained during the determination of the dimensions of the basic layout object.

If neither the range of allowed heights nor the range of allowed widths is specified (case d), this attribute specifies that the aspect ratio of the basic layout object shall be the same as the aspect ration of the clipped pel array, and also that the dimension of the basic layout object in the direction of the pel path shall be equal to the dimension of the available area in that direction.

All parameters specifying a width or height are specified in SMUs.

6.3.2 pel spacing

CATEGORY:	Logical	
APPLICABILITY:	Formatted processable content architecture class	
STRUCTURE:	Two parameters: or the value:	"length" "pel spaces" 'null'
PERMISSIBLE VALUES:	"length": "pel spaces":	positive integer positive integer
DEFAULT VALUES:	"length": "pel spaces":	4 1

DEFINITION:

This attribute specifies the method for determining the distance between successive pels along a line. The attribute consists of either 'null', or the two parameters "length" (with integer value m) and "pel spaces" (with integer value n).

If the attribute takes a value of 'null' the scalable dimension content layout method is followed.

If the attribute consists of the two parameters, the ratio of the integers m and n (m/n) specifies the spacing in SMUs between two successive pels, and the fixed dimension content layout method is followed.

REMARKS:

The scalable and fixed dimension content layout methods are described in Section 10.

6.3.3 spacing ratio

CATEGORY:	Logical	
APPLICABILITY:	Formatted processable content architecture class	
STRUCTURE:	Two parameters:	"line spacing value" "pel spacing value"
PERMISSIBLE VALUES:	"line spacing value": "pel spacing value":	positive integer positive integer

DEFAULT VALUES: "line spacing value":   1
                                 "pel spacing value":   1

DEFINITION:

This attribute specifies the ratio between the line spacing and the pel spacing of the image represented by the content portion. This ratio is to be observed by the raster graphics content layout process (defined in Section 10) in determining the block size, and by the imaging process (defined in Section 11) to avoid image distortion.

The value of this attribute consists of two parameters, the "line spacing value" and the "pel spacing value", the ratio of which equals the ratio of the line spacing to be pel spacing.

REMARKS:

The attribute "spacing ratio" one has an effect when the value of the parameter "aspect ratio flag" in the attribute "image dimensions" is set to 'fixed'.

6.4 Content architecture class attributes

6.4.1 Content architecture class

The value of the attribute "content architecture class" of a basic component description that conforms to this Recommendation is an ASN.1 object identifier with one of the following values:

{ 2 8 2 7 0 } for the formatted content architecture class;

{ 2 8 2 7 2 } for the formatted processable content architecture class.

6.4.2 Content type

The formatted content architecture class can be specified by the attribute "content type" with the value 1.

Note – Use of the attribute "content type" as an alternative to use of the content architecture class attribute "content architecture class" is permitted only for compatibility with Recommendation T.73 (1984).

7 Definition of raster graphics content portion attributes

According to Recommendation T.412, content portion attributes consist of four categories:

- identification attributes;
- common coding attributes;
- coding attributes;
- content information attributes.

The identification attributes are completely defined in Recommendation T.412.

The common coding attributes are described in Recommendation T.412; attribute values that are specific to raster graphics content architectures are specified in § 7.1.

Coding attributes are defined in § 7.2 and the format of the content information, i.e. the possible values of content information attributes, is specified in § 7.3.

7.1 Common coding attributes

7.1.1 type of coding

CLASSIFICATION: Defaultable

APPLICABILITY: Formatted and formatted processable content architecture class

STRUCTURE: ASN.1 object identifier or non-negative integer

PERMISSIBLE VALUES: ASN.1 object identifier:

{ 2 8 3 7 0 } for 'Rec. T.6 encoding',
{ 2 8 3 7 1 } for 'Rec. T.4 one dimensional encoding',
{ 2 8 3 7 2 } for 'Rec. T.4 two dimensional encoding',
{ 2 8 3 7 3 } for 'bitmap encoding'.
non-negative integer:
0 for 'Rec. T.6 encoding';

DEFAULT VALUE: 'Rec. T.6 encoding'

DEFINITION:

For the raster graphics content architectures, the possible values of this attribute are:

-   'Rec. T.6 encoding', according to the two dimensional encoding scheme defined in Recommendation T.6;

-   'Rec. T.4 one dimensional encoding', according to the one dimensional encoding scheme defined in Recommendation T.4;

-   'Rec. T.4 two dimensional encoding', according to the two dimensional encoding scheme defined in Recommendation T.4;

-   'bitmap encoding'.

An explanation of these coding schemes is given in Section 9.

The value of the attribute "type of coding" of a content portion description, that conforms to this Recommendation, is an ASN.1 object identifier or an integer.

REMARKS:

For bitmap encoding, the relationship between the order of the pels and the order of the bits within an octet is such that the first pel in the order of bits is allocated to the most significant bit of an octet.

7.2 *Coding attributes*

These attributes provide information required for encoding and decoding the content information, as well as other information that is intrinsic to the content portion and required by the content layout and imaging processes.

7.2.1 **compression**

CLASSIFICATION: Defaultable

APPLICABILITY: Formatted and formatted processable content architecture class

PERMISSIBLE VALUES: 'Compressed', 'uncompressed'

DEFAULT VALUE: 'Compressed'

DEFINITION:

This attribute indicates if the code extension technique for uncompressed mode is present in the content portion. This attribute can have one of two values:

-   'compressed' indicates that the code extension technique for uncompressed mode is not used.;

-   'uncompressed' indicates that the code extension technique for uncompressed mode may be used.

Note – Basic mode (compressed) encoding is used initially for coding all such content portions. The occurrence of the code extension technique for uncompressed mode coding results in subsequent content being coded in uncompressed mode.

REMARKS:

This attribute is only applicable if the value of the attribute "type of coding" is 'Rec. T.6 encoding' or 'Rec. T.4 two dimensional encoding'.

7.2.2 number of lines

CLASSIFICATION: Non-mandatory

APPLICABILITY: Formatted processable content architecture class

PERMISSIBLE VALUES: Positive integer

DEFINITION:
This attribute specifies the number of lines of pels within a content portion.

REMARKS:

This attribute takes effect during the content layout process.

7.2.3 number of pels per line

CLASSIFICATION: Defaultable for components of the formatted content architecture class;

Mandatory for components of the formatted processable content architecture class.

APPLICABILITY: Formatted and formatted processable content architecture classes

PERMISSIBLE VALUES: Non-negative integer

DEFAULT VALUE: The default value for components of the formatted content architecture class depends upon the "pel transmission density" as specified in Table 3/T.417.

No default value is specified for components of the formatted processable content architecture class.

TABLE 3/T.417

Default value of the presentation attribute
"number of pels per line"

Pel transmission density (BMU)	Default number of pels per line
1	10 368
2	5 184
3	3 456
4	2 592
5	2 074
6	1 728

DEFINITION:

This attribute specifies the number of pels in each line within a content portion.

7.2.4 number of discarded pels

CLASSIFICATION: Defaultable

APPLICABILITY: Formatted content architecture class

PERMISSIBLE VALUES: Non-negative integer

DEFAULT VALUE: If the number of pels per line exceeds the line length, then the default value is half the excess number of discarded pels, otherwise it is 0.

DEFINITION:

This attribute specifies the number of pels that are to be ignored at the beginning of each line within a content portion. The positioning of each line is started from the next pel in the line.

7.3 *Content information attributes*

7.3.1 **content information**

For raster graphics content architectures, the value of this attribute is an octet string representing a pel array encoded according to the value of the attribute "type of coding".

7.4 *Interactions with document architecture attributes*

The layout directives "indivisibility" and "concatenation" are not taken into account during the layout of raster graphics content associated with a basic logical component.

8 Formal definitions of raster graphics content architecture dependent data types

8.1 *Introduction*

This section contains formal definitions in ASN.1 notation (defined in Recommendation X.208) of the data types corresponding to presentation and coding attributes that are applicable to raster graphics content architectures.

These data types are:

- the data type to represent raster graphics content architecture specific presentation attributes in basic layout components, presentation styles and default value lists;

- the data type to represent raster graphics content architecture specific coding attributes in content portions;

- the data type to represent non-basic values of raster graphics content architecture presentation attributes in the document profile;

- the data type to represent non-basic values of raster graphics content architecture coding attributes in the document profile;

- the data type to represent non-standard default values of raster graphics content architecture presentation and coding attributes in the document profile.

8.2 *Representation of presentation attributes*

The data type "Raster-Graphics-Attributes" contains a set of subordinate data types that specify the raster graphics presentation attributes. Some of these subordinate data types are elementary but others are structured and are themselves made up of subordinate data types. The format of these data types is given below.

The subset of subordinate data types that may occur within a particular instance of the data type "Raster-Graphics-Attributes" depends upon the particular raster graphics content architecture level that is specified.

Raster-Gr-Presentation-Attributes { 2 8 1 7 2 }

DEFINITIONS ::= BEGIN

EXPORTS Raster-Graphics-Attributes,
     One-Of-Four-Angles,
     One-of-Two-Angles,
     Pel-Transmission-Density,
     Measure-Pair,
     Clipping,
     Pel-Spacing,
     Spacing-Ratio,
     Image-Dimensions;

Raster-Graphics-Attributes	::=	SET {	
pel-path	[0]	IMPLICIT One-Of-Four-Angles OPTIONAL,	
line-progression	[1]	IMPLICIT One-Of-Two-Angles OPTIONAL,	
pel-transmission-density	[2]	IMPLICIT Pel-Transmission-Density OPTIONAL,	
initial-offset	[3]	IMPLICIT Measure-Pair OPTIONAL,	
clipping	[4]	IMPLICIT Clipping OPTIONAL,	
pel-spacing	[5]	Pel-Spacing OPTIONAL,	
spacing-ratio	[6]	IMPLICIT Spacing-Ratio OPTIONAL,	
image-dimensions	[7]	Image-Dimensions OPTIONAL }	

One-Of-Four-Angles   ::= INTEGER { d0 (0),
                                  d90 (1),
                                  d180 (2),
                                  d270 (3) }

One-Of-Two-Angles   ::= INTEGER { d90 (1),
                                  d270 (3) }

Pel-Transmission-Density   ::= INTEGER {
                 p6 (1), -- 6 BMU
                 p5 (2), -- 5 BMU
                 p4 (3), -- 4 BMU
                 p3 (4), -- 3 BMU
                 p2 (5), -- 2 BMU
                 p1 (6), -- 1 BMU }

Measure-Pair	::=	SEQUENCE {	
horizontal	[0]	IMPLICIT INTEGER	
vertical	[0]	IMPLICIT INTEGER }	

Clipping	::=	SEQUENCE {	
first-coordinate-pair	[0]	IMPLICIT Coordinate-Pair OPTIONAL,	
second-coordinate-pair	[1]	IMPLICIT Coordinate-Pair OPTIONAL }	

Coordinate-Pair	::=	SEQUENCE {	
x coordinate		INTEGER	
y coordinate		INTEGER }	

Pel-Spacing	::=	CHOICE {	
spacing	[0]	IMPLICIT SEQUENCE {	
length		INTEGER,	
pel-spacing		INTEGER },	
null	[1]	IMPLICIT NULL }	

Spacing-Ratio	::=	SEQUENCE {	
line-spacing-value		INTEGER	
pel-spacing-value		INTEGER }	

Image-Dimension	::=	CHOICE {	
width-controlled	[0]	IMPLICIT SEQUENCE {	
minimum-width		INTEGER,	
preferred-width		INTEGER },	
height-controlled	[1]	IMPLICIT SEQUENCE {	
minimum-height		INTEGER,	

```
            preferred-height             INTEGER },
            area-controlled        [2]   IMPLICIT SEQUENCE {
            minimum-width                INTEGER,
            preferred-width              INTEGER,
            minimum-height               INTEGER,
            preferred-height             INTEGER,
            aspect-ratio-flag            INTEGER {
                                              fixed    (0),
                                              variable (1) }},
            automatic              [3]   IMPLICIT NULL }

        END
```

Note - The following types are also defined in orther Recommendations in the T.410 Series: 'One-Of-Four-Angles', 'One-Of-Two-Angles', 'MeasurePair'.

8.3   *Representation of coding attributes*

Raster-Gr-Coding-Attributes { 2 8 1 7 3 }

DEFINITIONS ::= BEGIN

EXPORTS Raster-Gr-Coding-Attributes,
        Compression;

```
Raster-Gr-Coding-Attributes      ::=  SET {
        number-of-pels-per-line        [0]  IMPLICIT INTEGER OPTIONAL,
        number-of-lines                [1]  IMPLICIT INTEGER OPTIONAL,
        compression                    [2]  IMPLICIT Compression OPTIONAL,
        number-of-discarded-pels       [3]  IMPLICIT INTEGER OPTIONAL }

Compression                      ::=  INTEGER {uncompressed (0),
                                              compressed (1)}
```

END

8.4   *Representation of non-basic features and non-standard defaults*

Raster-Gr-Profile-Attributes { 2 8 1 7 4 }

DEFINITIONS ::= BEGIN

EXPORTS Ra-Gr-Presentation-Feature,
        Ra-Gr-Coding-Attribute,
        Raster-Gr-Content-Defaults;

IMPORTS One-Of-Four-Angles,
        One-Of-Two-Angles,
        Pel-Transmission-Density,
        Measure-Pair,
        Clipping,
        Pel-Spacing,
        Spacing-Ratio,
        Image-Dimensions;

        FROM Raster-Gr-Presentation-Attributes,
             Compression,

        FROM Raster-Gr-Coding-Attributes;

```
Ra-Gr-Presentation-Feature       ::=  CHOICE {
        pel-path                       [9]   IMPLICIT One-Of-Four-Angles,
        line-progression               [10]  IMPLICIT One-Of-Two-Angles,
        pel-transmission-density       [11]  IMPLICIT Pel-Transmission-Density }
```

-- *The tag values used above preserve compatibility with Group 4 Class 1 facsimile data streams.*

```
Ra-Gr-Coding-Attribute           ::=  CHOICE {
        compression                    [0]   IMPLICIT Compression }
```

-- *The tag values used above preserve compatibility with Group 4 Class 1 facsimile data streams.*

```
Raster-Gr-Content-Defaults          ::=   SET {
    pel-path                              [0]   IMPLICIT One-Of-Four-Angles OPTIONAL,
    line-progression                      [1]   IMPLICIT One-Of-Two-Angles OPTIONAL,
    pel-transmission-density              [2]   IMPLICIT Pel-Transmission-Density OPTIONAL,
    pel-spacing                           [5]   Pel-Spacing OPTIONAL,
    spacing-ratio                         [6]   IMPLICIT Spacing-Ratio OPTIONAL,
    compression                           [8]   IMPLICIT Compression OPTIONAL }
    END
```

9   Coding schemes

A pel array may be represented within a text unit by means of one of the following encoding schemes:

- Group 4 facsimile encoding scheme;
- Group 3 facsimile encoding schemes;
- bitmap encoding scheme.

9.1   *Group 4 facsimile encoding scheme*

In this encoding scheme, a pel array is encoded according to Recommendation T.6. The colours "black" and "white" referred to in this Recommendation should be interpreted as "foreground" and "background", or "set" and "unset", respectively.

9.2   *Group 3 facsimile encoding scheme*

In these encoding schemes, a pel array is encoded according to one or two dimensional encoding schemes defined in Recommendation T.4. The colours "black" and "white" referred to in this Recommendation should be interpreted as "foreground" and "background", or "set" and "unset" respectively.

When using the Recommendation T.4 one- or two-dimensional encoding scheme, the encoded data belonging to each content portion must be terminated by a RTC (Return to control), the format of which is defined in Recommendation T.4. If the total number of bits belonging to a content portion is not a multiple of eight (i.e., an integral number of octets), then the RTC must be followed by the minimum number of '0' bits such that the last bit aligns on an octet boundary. In addition, the use of EOL is required to indicate the end of encoding of each line of pels and make up codewords can be used recursively to encode runs of pels longer than 2624.

When using the two-dimensional encoding scheme, any number of fill bits and any value of K-parameter may be used without declaration in coding attributes.

9.3   *Bitmap encoding scheme*

Each element in a pel array may have one of two distinct states. These are the set state, corresponding to foreground colour and the unset state corresponding to background colour. For the purpose of representing such an array within a content portion, each pel may be represented by a single bit which has the value '0' or '1' depending on the state of the pel. If the pel has the unset state, the value of the bit is '0'; otherwise the value of the bit is '1'.

In the bitmap encoding scheme, each row of the resulting array of bits is encoded, within a content portion, by a string of octets. If the number of bits in each row of the pel array is not a multiple of eight, then it is extended by the minimum number of '0' bits such that the last bit aligns on an octet boundary.

When the content portion is decoded, the coding attribute "number of pels per line" is used to determine the number of bits in each line that are significant, the remaining bits being ignored.

The relationship between the order of the pels and the order of the bits within an octet is such that the first pel in the order of bits is allocated to the most significant bit of an octet.

Note - This encoding scheme is distinct from the uncompressed mode of the Groups 3 and 4 facsimile encoding schemes.

10   Content layout process

This section describes a content layout process for basic logical objects associated with raster graphics content architectures.

Its purpose is to aid understanding of the semantics of the presentation attributes and coding attributes by describing the required results of such a process. However, it is not intended to specify any process that might be carried out in a particular implementation to achieve these results.

10.1   *Introduction*

10.1.1   *Purpose*

The content layout process describes the process of laying out raster graphic content into an allocated area. This area is referred to as the available area and is determined by the document layout process defined in Recommendation T.412.

The purpose of the content layout process is to convert content associated with basic logical components into content associated with basic layout objects.

The content layout process results in the creation of a basic layout object(s) into which the content is to be positioned. The dimensions of each basic layout object are returned to the document layout process, which determines the precise position of that basic layout object within the available area.

One of two methods can be followed for laying out the content of a basic logical object. These methods are:

- the fixed dimension content layout method;

- the scalable dimension content layout method.

The choice of method depends on the particular presentation attributes associated with the basic logical object.

10.1.2   *Available area*

The content layout process is constrained by the available area. The maximum dimensions that a basic layout object can take are constrained by the dimensions of the available area.

During the layout of the content associated with a basic logical object into a basic layout object, the following cases can occur:

- the formatted processable content fits into the available area;

- the formatted processable content does not fit into the dimensions of the available area. In this case, a new available area is required.

10.1.3   *Presentation attributes*

The content layout process takes into account the presentation attributes applying to the basic logical object with which the content is associated.

The presentation attributes applying to the content layout process can be specified in the generic layout structure and presentation styles. The values of these presentation attributes are determined according to the defaulting rules specified in Recommendation T.412.

10.1.4   *Coding attributes*

The content layout process takes into account the coding attributes applying to the content portion.

10.1.5   *Raster graphics content architecture classes*

The content layout process is only specified for basic logical objects associated with the formatted processable raster graphics content architecture class. The content layout process does not modify the form of the content.

10.1.6 *Layout of the content*

For the raster graphics formatted processable content architecture class, one case of laying out the content into basic objects is possible:

- single basic logical to single basic layout object: the content of a single basic logical object can be laid out into a single basic layout object and is the only content associated with this basic layout object.

10.2 *Notation*

The following notation is used in the description of the description of the determination of block dimensions:

BDH   Horizontal block dimension
BDV   Vertical block dimension.

NLC   Number of lines of the clipped array
NPC   Number of pels per line of the clipped array

AAH   Horizontal dimension of available area
AAV   Vertical dimension of available area

PS   Pel spacing
SR   Spacing ratio

10.3 *The fixed dimension content layout method*

If the value of the attribute "pel spacing" is specified as other than 'null', the fixed dimension content layout method is followed.

The fixed dimension content layout method creates a block with dimensions that satisfy the values of the following attributes:

- the presentation attributes (defined in Section 6):

  - "clipping";
  - "pel path";
  - "pel spacing";
  - "spacing ratio";

- the coding attributes (defined in § 7.2):

  - "number of lines";
  - "number of pels per line".

The fixed dimension content layout process creates a block of the minimum dimensions that are required to accommodate the clipped pel array in accordance with the pel spacing and line spacing. Note that the pel spacing is explicitly specified by the attribute "pel spacing", whereas the line spacing is determined from the pel spacing and the attribute "spacing ratio".

The horizontal and vertical block dimensions are determined, such that the reference areas of all the pels of the clipped pel array completely fill the basic layout object. The block dimensions depend on the pel path, pel spacing, spacing ratio, number of pels per line and number of lines as defined in Table 4/T.417.

TABLE 4/T.417

Dimensions of a basic layout object

Pel path	Horizontal block dimension (BDH)	Vertical block dimension (BDV)
0, 180 90, 270	NPC * PS NLC * PS * SR	NLC * SR * PS NPC * PS

Note – The notation used in this table is described in § 10.2.

If one of the following conditions occurs:

BDH > AAH or BDV > AAV

then the block will not fit into the available area. It is then the responsibility of the document layout process to determine whether or not the content layout process is to be repeated for an alternative available area.

10.4   *The scalable dimension content layout method*

If the value of the attribute "Pel spacing" is specified as 'null', the scalable content layout method is followed. In this case the pel spacing will depend upon the value of the attribute "image dimensions" and the available area provided by the document layout process.

The aim of the content layout process for scalable dimension content portions is to lay out the content, within the available area, in a basic layout object with the maximum dimensions possible, considering the image dimensions and the spacing ratio specified.

The block dimensions are determined by:

a)   the presentation attributes values (defined in Section 6):

- "clipping";
  "pel path";
- "image dimensions";
- "spacing ratio";

b)   the coding attributes (defined in § 7.2):

- "number of lines";
- "number of pels per line".

The scalable dimension content layout method first determines the aspect ratio of the clipped pel array, from the "number of pels per line" and "number of lines", taking into consideration the "spacing ratio":

$$\text{aspect ratio} = \frac{NPC}{NLC * SR}$$

Determination of the dimensions of the basic layout object depends on the value of the presentation attribute "image dimensions". The four possible cases are illustrated in Figures 4/T.417 to 7/T.417 and are described below:
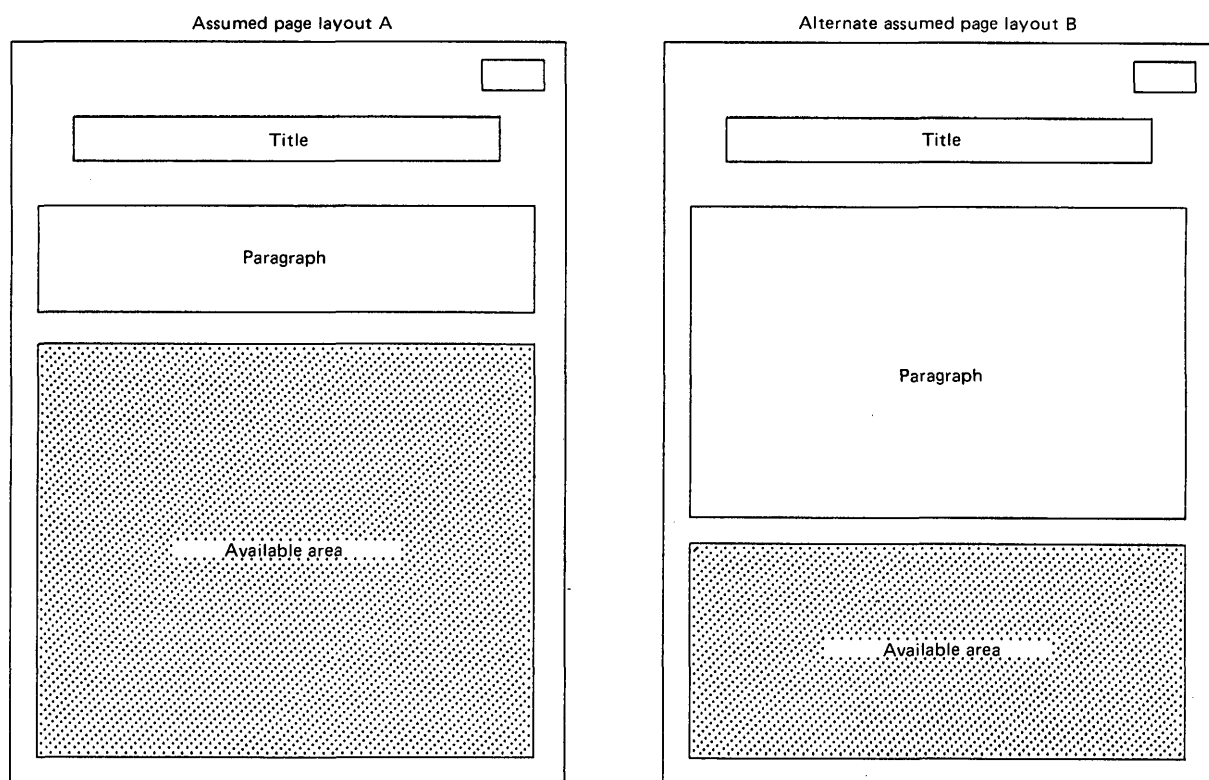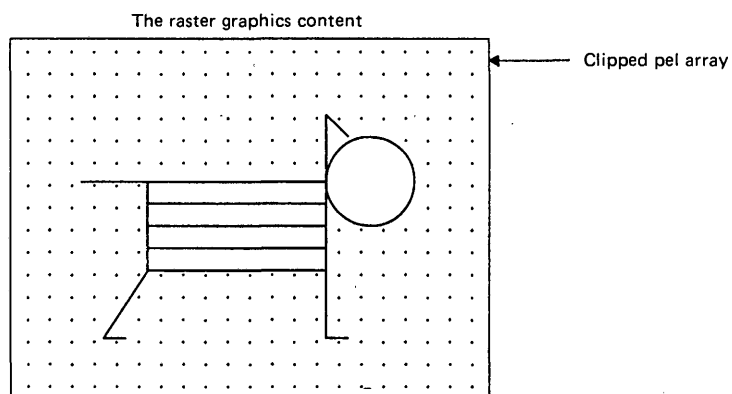
a)   The attribute "image dimensions" specifies a value for the parameter "width controlled". In this case the width of the basic layout object will be within the range specified by the originator.

The determination of basic layout object dimentions is constrained by the range of allowed widths given by the value of the parameter "width controlled", the dimensions of the available area and the aspect ratio of the clipped array.

The dimension of the basic layout object shall be determined such that: the basic layout object fits into the available area; the aspect ratio of the basic layout object is the same as that of the clipped pel array; and the width of the basic layout object has a value that is within the range of allowed widths. Also, the width of the basic layout object is determined such that the deviation from the value of "preferred width", specified by the parameter "width controlled", is as small as possible.

b)   The presentation attribute "image dimensions" specifies a value for the parameter "height controlled". In this case the height of the basic layout object will be within the range specified by the originator.

The determination of basic layout object dimensions is constrained by the range of allowed heights given by the value of the paramenter "height controlled", the dimensions of the available area and the aspect ratio of the clipped pel array.

The dimensions of the basic layout object shall be determined such that: the basic layout object fits into the available area; the aspect ratio of the basic layout object is the same as that of the clipped pel array; and the height of the basic layout object has a value that is within the range of allowed heights. Also, the height of the basic layout object is determined such that the deviation from the value of "preferred height", specified by the parameter "height controlled", is as small as possible.

c)   The attribute "image dimensions" specifies a value for the parameter "area controlled". In this case the dimensions of the basic layout object will be within the range specified by the originator. In particular, this can be used to ensure the basic layout object will have a fixed size.

The determination of basic layout object dimensions is constrained by the range of allowed heights and widths given by the value of the parameter "area controlled", the dimensions of the available area and, depending upon the value of the "aspect ratio flag" of the parameter "area controlled", by the aspect ratio of the clipped pel array.

The dimensions of the basic layout object shall be determined such that: the basic layout object fits into the available area; the width of the basic layout object has a value that is within the range of allowed widths; and the height of the basic layout object has a value that is within the range of allowed widths; and the height of the basic layout object has a value that is within the range of allowed heights. If the value of "aspect ratio flag" is 'fixed' there is the further constraint to the basic layout object dimensions, that the aspect ratio of the basic layout object must be the same as that of the clipped pel array. Also both the width and height of the basic layout object shall be chosen such, that their deviations from their preferred values, specified by the parameter "area controlled", are both as small as possible.

d)   The attribute "image dimensions" specifies a value for the parameter "automatic". In this case the dimensions of the basic layout object will be automatically adjusted to the page layout.

The determination of basic layout object dimensions is constrained by the dimensions of the available area and the aspect ratio of the clipped pel array.

The dimensions of the basic layout object have to be determined such that: the basic layout object fits into the available area; the width of the basic layout object is given the same value as the dimension of the available area in the same direction; and the height of the basic layout object is determined such, that the aspect ratio of the basic layout object is the same as that of the clipped pel array.

If the given constraints cannot be met, then the document layout process (defined in Recommendation T.412) is responsible for determining if the content layout method is to be repeated for an alternative available area.

The dimensions of a basic layout object are restricted to integral multiples of 1 SMU.
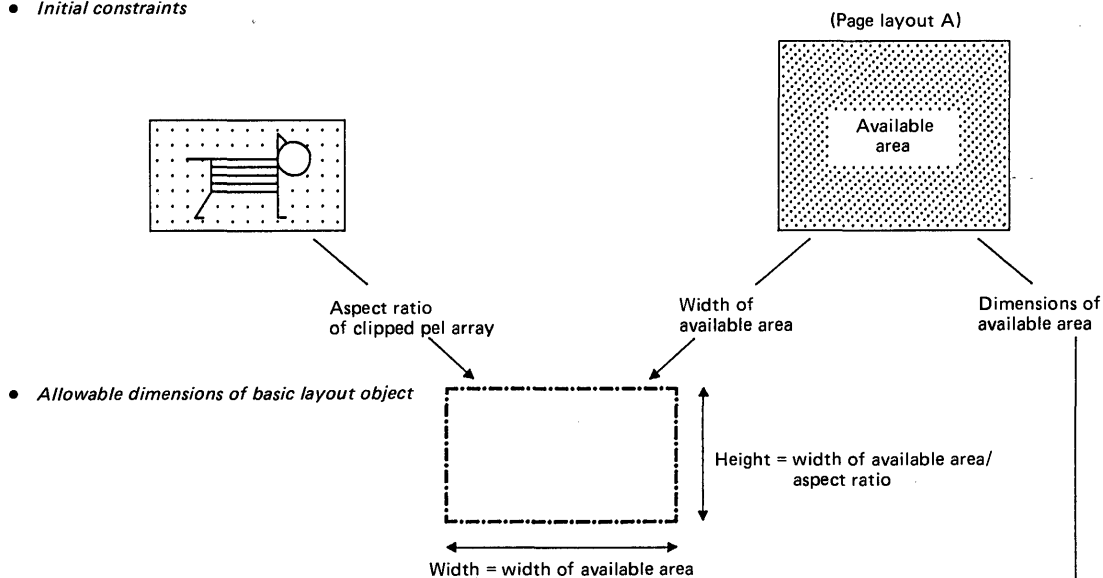
The raster graphics content

Clipped pel array

Assumed page layout A

Title

Paragraph

Available area

Alternate assumed page layout B

Title

Paragraph

Available area

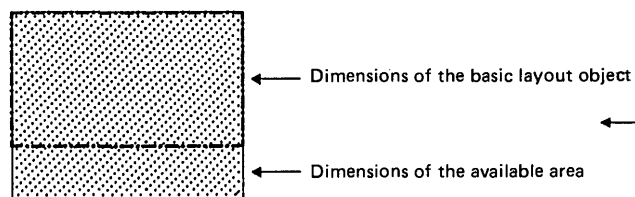T0801210-87

FIGURE 4/T.417

**Diagrams used to illustrate the process of determining
the basic layout object dimensions**

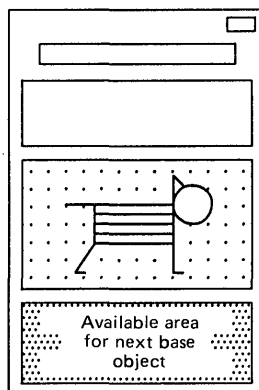Value of presentation attribute "image dimensions": automatic

● *Initial constraints*

(Page layout A)

Aspect ratio
of clipped pel array

Width of
available area

Dimensions of
available area

● *Allowable dimensions of basic layout object*

Height = width of available area/
aspect ratio

Width = width of available area

● *Basic layout object dimensions determined*

Dimensions of the basic layout object

Dimensions of the available area

● *Basic objects laid out, positioned and imaged*

Note – In this example, the positioning of these basic layout objects assumes normal fill order, the attribute "block alignment" has value "centered" and a certain separation between two consecutive blocks.

Available area
for next base
object

T0801220-87

FIGURE 5/T.417

Layout process for the presentation attribute "image dimensions"
when a value is specified for the parameter "automatic"

Value of "image dimensions": width controlled   Value of "image dimensions": height controlled

● *Initial constraints*

Preferred width

Minimum width

Aspect ratio
of clipped pel array

Minimum height

Preferred height

● *Allowable dimensions of basic layout object*

Minimum height =
Minimum width/aspect ratio

Preferred width

Minimum width

Minimum
height

Preferred
height

Minimum width = minimum height X aspect ratio
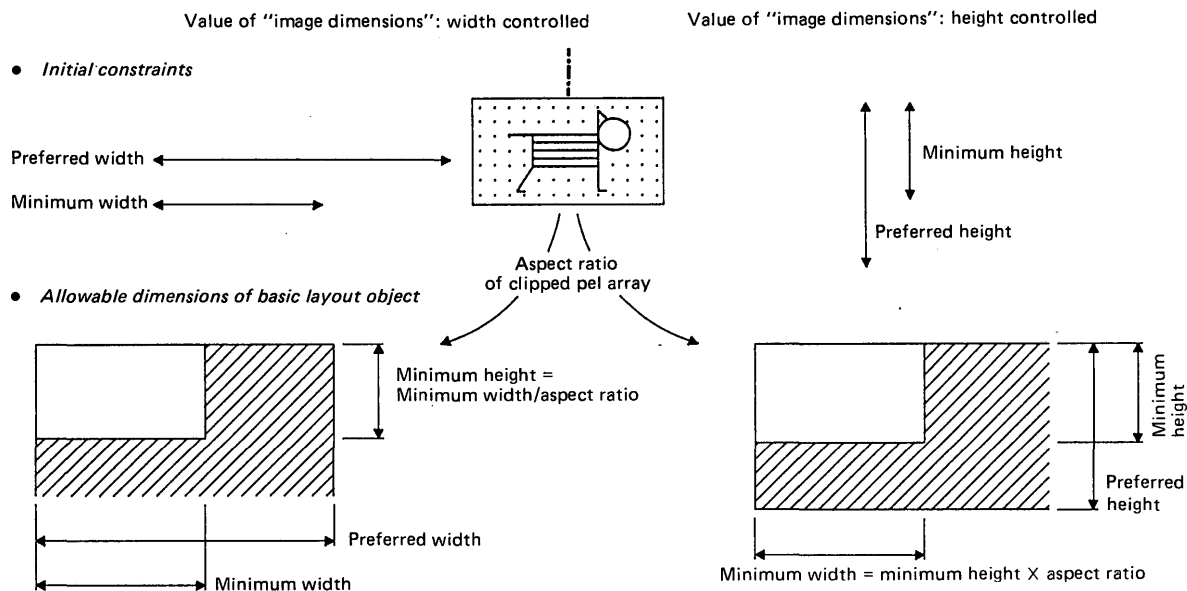
Note – The hatched areas show a range of allowable dimensions of basic layout object.

● *Basic layout object dimensions determined*

(Page layout A)

(Page layout B)

Dimensions of
available area

Deviation

Deviation
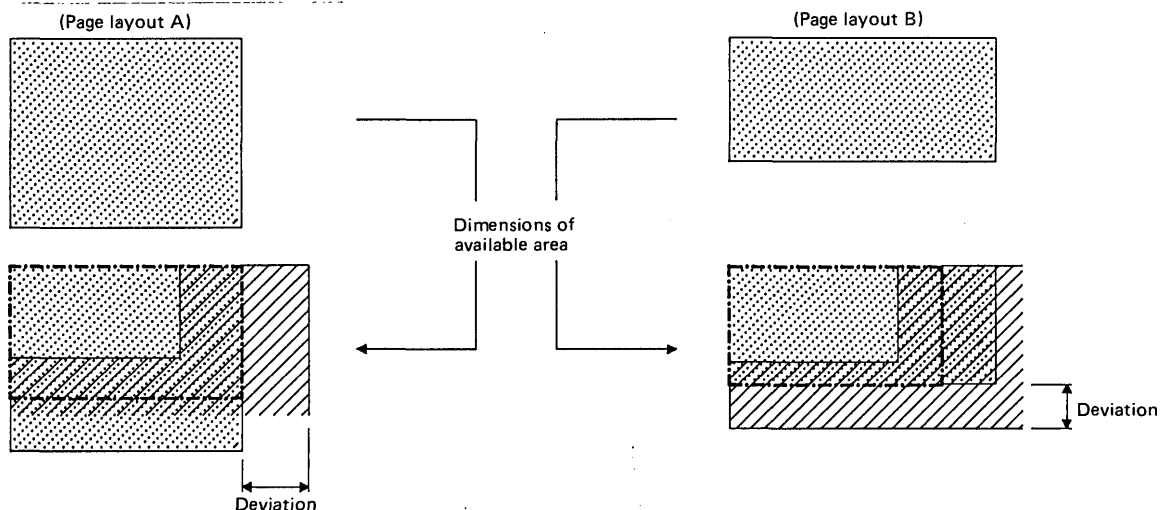
Note 1 – The basic layout object is indicated by the dashed-dotted boundary.

Note 2 – For specifying range of allowed image widths and layout A the preferred width cannot be satisfied due to the available width.

Note 3 – For specifying range of allowed image heights and layout B the major constraint is the height of the available area.

● *Basic objects laid out, positioned and imaged*

Note – In this example the positioning of these basic layout objects assumes normal fill order, the attribute "block alignment" has value "centered" and a certain separation between two consecutive blocks.

Available area
for next base
object
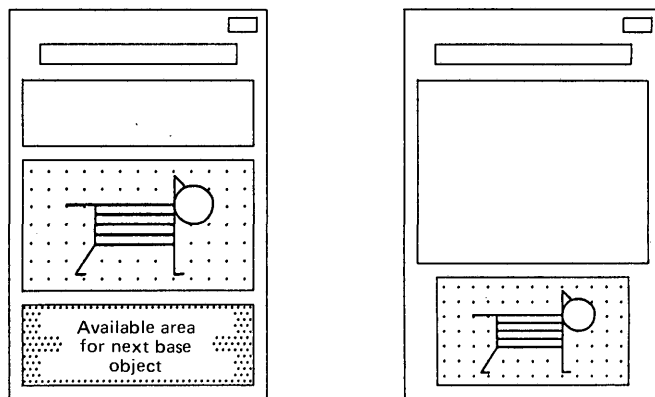
T0801230-87

FIGURE 6/T.417

**Layout process for the presentation attribute "image dimensions"
when a value is specified for the parameter "width controlled"
or "height controlled"**

Value of presentation attribute "image dimensions": area controlled

● *Initial constraints*
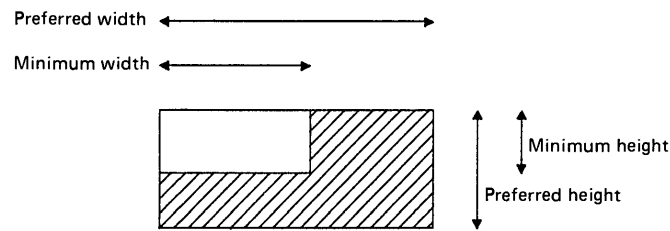
Note – The hatched area shows a range of allowable image dimensions.

● *Allowable dimensions of basic layout object*

The allowable dimensions of basic layout object are completely determined by the initial constraints.

● *Basic layout object dimensions determined (page layout A is used)*
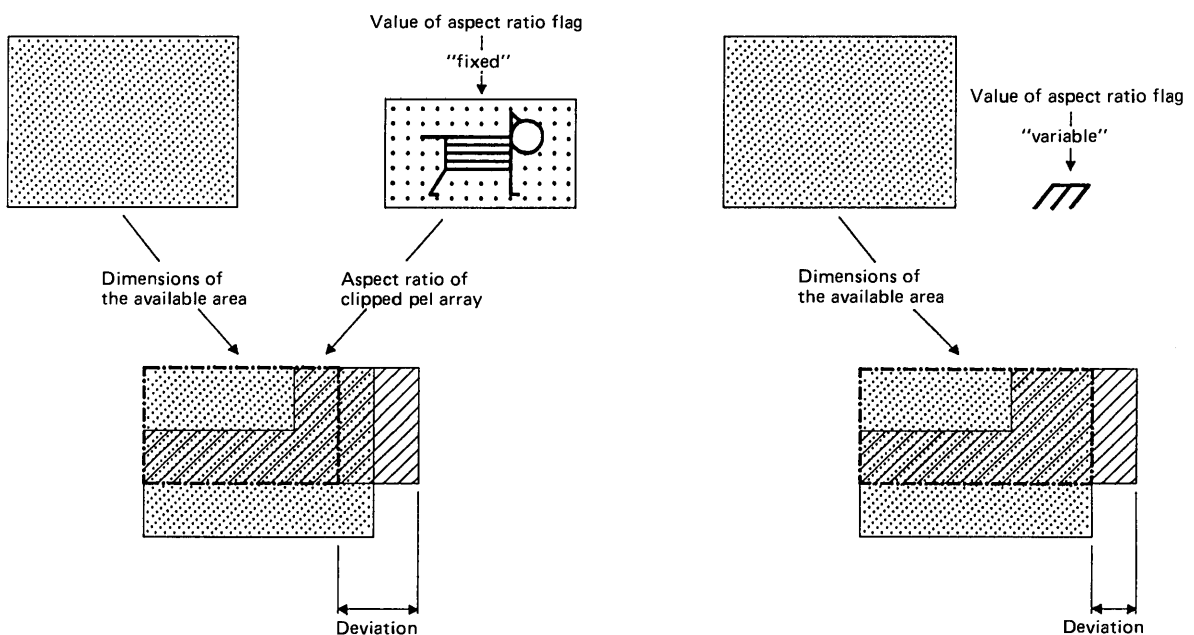
Note – The basic layout object is indicated by the dashed-dotted boundary.

● *Basic objects positioned, laid out and imaged*

Note – In this example the positioning of these basic layout objects assumes normal fill order, the attribute "block alignment" has value "centered" and a certain separation between two consecutive blocks.
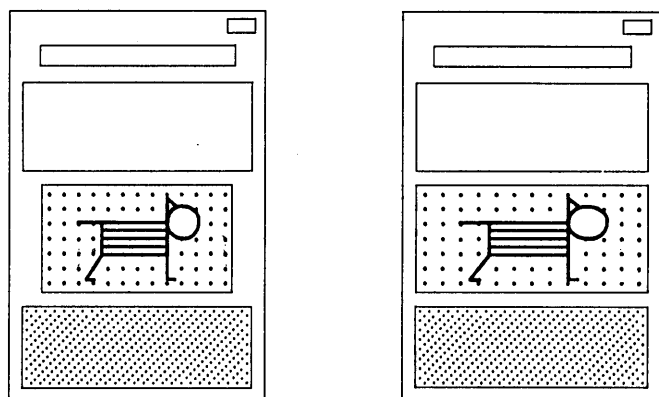
T0801240-87

FIGURE 7/T.417

**Layout process for the presentation attribute "image dimensions"
when a value is specified for the parameter "area controlled"**

11   Content imaging process

This section describes a content imaging process for basic layout objects associated with raster graphics content architectures.

Its purpose is to aid understanding of the semantics of the shared and layout presentation attributes and coding attributes by describing the required results of such a process. However, it is not intended to specify any process that might be carried out in a particular implementation to achieve these results.

11.1   *Introduction*

The content imaging process is only concerned with the layout structues, the presentation styles and the content of basic layout components conforming to this Recommendation.

The content imaging process is applicable to basic layout objects associated with the formatted and formatted processable raster graphics content architecture classes.

11.2   *Content imaging process for formatted form*

This section describes how the image of the content is influenced by the various presentation and coding attributes applying to the formatted raster graphics content architecture class.

The array of pels to be imaged consists only of those pels of the interchanged pel array which remain after the pels at the beginning of each line, specified by coding attribute "Number of discarded pels", have been subtracted. The first pel of this array is positioned at the initial point.

The initial point is determined by the attribute "initial offset".

Only the pels, which are positioned completely within the basic layout object are imaged.

11.3   *Content imaging process for formatted processable form*

This section describes how the image of the content is influenced by the various presentation and coding attributes applying to the formatted processable raster graphics content architecture class.

The clipped pel array is imaged in the basic layout object with the first pel at the initial point.

The initial point is determined by the pel path, line progression and by the dimensions of the basic layout object, as defined in Table 2/T.417.

The pel spacing is defined as the width of the basic layout object divided by the number of pels in a line of the clipped pel array.

The line spacing is defined as the height of the basic layout object divided by the number of lines in the clipped pel array.

12   Definition of raster graphics content architecture classes

This section defines the two classes of raster graphics content architectures described in § 4:

- formatted raster graphics content architecture class;

- formatted processable raster graphics content architecture class.

Tables 5/T.417 and 6/T.417 specify the categories of presentation and content portion attributes that pertain to these content architecture classes. Content architecture levels for use in application profiles can be defined from these classes using the rules specified in Recommendation T.411.

12.1   *Summary of raster graphic presentation attributes*

Table 5/T.417 contains a list of raster graphics presentation attributes, and identifies, for each content architecture class, those which are defaultable and those which are not applicable.

TABLE 5/T.417

Raster graphics presentation attributes

Presentation attribute	Content architecture class	
	Formatted form	Formatted processable form
Pel path	D	D
Line progression	D	D
Pel transmission density	D	-
Initial offset	D	-
Pel spacing	-	D
Spacing ratio	-	D
Clipping	-	D
Image dimensions	-	D

Note - The notation used in this table is: ·

- Non applicable
D Applicable and defaultable

12.2 *Summary of raster graphic content portion attributes*

Table 6/T.417 contains a list of raster graphics content portion attributes, and identifies, for each content architecture class, those which are mandatory, non-mandatory, defaultable and not applicable.

TABLE 6/T.417

Raster graphics content portion attributes

Content portion attribute	Content architecture class	
	Formatted form	Formatted processable form
Number of pels per line	D	M
Type of coding	D	D
Compression	D	D
Number of discarded pels	D	-
Number of lines	-	NM

Note 1 - The notation used in this table is:

- Not applicable;
D Applicable and defaultable;
NM Applicable and non-mandatory;
M Applicable and mandatory.

Note 2 - The attribute "compression" is only applicable if the value of the attribute "type of coding" is 'Rec. T.6 encoding' of 'Rec. T.4 two dimensional encoding'.

ANNEX A

(to Recommendation T.417)

Summary of raster graphics content architecture classes

(Informative)

This Annex summarizes the presentation attributes and content portion attributes that apply to each of the two content architecture classes (formatted and formatted processable) defined in Section 12, together with their permissible values and default values.

The purpose of this Annex is to facilitate the definition of raster graphisc content architecture levels for use in document application profiles (see Recommendation T.411).

A.1 Formatted raster graphics content architecture class

Content pertaining to the formatted raster graphics content architecture class may only be associated with basic layout components.

A.1.1 Presentation attributes

Attribute	Permissible values	Default value
Pel path	0, 90, 180, 270 degrees	0 degrees
Line progression	90, 270 degrees	270 degrees
Pel transmission density	6, 5, 4, 3, 2, 1 BMU	6 BMU
Initial offser	(Any integer, any integer)	See Note

Note – The default value of "initial offset" depends upon the pel path and line progression as defined in Table 2/T.417.

A.1.2 Content portion arttributes

Attribute	Permissible values	Default value
Number of pels per line	Any positive integer	See Note 1
Number of discarded pels	Any non-negative integer	See Note 2
Type de coding	Rec. T.6 encoding	Rec. T.6 encoding
Compression	Compressed, uncompressed as in Rec. T.6	Compressed as in Rec. T.6

Note 1 – The default number of pels per line depends upon the pel transmission density as defined in Table 3/T.417.

Note 2 – If the number of pels per line exceeds the image line length, the default number of discarded pels is half the excess number of pels, otherwise it is zero.

A.2 *Formatted processable raster graphics content architecture class*

Content pertaining to the formatted processable raster graphics form content architecture class may be associated with basic layout or logical objects.

A.2.1 *Presentation attributes*

Attribute	Permissible values	Default value
Pel path	0, 90, 180, 270 degrees	0 degrees
Line progression	90, 270 degrees	270 degrees
Pel spacing	(Any positive integer, any positive integer) SMU, 'null'	(4,1) SMU
Spacing ratio	(Any positive integer, any positive integer	(1,1)

Clipping
   First pair
 	
   Second pair
	
	(Any non-negative integer any non-negative integer)
	(Any non-negative integer any non-negative integer)
		See Note 1

Image dimensions
 	
Width controlled
   Minimum width
   Preferred width
 	
Height controlled
   Minimum height
   Preferred height
	See Note 2
	
	Any non-negative integer
	Any non-negative integer
	
	Any non-negative integer
	Any non-negative integer
		Automatic

Area controlled
   Minimum height
   Preferred height
   Minimum width
   Preferred width
   Aspect ratio flag
 	
Automatique
	Any non-negative integer
	Any non-negative integer
	Any non-negative integer
	Any non-negative integer
	Variable, fixed
	
	Null	

Note 1 - The default value of "clipping" is the first coordinate in the content portion (0,0) and the last coordinate (N-1, L-1), where N is number of pels per line and L is number of lines.

Note 2 - Minimum values must not be greater than preferred values.

Attribute	Permissible values	Default value
Number of pels per line	Any positive integer	None
Number of lines	Any positive integer	None
Type of coding	Bitman encoding, Rec. T.4 encoding (one-dim.), Rec. T.4 encoding (two dim.), Rec. T.6 encoding	Rec. T.6 encoding
Compression	Compressed, uncompressed as in Rec. T.6	Compressed as in Rec. T.6 See Note

Note - The attribute "compression" is only applicable if the value of the attribute "type of coding" is 'Rec. T.6. encoding' or 'Rec. T.4 two dimensional encoding'.

ANNEX B

(to Recommendation T.417)

**Recommendations for the development of raster graphics content
architecture levels in document application profiles**

(Informative)

This section provides examples of the definition of four raster graphics content architecture levels, in accordance with the rules specified in Recommendation T.411:

- RF-0 is an example of a content architecture level belonging to the formatted form content architecture class. RF-0 is identical to the content architecture used is the context of the document application profile defined in Recommendation T.503.

- RF-1 is an example of a content architecture level belonging to the formatted form content architecture class. RF-1 is identical to the content architecture used in the context of the document application profile defined in Recommendation T.501.

- RP-0 is an example of a content architecture level belonging to the formatted processable content architecture class. Content pertaining to this level may be laid out using the fixed dimension layout method.

- RP-1 is an example of a content architecture level belonging to the formatted processable content architecture class. Content pertaining to this level may be laid out using either the fixed dimension layout method (defined in § 10.3) or the scalable dimension layout method (defined in § 10.4).

Note - The application profile may have to specify additional rules for the use of these content architecture levels in particular applications.

B.1 *Raster graphics content architecture level RF-O*

RF-0 is a content architecture level derived from the formatted form raster graphics content architecture class.

B.1.1 *Presentation attributes*

Attribute	Basic values	Non-basic values	Default value
Pel path	0 degrees	None	Standard default value
Line progression	270 degrees	None	Standard default value
Pel transmission density	6 BMU	5, 4, 3, BMU	Standard default value

B.1.2 *Content portion attributes*

Attribute	Basic values	Non-basic values	Default value
Number of pels per line	Any positive integer	None	Standard default value
Number of discarded pels	Any positive integer	None	Standard default value
Type of coding	Rec. T.6 encoding	None	Standard default value
Compression	compressed as in Rec. T.6	Uncompressed as in Rec. T.6	Standard default value

B.2 *Raster graphics content architecture level RF-1*

RF-1 is a content architecture level derived from the formatted form raster graphics content architecture class.

B.2.1 *Presentation attributes*

Attribute	Basic values	Non-basic values	Default value
Pel path	0 degrees	None	Standard default value
Line progression	270 degrees	None	Standard default value
Pel transmission density	5, 4 BMUs	6, 3, 2, 1 BMUs	None See Note
Initial offset	Any integer Any integer	None	Standard default value

Note - "Pel transmission density" is specified as 'mandatory' for raster graphics content architecture level RF-1.

B.2.2 Content portion attribute

Attribute	Basic values	Non-basic values	Default value
Number of pels per line	Any positive integer	None	None See Note
Type of coding	Rec. T.6 encoding	None	None See Note
Compression	Compressed as in Rec. T.6	Uncompressed as in Rec. T.6	Standard default value

Note - "Number of lines per line" and "type of coding" are specified as 'mandatory' for raster graphics content architecture level RF-1.

B.3 Raster graphics content architecture level RP-0

RP-0 is a raster graphics content architecture level derived from the formatted processable content architecture class; it is laid out using the fixed dimension method of the processable content layout process.

B.3.1 Presentation attributes

Attribute	Basic values	Non-basic values	Default value
Pel path	0, 90, 180, 270 degrees	None	Standard default value
Line progression	90, 270 degrees	None	Standard default value
Pel spacing	(Any positive integer, any positive integer) SMU	None	Standard default value
Spacing ratio	(Any positive integer, any positive integer	None	Standard default value

Clipping
  First pair
 	
  Second pair
	(Any non-negative integer, any non-negative integer)
	(Any non-negative integer, any non-negative integer)
		None	Standard default value

B.3.2 *Content portion attributes*

Attribute	Basic values	Non-basic values	Default value
Number of pels per line	Any positive integer	None	None
Number of lines	Any positive integer	None	None
Type of coding	Rec. T.6 encoding	Bitman encoding Rec. T.4 encoding (one-dimensional) Rec. T.4 encoding (two-dimensional)	Standard default value
Compression See Note	Compressed as in Rec. T.6	Uncompressed as in Rec. T.6	Standard default value

Note - The attribute "compression" is only applicable if the value of the attribute "type of coding" is 'Rec. T.6 encoding' or 'Rec. T.4 two dimensional encoding'.

B.4 *Raster graphics content architecture level RP-1*

RP-1 is a raster graphics content architecture level derived from the formatted processable raster graphics content architecture class; it is laid out using either the fixed or scalable dimension methods of the processable content layout process (depending upon the value of "pel spacing").

B.4.1 *Presentation attributes*

Attribute	Basic values	Non-basic values	Default value
Pel path	0, 90, 180, 270 degrees	None	Standard default value
Line progression	90, 270 degrees	None	Standard default value
Pel spacing	(Any positive integer, any positive integer) or 'null'	None	Standard default value
Spacing ratio	(Any positive integer, any positive integer)	None	Standard default value

Attribute	Basic values	Non-basic values	Default value
Clipping		None	Standard default value
First pair	(Any non-negative integer, any non-negative integer)		
Second pair	(Any non-negative integer, any non-negative integer)		
Image dimensions	See Note	None	Standard default value
Width controlled			
Minimum width	Any non-negative integer	None	
Preferred width	Any non-negative integer	. None	
Height controlled			
Minimum height	Any non-negative integer	None	
Preferred height	Any non-negative integer	None	
Area controlled			
Minimum height	Any non-negative integer	None	
Preferred height	Any non-negative integer	None	
Minimum width	Any non-negative integer	None	
Preferred width	Any non-negative integer	None	
Aspect ratio flag	Variable, fixed	None	
Automatic	Null		

Note - Minimum values must not be greater than preferred values.

B.4.2 *Content portion attributes*

Attribute	Basic values	Non-basic values	Default value
Number of pels per line	Any positive integer	None	None
Number of lines	Any positive integer	None	None
Type of coding	Rec. T.6 encoding	Bitman encoding Rec. T.4 encoding (one-dimensional) Rec. T.4 encoding (two-dimensional)	Standard default value
Compression See Note	Compressed as in Rec. T.6	Uncompressed as in Rec. T.6	Standard default value

Note - The attribute "compression" is only applicable if the value of the attribute "type of coding" is 'Rec. T.6 encoding' or 'Rec. T.4 two dimensional encoding'.

OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT –
GEOMETRIC GRAPHICS CONTENT ARCHITECTURE [1]

CONTENT

---

[1]   This text is aligned with the final text of the corresponding International Standard ISO 8613-2.

Annex A   -   Summary of ASN.1 object identifiers

Annex B   -   Recommendation for the development of geometric graphics content architecture levels in document application profiles

Annex C   -   Basic differences between character primitives in the geometric graphics and the content of a basic component structured according to the character content architectures defined in Recommendation T.416

1   Scope

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.

1.2   The T.410-Series applies to the interchange of documents by means of data communication 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.

1.3   This Recommendation

    a)   defines a geometric graphics content architecture that can be used in conjunction with the document architecture defined in the Recommendation T.412;

    b)   defines an interface which allows the use of content structured according to ISO 8632 within documents structured according to the Recommendation T.412;

    c)   defines those aspects of positioning and imaging applicable to the presentation of this geometric graphics content architecture in a basic layout object;

    d)   defines the presentation attributes applicable to this geometric graphics content architecture;

    e)   describes a content layout process, which together with the document layout process described in the Recommendation T.412, describes the layout of geometric graphics content in basic layout and determines the dimensions of these basic layout objects.

2 Normative references

The following Recommendations and International Standards contain provision which, through reference in the 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.

- Rec. T.50 (1984), International Alphabet No. 5.

- Rec. X.208 (1988), Specification of abstract syntax notation one (ASN.1).

- ISO 8632-1: 1987, Information processing systems - Computer graphics - Metafile for the storage and transfer of picture description information - Part 1: Functional specification.

- ISO 8632-3: 1987, Information processing systems - Computer graphics - Metafile for the storage and transfer of picture description information - Part 3: Binary encoding.

3 Definitions, symbols, abbreviations and conventions

3.1 *Definitions*

For the purpose of this Recommendation, the definitions given, in Recommendation T.411 apply. In addition, the definitions given in ISO 8632 apply to this Recommendation.

3.2 *Symbols, abbreviations and conventions*

3.2.1 *CGM*

This term is used to reference the Computer Graphics Metafile defined in ISO 8632. It is used as a qualifier for terms defined in ISO 8632 (for example, CGM elements).

3.2.2 *Individual CGM elements*

Throughout this part whenever individual CGM elements are referred to they are written in uppercase; for example, SCALING MODE.

3.2.3 *CGM concepts*

Whenever the concepts defined in CGM are referred to they are written in mixed upper and lower case as appropriate; for example, Scaling Mode or Virtual Device Coordinates.

3.2.4 *Width and height*

Width is used throughout this Recommendation to express the extent of a 2-dimensional area in the direction given by the counter-clockwise rotation from the horizontal direction as specified by the geometric graphics presentation attribute "orientation".

Height is used throughout this Recommendation to express the extent of a 2-dimensional area orthogonal to its width.

Note - Width or height are mostly used in combination with a reference to an area; for example, width of the available area.

4 General principles

4.1 *Content architecture classes*

This Recommendation defines one class of geometric graphics content architectures:

- a formatted processable form, which allows for document content to be processed and also to be presented as intended by the originator. Formatted processable form content can be associated with any basic component.

4.2 Content

A content portion that is structured according to a geometric graphics content architecture represents a single pictorial image. The representation is based on the Computer Graphics Metafile (CGM) defined in ISO 8632 (see § 7.2).

The CGM provides a format suitable for the storage, retrieval and interchange of picture description information. The format consists of an ordered set of elements. These elements are split into groups that:

a) structure the information in the metafile;

b) specify the precision of the values used within the metafile;

c) control the display of the picture;

d) perform basic drawing actions;

e) control the attributes of the basic drawing actions;

f) provide access to non-standard device capabilities.

ISO 8632 defines the form (syntax) and the functional behaviour (semantics) of these elements.

4.3 Presentation attributes

The geometric graphics content architecture defines geometric graphics presentation attributes applicable to basic layout and basic logical components. The geometric graphics presentation attributes direct the content layout process and specify the initial conditions at the start of the presentation of the content associated with a basic object.

Only the geometric graphics presentation attributes specifying CGM defaults (see § 6.1.1) can be overwritten by CGM elements in the content of the basic component to which they apply.

4.4 Coding of content information

The ordered set of elements of the content portion is encoded according to the "binary" encoding defined in ISO 8632-3 and constitutes a complete CGM.

The functionality represented by the geometric graphics presentation attributes specifying CGM defaults (see § 6.1.1) and CGM element groups is that defined by ISO 8632-1 and ISO 8632-3, except that:

a) the defaulting rules are modified (see § 10.2.1);

b) the CGM shall contain only one picture.

4.5 Layout and imaging of the content

The geometric graphics content architecture describes a content layout process which creates a basic layout object and determines the dimensions of this object into which the content associated with a basic logical object is to be laid out.

It also describes a content imaging process which determines the image of the content.

5 Positioning

5.1 Introduction

This clause describes the general principles concerning the positioning of a part of the VDC Space within basic layout objects.

This part is known as the *region of interest*. It is a rectangular region within the VDC Space, and is defined by two Virtual Device Coordinate pairs termed "first corner" and "second corner".

Note – The VDC Space is used within ISO 8632 for positioning geometric graphics elements, specifying directions, specifying dimensions etc.

5.2 Measurement units and directions

The positioning of geometric graphics content within a basic layout object is specified with relation to an orthogonal coordinate system.

The definition of the region of interest specifies the origin and directions of the axes of the coordinate system, with respect to the basic layout object.

Figure 1/T.418 illustrates that depending on which coordinates of the VDC Space are referenced by "first corner" and "second corner" the region of interest can affect the orientation of the axes used when imaging the geometric graphics content.

The measurement units for the x- and y-axis of the coordinate system are determined by the relationship of the dimensions of the region of interest to the dimensions of the basic layout object.
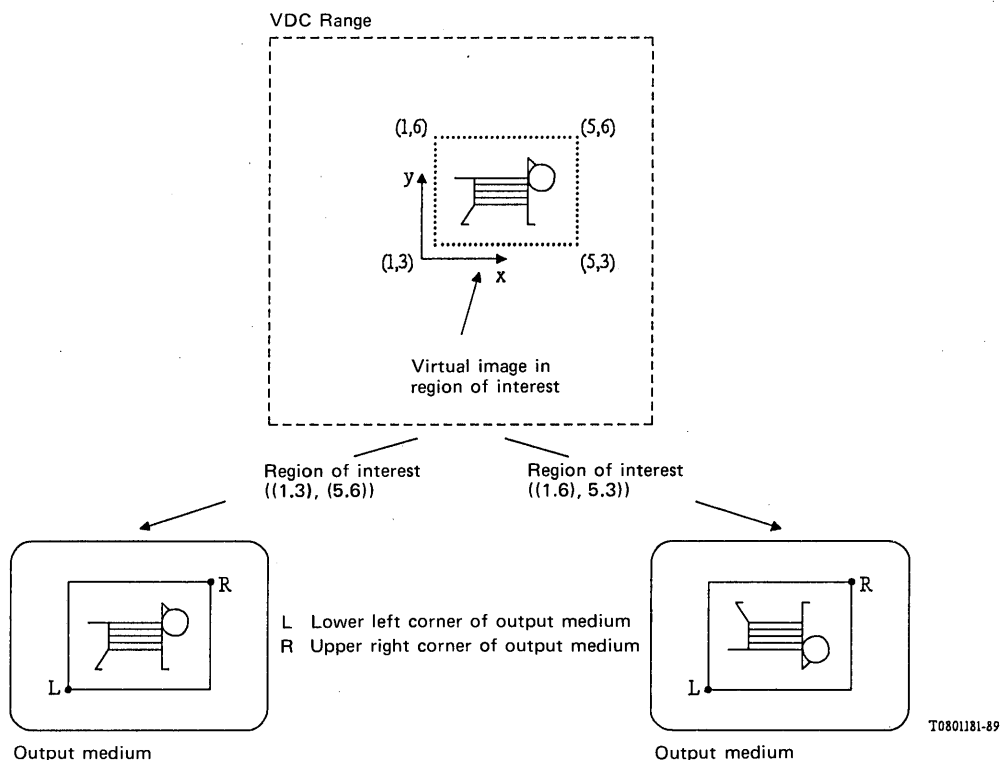

FIGURE 1/T.418

**Mapping of a virtual image defined in the VDC Space to an output
medium (e.g. a softcopy device) using different region
of interest specifications**

5.3   *The relationship between the region of interest and the basic layout object*

When imaging geometric graphics content, the geometric graphics presentation attribute "picture orientation" determines the relationship of the "first corner" of the region of interest to the corners of the basic layout object (see § 7.1.3).

The "first corner" of the region of interest is coincident with the corner of the basic layout object defined by the geometric graphics presentation attribute "picture orientation" (for example, the bottom left corner if the presentation attribute "picture orientation" has value '0°', see § 6.1.3). The "second corner" of the region of interest is coincident with the diagonally opposite corner of the basic layout object. It is implied that the x-axis of the VDC Space always maps to the direction parallel to the width of the basic layout object. Figures 2/T.418 and 3/T.418 illustrate this mapping.

CGM picture with region of interest ((1.3), (5.6)) as normally mapped to an output medium
the dotted line represents the region of interest, the arrows indicate the direction of increasing
coordinate values
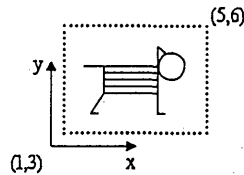
The picture above is imaged into the basic layout object as shown. The heavy border
represents the boundary of the basic layout object

T0801191-89

FIGURE 2/T.418

**Relationship of region of interest to the
basic layout object (right handed axes)**

CGM picture with region of interest ((1.6), (5.3)) as normally mapped to an output medium
the dotted line represents the region of interest, the arrows indicate the direction of increasing
coordinate values

The picture above is imaged into the layout object as shown. The heavy border
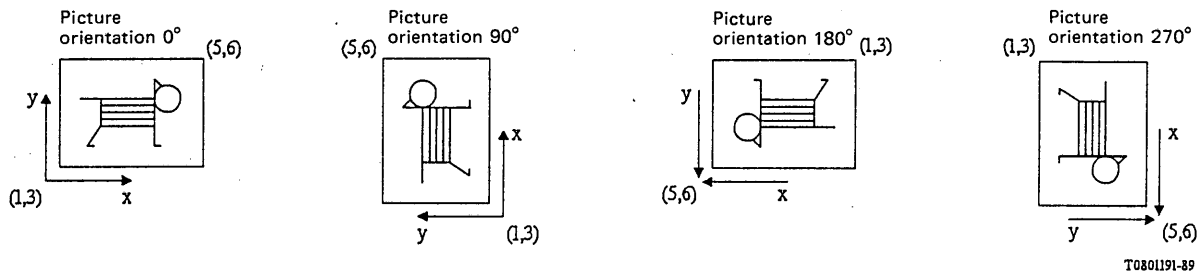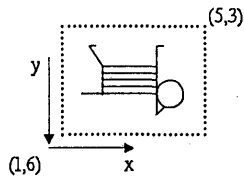represents the boundary of the basic layout object

T0801201-89
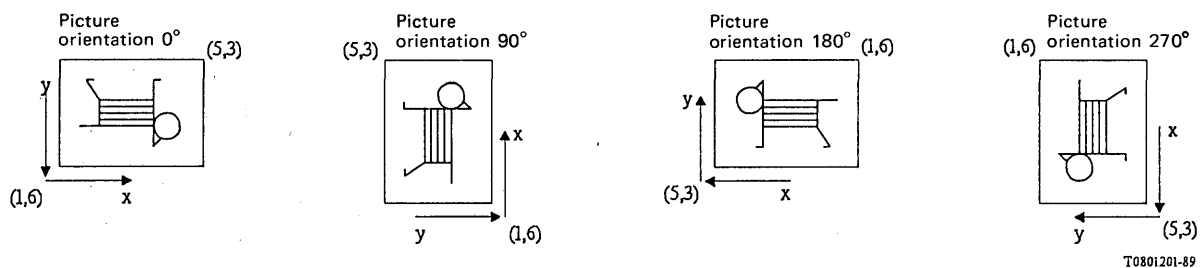
FIGURE 3/T.418

**Relationship of region of interest to the
basic layout object (left handed axes)**

6   Definition of geometric graphics presentation attributes

Presentation attributes specify the constraints and initial conditions relating to the layout
and imaging of a basic component. They may be specified for basic layout components, presentation
styles and default value lists.

The following categories of presentation attributes are defined:

a) logical presentation attributes which take effect during the content layout process but are ignored during the content imaging process;

b) layout presentation attributes which take effect during the content imaging process. Their values are either determined by a content layout process or specified by a process that creates or edits the content;

c) shared presentation attributes which take effect during both the content layout and imaging processes.

The geometric graphics presentation attributes are summarized in Table 1/T.418.

TABLE 1/T.418

Geometric graphics presentation attributes

Shared attributes	Layout attributes
Geometric graphics encoding announcer Line rendition Marker rendition Text rendition Filled area rendition	No layout presentation attributes are specified for this content architecture
Edge rendition Colour representation	Logical attributes
Transparency specification Transformation specification Region of interest specification Picture orientation	Picture dimensions

For each presentation attribute, a default value is defined. This value is used in the defaulting rules as defined in the Recommendation T.412.

This section also defines values specific to the geometric graphics content architecture for the content architecture class attributes. These attributes are defined in the Recommendation T.412.

6.1   *Shared presentation attributes*

6.1.1   *Attributes specifying CGM defaults*

The following presentation attributes provided information used for the construction and interpretation of the CGM defaults. They provide information used by the layout and imaging processes.

Note - This Recommendation uses the term "CGM defaults" whenever ISO 8632 uses the term "metafile defaults". This is intended to indicate the different semantics, in conformance with § 3.2.1, of "defaults" when used in the context of the T.410-Series of Recommendations (ODA) or ISO 8632 (CGM).

The default values given for the parameters of these presentation attributes have been derived from the defaults of the corresponding CGM elements as given in ISO 8632-1 and ISO 8632-3.

Note - Defaults for parameters specifying Direct Colour Values are given either as "foreground" representing the foreground colour, or "background" representing the background colour. The choice of foreground and background colour is implementation dependent. For reproduction on paper the background colour will normally be the colour of the paper, for instance white, and the foreground colour a contrasting colour, for instance black.

.The presentation attribute "geometric graphics encoding announcer" specifies the encoding of parameters of the remaining CGM defaults attributes and specifies default values for the corresponding CGM elements.

The values of the CGM defaults attributes applicable to a basic object are determined by the defaulting rules defined in the Recommendation T.412.

The values of each parameters of a CGM defaults attribute is:

a)   the value specified;

b)   if not specified, the value defined in the specification of the default values for the attribute applicable to this parameter.

This Recommendation does not contain definitions of parameters of presentation attributes specifying CGM defaults which have the same definitions and semantics as CGM elements or parameters of these CGM elements with corresponding names defined in ISO 8632-1. This clause and its sub-clauses contain definitions of parameters for which the definitions differ from the definitions given in ISO 8632-1 and those parameters which are not defined in ISO 8632-1.

For certain parameters the CGM defines value ranges as being reserved for registration. The meaning of these values will be defined using the established procedures of the ISO International Registration Authority for Graphical Items.

The specification of the parameters of the CGM defaults attributes, their permissible and default values is made in tabular form. Some of these parameters have values composed of several sub-parameters. They are shown intended beyond the parameters using a smaller sized font. The sub-parameters may be further substructured. This is shown by further indentation.

Tables 2/T.418, 3/T.418 and 4/T.418 define the default values for the bundle representations, the pattern representations and colour representations, respectively. These tables are used when determining the default state of the imaging process (see § 10.2.1).

TABLE 2/T.418

Default bundle representations

Representations	Bundle index				
	1	2	3	4	5
Line					
Line type	1 (solid)	2 (dash)	3 (dot)	4 (dash-dot)	5 (dash-dot-dot)
Line Width					
(if scaled)	1,0	1,0	1,0	1,0	1,0
(if absolute)	0,001 × length of longest side of default VDC Extent	0,001 × length of longest side of default VDC Extent	0,001 × length of longest side of default VDC Extent	0,001 × length of longest side of default VDC Extent	0,001 × length of longest side of default VDC Extent
Line Colour					
(if indexed)	1	1	1	1	1
(if direct)	foreground	foreground	foreground	foreground	foreground
Marker					
Marker type	1 (dot)	2 (plus)	3 (asterisk)	4 (circle)	5 (cross)
Marker size					
(if scaled)	1,0	1,0	1,0	1,0	1,0
(if absolute)	0,01 × length of longest side of default VDC Extent	0,01 × length of longest side of default VDC Extent	0,01 × length of longest side of default VDC Extent	0,01 × length of longest side of default VDC Extent	0,01 × length of longest side of default VDC Extent
Marker colour					
(if indexed)	1	1	1	1	1
(if direct)	foreground	foreground	foreground	foreground	foreground

Representations	Bundle index				
	1	2	3	4	5
Text					
Font index	1	1	none defined	none defined	none defined
Text precision	string	character	none defined	none defined	none defined
Character expansion factor	1,0	0,7	none defined	none defined	none defined
Character spacing	0,0	0,0	none defined	none defined	none defined
Text colour (if indexed) (if direct)	1 foreground	1 foreground	none defined none defined	none defined none defined	none defined none defined
Filled area					
Interior style	hollow	hatch	hatch	hatch	hatch
Fill colour (if indexed) (if direct)	1 foreground	1 foreground	1 foreground	1 foreground	1 foreground
Hatch index	1 (horizontal equally spaced parallel lines)	1 (horizontal equally spaced parallel lines)	2 (vertical equally spaced parallel lines)	3 (positive slope equally spaced parallel lines)	4 (negative slope equally spaced parallel lines)
Pattern index	1	1	1	1	1
Edge					
Edge type	1 (solid)	2 (dash)	3 (dot)	4 (dash-dot)	5 (dash-dot-dot)
Edge width (if scaled) (if absolute)	1,0 $0,001 \times$ length of longest side of default VDC Extent	1,0 $0,001 \times$ length of longest side of default VDC Extent	1,0 $0,001 \times$ length of longest side of default VDC Extent	1,0 $0,001 \times$ length of longest side of default VDC Extent	1,0 $0,001 \times$ length of longest side of default VDC Extent
Edge colour (if indexed) (if direct)	1 foreground	1 foreground	1 foreground	1 foreground	1 foreground

TABLE 3/T.418

Default pattern representations

Pattern table entry	Pattern table index
	1
NX (number of columns in pattern array)	1
NY (number of rows in pattern array)	1
Local colour precision	0
Colour Index array (if indexed)	{1}
Value array (if direct)	{foreground}

TABLE 4/T.418

Default colour representations

Colour table entry	Colour table index	
	0	1
Direct colour value	background	foreground

6.1.1.1   *Geometric graphics encoding announcer*

Parameter	Permissible values	Default
VDC type	integer, real	integer
Integer precision	8, 16, 24, 32	16
Real precision	(floating point format, 9, 23) (floating point format, 12, 52) (fixed point format, 16, 16) (fixed point format, 32, 32)	(fixed point format, 16,16)
Index precision	8, 16, 24, 32	16
Colour precision	8, 16, 24, 32	8
Colour index precision	8, 16, 24, 32	8
Maximum Colour index	any integer $\geq$ 0	63
Colour value extent	any pair of direct colour values	((0,0,0), (255,255,255))
Colour selection mode	indexed - direct	indexed
VDC integer precision	16, 24, 32	16
VDC real precision	(floating point format, 9, 23) (floating point format, 12, 52) (fixed point format, 16, 16) (fixed point format, 16, 16)	(fixed point format 16,16)

This presentation attribute specifies default values for VDC Type, Integer Precision, Real Precision, Index Precision, Colour Precision, Colour Index Precision, Maximum Colour Index, Colour Value Extent, Colour Selection Mode, VDC Integer Precision and VDC Real Precision.

This presentation attribute also determines the encoding of parameters of the remainder of the CGM default attributes.

6.1.1.2   *Line rendition*

Parameter	Permissible values	Default
Line width specification mode	absolute, scaled	scaled
Line bundle index	any integer > 0	1
Line type	1 through 5 plus any registered line type > 5 (see Note)	1 (solid)
Line width (if scaled) (if absolute)	any real $\geq$ 0,0 any non-negative VDC Value	1,0 0,001 x length of longest side of default VDC extent
Line colour (if indexed) (if direct)	any integer $\geq$ 0 any Direct Colour Value	1 foreground
Line aspect source flags line type asf line width asf line colour asf	any three-tuple of (line type asf, line width asf, line colour asf) bundled, individual bundled, individual bundled, individual	(individual, individual, individual)
Line bundle specifications line bundle index line bundle representation line type line width (if scaled) (if absolute) line colour (if indexed) (if direct)	any list containing zero, one or more elements any integer > 0 any three-tuple of (line type, line width, line colour) as for individual as for individual as for individual as for individual as for individual	empty list

Note - The permissible values of the parameter are restricted to values which are standardized and registered. Private values are not permitted.

This presentation attribute sets default values used for the presentation of the line primitives in the geometric graphics content portion. It specifies the default values for the Line Width Specification Mode, the Line Bundle Index, the individual CGM line attributes, the line aspect source flags and specifies the default line bundle representations.

The line bundle specifications parameter defines the initial line representations to be used for imaging a basic object. For each unspecified representation the values in Table 2/T.418 apply.

This parameter consists of a list of zero, one or more pairs. Each pair consists of:

a) line bundle index;

b) line bundle representation, which supplies values for the bundled CGM line attributes.

6.1.1.3 *Marker rendition*

Parameter	Permissible values	Default
Marker size specification mode	absolute, scaled	scaled
Marker bundle index	any integer > 0	1
Marker type	1 to 5 plus any registered marker type > 5 (see Note)	3 (asterisk)
Marker size (if scaled) (if absolute)	any real $\geq$ 0,0 any non-negative VDC value	1,0 0,01 $\times$ length of longest side of default VDC extent
Marker colour (if indexed) (if direct)	any integer $\geq$ 0 any Direct Colour Value	1 foreground
Marker aspect source flags marker type asf marker size asf marker colour asf	any three-tuple of (marker-type asf, marker size asf, marker colour asf) bundled, individual bundled, individual bundled, individual	(individual, individual, individual)
Marker bundle specifications marker bundle index marker bundle representation marker type marker size (if scaled) (if absolute) marker colour (if indexed) (if direct)	any list containing zero, one or more elements any integer > 0 any three-tuple of (marker type, marker size, marker colour) as for individual as for individual as for individual as for individual as for individual	empty list

Note - The permissible values of the parameter are restricted to values which are standardized and registered. Private values are not permitted.

This presentation attribute sets default values used for the rendition of the marker primitives in the geometric graphics content portion. It specifies the default values for the Marker Size Specification Mode, the Marker Bundle Index, the individual CGM marker attributes, the marker aspect source flags and specifies the default marker bundle representations.

The marker bundle specifications parameter defines the initial marker representations to be used for imaging a basic object. For each unspecified representation the values in Table 2/T.418 apply.

This parameter consists of a list of zero, one or more pairs. Each pair consists of:

a)   marker bundle index;

b)   marker bundle representation, which supplies values for the bundled CGM marker attributes.

6.1.1.4   *Text rendition*

Parameter	Permissible values	Default
Font list	any list of registered font names (see Note)	List containing one element: the registered name of any font that can represent the nationality-independent character subset of Recommendation T.50
Character set list	any list of (character set type, designation sequence tail)	(94-character sets, designation sequence tail that is registered for a character set which includes the nationality-independent subset of Recommendation T.50 in the positions specified in Recommendation T.50)
Character set type	94-character sets, 96-character sets, 94-character multibyte sets, 96-character multibyte sets, complete code	
Designation sequence tail	any registered designation sequence tail (see Note)	
Character coding announcer	basic 7-bit, basic 8-bit, extended 7-bit, extended 8-bit (see Note)	basic 7-bit
Text bundle index	any integer > 0	1
Text font index	any integer > 0	1
Text precision	string, character, stroke	string
Character expansion factor	any real > 0,0	1.0
Character spacing	any real	0.0
Text colour		
(if indexed)	any integer ≥ 0	1
(if direct)	any direct colour value	foreground
Character height	any non-negative VDC value	0.1 × length of the longest side of the default VDC extent
Character orientation	any pair of VDC Vectors which have non-zero length and are not collinear	((0;1), (1;0))
Text path	right, left, up, down	right

Parameter	Permissible values	Default
Text alignment	any four-tuple of (horizontal alignment, vertical alignment, continuous horizontal alignment, continuous vertical alignment)	(normal horizontal, normal vertical n/a, n/a)
horizontal alignment	normal horizontal, left, centre, right, continuous horizontal	
vertical alignment	normal vertical, top, cap, half, base, bottom, continuous vertical	
continuous horizontal alignment	any real	
continuous vertical alignment	any real	
Character set index	any integer > 0	1
Alternate character set index	any integer > 0	1
Text aspect source flags	any five-tuple of (text font index asf, text precision asf, character expansion factor asf, character spacing asf, text colour asf)	(individual, individual, individual, individual, individual)
text font asf	bundled, individual	
text precision asf	bundled, individual	
character expansion factor asf	bundled, individual	
character spacing asf	bundled, individual	
text colour asf	bundled, individual	
Text bundle specifications	any list containing zero, one or more elements	empty list
Text bundle index text bundle representation	any integer > 0 any five-tuple of (text font index, text precision, character expansion factor, character spacing, text colour)	

Parameter	Permissible values	Default
Text font index	as for individual	
Text precision	as for individual	
Character expansion factor	as for individual	
Character spacing	as for individual	
Text colour (if indexed) (if direct)	as for individual as for individual	

Note – The permissible values of the parameter are restricted to values which are standardized and registered. Private values are not permitted.

This presentation attribute sets default values used for the rendition of the text primitives in the geometric graphics content portion. It specifies the default values for the Font List, Character Set List, Character Coding Announcer, the Text Bundle Index, the individual CGM text attributes, the text aspect source flags and specifies the default text bundle representations.

The text bundle specifications parameter defines the initial text representations to be used for imaging a basic object. For each unspecified representation the values in Table 2/T.418 apply.

This parameter consists of a list of zero, one or more pairs. Each pair consists of:

a)   text bundle index;

b)   text bundle representation, which supplies values for the bundled CGM text attributes.

6.1.1.5   *Filled area rendition*

Parameter	Permissible values	Default
Fill bundle index	any integer > 0	1
Interior style	hollow, solid, pattern, hatch, empty	hollow
Fill colour (if indexed) (if direct)	any integer ≥ 0 any direct colour value	1 foreground
Hatch index	1 through 6 plus any registered hatch index > 6 (see Note)	1 (horizontal equally spaced parallel lines)
Pattern index	any integer > 0	1
Fill reference point	any virtual device coordinate	First corner of default VDC Extent

Parameter	Permissible values	Default
Pattern size	any four-tuple of (height vector x component, height vector y component, width vector x component, width vector y component)	(0, height of default VDC Extent, width of default VDC Extent, 0)
height vector x component	any VDC value	
height vector y component	any VDC value	
width vector x component	any VDC value	
width vector y component	any VDC value	
Pattern table specifications	any list containing zero, one or more pattern table elements	empty list
pattern table index	any integer > 0	
nx (number of columns in pattern)	any integer > 0	
ny (number of rows in pattern)	any integer > 0	
local colour recision	0, 1, 2, 4, 8, 16, 24, 32	
Colour index array (if indexed)	nx * ny cells of any integer $\geq$ 0	
value array (if direct)	nx * ny cells of any direct colour Value	
Fill aspect source flags	any four-tuple of (interior style asf, fill colour asf, hatch index asf, patter index asf)	(individual, individual individual, individual)
interior style asf	bundled, individual	
fill colour asf	bundled, individual	
hatch index asf	bundled, individual	
pattern index asf	bundled, individual	
Fill bundle specifications	any list containing zero, one or more elements	empty list

Parameter	Permissible values	Default
Fill bundle index Fill bundle representation	any integer > 0 any four-tuple of (interior style, fill colour, hatch index, pattern index)	
Interior style	as for individual	
Fill colour (if indexed) (if direct)	as for individual as for individual	
Hatch index	as for individual	
Pattern index	as for individual	

Note - The permissible values for the parameter are restricted to values which are standardized and registered. Private values are not permitted.

This presentation attribute sets default values used for the presentation of the interior of filled area primitives of a geometric graphics content portion. It specifies the default values for the Fill Bundle Index, the individual CGM filled area attributes, the pattern representations, the filled area aspect source flags and default fill bundle representations, applicable to the interior region of the filled area.

The pattern table specifications parameter is a list which supplies a complete set of values for zero, one or more pattern table entries. For each unspecified pattern table entry the values in Table 3/T.418 apply.

The fill bundle specifications parameter defines the initial bundle representations to be used for imaging a basic objet. For each unspecified representation the values in Table 2/T.418 apply.

This parameter consists of a list of zero, one or more pairs. Each pair consists of:

a) fill bundle index;

b) fill bundle representation, which supplies values for the bundled CGM filled area attributes.

6.1.1.6 *Edge rendition*

Parameter	Permissible values	Default
Edge width specification mode	absolute, scaled	scaled
Edge visibility	off, on	off
Edge bundle index	any integer > 0	1
Edge type	1 through 5 plus any registered line type (see Note)	1 (solid)
Edge width (if scaled) (if absolute)	any real $\geq 0,0$ any non-negative VDC value	1.0 0.001 × length of longest side of default VDC extent

Parameter	Permissible values	Default
Edge colour (if indexed) (if direct)	any integer > 0 any direct colour value	1 foreground
Edge aspect source flag edge type asf edge width asf edge colour asf	any three-tuple of (edge type asf, edge width asf, edge colour asf) bundled, individual bundled, individual bundled, individual	(individual, individual, individual)
Edge bundle specifications Edge bundle index Edge bundle representation Edge type Edge width (if scaled) (if absolute) Edge colour (if indexed) (if direct)	any list containing zero, one or more elements any integer > 0 any three-tuple of (edge type, edge width, edge colour) as for individual as for individual as for individual as for individual as for individual	empty list

Note - The permissible values of the parameter are restricted to values which are standardized and registered. Private values are not permitted.

This presentation attribute sets default values used for the presentation of the edges of the filled area primitives in the geometric graphics content portion. It specifies the default values for the Edge Width Specification Mode, the Edge Visibility, the Edge Bundle Index, the individual CGM edge attributes, the edge aspect source flags and specifies the default edge bundle representations, applicable to the boundary of the filled area.

The edge bundle specifications parameter defines the initial edge representations to be used for imaging a basic object. For each unspecified representation the values in Table 2/T.418 apply.

This parameter consists of a list of zero, one or more pairs. Each pair consists of:

a)   edge bundle index;

b)   edge bundle representation, which supplies values for the bundled CGM edge attributes.

6.1.1.7   *Colour representations*

Parameter	Permissible values	Default
Background colour	any direct colour value	background
Colour table specifications	any list containing zero, one or more colour table elements	empty list
Starting index	any integer ≥ 0	
Colour list	any list containing one or more direct colour values	

This presentation attribute sets the default value for Background Colour and defines the initial colour representations to be used for imaging a basic object.

The parameter colour table specifications is a list which supplies a complete set of values for zero, one or more elements each of which supplies a list of Direct Colour Value specifications together with the index of the starting colour table entry for a continuous interval of one or more colour table entries. For each unspecified colour representations the values of Table 4/T.418 apply.

6.1.1.8   *Transparency specification*

Parameter	Permissible values	Default
Transparency	off, on	on
Auxiliary colour (if indexed) (if direct)	any integer $\geq$ 0 any direct colour value	0 background

This presentation attribute sets the default values for Transparency and Auxiliary Colour.

Note 1 - The default values of AUXILIARY COLOUR given in ISO 8632-1:1987(E) and ISO 8632-3:1987(E) are inconsistent. Due to these inconsistencies an arbitrary choice of 'background' was made.

Note 2 - Auxiliary Colour as defined in ISO 8632 is intended to address hardware features commonly available in raster devices. Some devices may have no such capabilities, or may have a subset of these capabilities to which this parameter pertains. Simulations of such a feature may be very complex. ISO 8632 does not require that a CGM interpreter need simulate the feature when it is not available in the hardware or firmware.

6.1.1.9   *Transformation specification*

Parameter	Permissible values	Default
VDC extent	any pair of Virtual Device Coordinates defining a rectangle	((0;0), (1;1))
Clip rectangle	any pair of Virtual Device Coordinates defining a rectangle	same as VDC extent
Clip indicator	off, on	on

This presentation attribute sets the default values for VDC extent, Clip rectangle and Clip indicator.

The Virtual Device Coordinate (0; 0) is the symbolic value of the origin of the coordinate system of the VDC Space. The Virtual Device Coordinate (1; 1) is the symbolic value of:

a)   (1,0; 1,0) for VDC Type 'real'

b)   (32767, 32767) if VDC Type is 'integer'.

Note - The default values for VDC EXTENT for VDC Type 'real' given in ISO 8632-1:1987(E) and ISO 8632-3:1987(E) are inconsistent. Due to these inconsistencies an arbitrary choice of '(1,0; 1,0)' was made for the Second Corner.

6.1.2 *Region of interest specification*

This presentation attribute specifies the region of interest used when laying out and imaging the content of the basic object.

This presentation attribute consists of one of the following parameters:

a) "rectangle", consisting of two sub-parameters specifying the Virtual Device Coordinates for the first corner and the Virtual Device Coordinates for the second corner of the region of interest;

b) "automatic", not comprising any sub-parameters and its value is 'null'.

The default value is the parameter "automatic".

If the parameter "automatic" is specified the region of interest is the same as the VDC Extent.

If coordinate pairs for the first corner and second corner are specified using the parameter "rectangle", then the region of interest is specified by these values.

6.1.3 *Picture orientation*

This presentation attribute specifies with which corner of the basic layout object the first corner of the region of interest is to be made coincident.

This presentation attribute specifies one of the four corners of a basic layout object:

0° bottom left corner;
90° bottom right corner;
180° top right corner;
270° top left corner.

The default value is 0° (bottom left corner).

6.2 *Layout presentation attributes*

No layout presentation attributes are specified for this content architecture.

6.3 *Logical presentation attributes*

6.3.1 *Picture dimensions*

This presentation attribute specifies the intended dimensions of the basic layout object that is to contain the image defined by the geometric graphics content portion.

The value of this attribute consists of one of four parameters:

a) "width controlled";

b) "height controlled";

c) "area controlled";

d) "automatic".

The parameter "width controlled" has two sub-parameters "minimum width" and "preferred width".

The parameter "height controlled" has two sub-parameters "minimum height" and "preferred height".

The parameter "area controlled" has five sub-parameters:

a) "minimum width";

b) "preferred width";

c) "minimum height";

d) "preferred height";

e) "aspect ratio flag".

The parameter "automatic" has no sub-parameter and its value is 'null'.

The sub-parameter "aspect ratio flag" has the value 'fixed' or 'variable'. All other sub-parameters have non-negative integer values in SMUs.

The default value of this attribute is the parameter "automatic".

The values of "minimum width" and "preferred width" specify the lower limit and the upper limit of the allowed widths of the basic layout object. The value of the "minimum width" shall not be greater than the value of the "preferred value".

The values of "minimum height" and "preferred height" specify the lower limit and the upper limit of the allowed heights of the basic layout object. The value of the "minimum height" shall not be greater that the value of the "preferred height".

If either or both of the values for "preferred width" and "preferred height" are specified the dimensions of the basic layout object shall be as close to the corresponding specified values as possible.

"Aspect ratio flag" has one of the values 'fixed' or 'variable', specifying whether or not the aspect ratio of the basic layout object shall be equal to that of the region of interest.

If only the range of allowed widths for the basic layout object is specified (case a), this attribute specifies that the height of the basic layout object shall be such that the aspect ratio of the region of interest is maintained.

If only the range of allowed heights for the basic layout object is specified (case b), this attribute specifies that the width of the basic layout object shall be such that the aspect ratio of the region of interest is maintained.

If both the ranges of allowed width and heights are specified (case c) the value of "aspect ratio flag" determines whether or not the aspect ratio of the region of interest shall be maintained during the determination of the dimensions of the basic layout object.

If neither the range of allowed heights nor the range of allowed widths is specified (case d), this attribute specifies that the width of the basic layout object shall be equal to the dimension of the available area in that direction and that the height is constrained to maintain the aspect ratio of the region of interest.

6.4   *Content architecture class attributes*

6.4.1   *Content architecture class*

The value of the attribute "content architecture class" of a basic component description that conforms to this Recommendation is an ASN.1 object identifier with the value

$$\{ 2\ 8\ 2\ 8\ 0 \}$$

6.4.2   *Content type*

The attribute "content type" cannot be used to specify the content architecture defined in this Recommendation.

6.5   *Interaction with document architecture attributes*

The value 'concatenate' of the layout directive attribute "concatenation" is ignored. This attribute is not taken into account during the layout of the geometric graphics content.

The layout directive attribute "indivisibility" may be ignored. It provides no additional constraint for the geometric graphics content layout process.

7   Geometric graphics content portions attributes

7.1   *Common coding attributes*

The value of the content portion attribute "type of coding" of a content portion description that conforms to this Recommendation is an ASN.1 object identifier with the value

$$\{ 2\ 8\ 3\ 8\ 0 \}.$$

7.2   *Content information*

The value of the content portion attribute "content information" of a content portion description that conforms to this Recommendation is an ASN.1 octet string representing a CGM conforming to the rules defined in ISO 8632-1 with the binary encoding defined in ISO 8632-3.

The relationship between this Recommendation and ISO 8632 is such that:

a) the string specified by the attribute "content information" in a geometric graphics content portion is a complete CGM as defined in ISO 8632-1 and ISO 8632-3;

b) any CGM as defined in ISO 8632-1 and ISO 8632-3 containing a single picture may be used as the value of a string specified by the attribute "content information" in a geometric graphics content portion.

Note - The presentation attributes specifying CGM defaults (see § 6.1.1) are provided for applying the factorization mechanism of an ODA environment to data that may be shared among several geometric graphics content portions. If a geometric graphics content portion is provided by importing a CGM into the ODA environment, then extreme care should be used in any attempt to assign this CGM as a data value for the content portion attribute "content information" and using the presentation attributes to change the default values of the CGM. The effects of changing the default values of such an imported CGM could quite possibly make the interpretation of the CGM non-sensible or impossible. The presentation attributes that should especially be considered are those that would cause a misunderstanding of the CGM data (for example, the presentation attribute "geometric graphics encoding announcer").

7.3 *Other coding attributes*

No other coding attributes are defined in this Recommendation.

8 Formal definitions of geometric graphics content architecture dependent data types

8.1 *Introduction*

This section contains the formal definitions, in ASN.1 notation (defined in Recommendation X.208), of data types corresponding to presentation and coding attributes that are applicable to geometric graphics content architectures.

These data types are:

a) the data type to represent the geometric graphics content architecture specific presentation attributes in basic layout components, presentation styles and default value lists;

b) the data type to represent the geometric graphics content architecture specific coding attributes in content portions;

c) the data type to represent the non-basic values of the geometric graphics content architecture presentation attributes in the document profile;

d) the data type to represent the non-basic values of the geometric graphics content architecture coding attributes in the document profiles;

e) the data type to represent the non-standard default values of geometric graphics content architecture presentation and coding attributes in the document profile.

8.2 *Representation of geometric graphics presentation attributes*

The data type "Geometric-Graphics-Attributes" contains a set of subordinate data types that specify the geometric graphics presentation attributes. Some of these subordinate data types are elementary but others are structured and themselves made up of subordinate data types. The format of these data types is given below.

The subset of subordinate data types that may occur within a particular instance of the data type "Geometric-Graphics-Attributes" depends upon the particular geometric graphics content architecture level that is specified.

Geo-Gr-Presentation-Attributes	{ 2 8 1 8 2 }
DEFINITIONS	::= BEGIN
EXPORTS	Geometric-Graphics-Attributes, Encoding-Announcer, Line-Rendition, Marker-Rendition, Text-Rendition, Filled-Area-Rendition, Edge-Rendition, Colour-Representations, Transparency-Specification Transformation-Specification, Region-Of-Interest, Picture-Orientation, Picture-Dimensions, ASF-Type, VDC-Pair, One-Of-Four-Angles;

Geometric-Graphics-Attributes	::=	SET {
encoding-announcer	[0]	IMPLICIT Encoding-Announcer OPTIONAL,
line-rendition	[1]	IMPLICIT Line-Rendition OPTIONAL,
marker-rendition	[2]	IMPLICIT Marker-Rendition OPTIONAL,
text-rendition	[3]	IMPLICIT Text-Rendition OPTIONAL,
filled-area-rendition	[4]	IMPLICIT Filled-Area-Rendition OPTIONAL,
edge-rendition	[5]	IMPLICIT Edge-Rendition OPTIONAL,
colour-representations	[6]	IMPLICIT Colour-Representations OPTIONAL,
transparency-specification	[7]	IMPLICIT Transparency-Specification OPTIONAL
transformation-specification	[8]	IMPLICIT Transformation-Specification OPTIONAL,
region-of-interest	[9]	Region-Of-Interest OPTIONAL,
picture-orientation	[10]	IMPLICIT Picture-Orientation OPTIONAL,
picture-dimensions	[11]	Picture-Dimensions OPTIONAL}

Encoding-Announcer   ::=   OCTET STRING
-- octet string representing the binary
-- encoding of any ordered set of CGM elements
-- identified in § 6.1.1.1

Line-Rendition   ::=   SEQUENCE {
   individual-part   [0]   IMPLICIT OCTET STRING OPTIONAL,
-- octet string representing the binary
-- encoding of any ordered set of CGM elements
-- identified in § 6.1.1.2 up to parameter
-- "line aspect source flags"
   asf-part   [1]   IMPLICIT SEQUENCE{
      line-type-asf   ASF-Type,
      line-width-asf   ASF-Type,
      line-colour-asf   ASF-Type
      } OPTIONAL,
   bundle-part   [2]   IMPLICIT SEQUENCE OF SEQUENCE {
      bundle-index   INTEGER
      bundle-representation   OCTET STRING } OPTIONAL
-- octet string representing the binary encoding
-- of the CGM elements
-- LINE TYPE, LINE WIDTH and
-- LINE COLOUR, as identified in § 6.1.1.2
}

Marker-Rendition   ::=   SEQUENCE{
   individual-part   [0]   IMPLICIT OCTET STRING OPTIONAL,
-- octet string representing the binary encoding
-- of any ordered set of CGM elements
-- identified in § 6.1.1.3 up to parameter
-- "marker aspect source flags"
   asf-part   [1]   IMPLICIT SEQUENCE{
      marker-type-asf   ASF-Type,
      marker-size-asf   ASF-Type,
      marker-colour-asf   ASF-Type
      } OPTIONAL,
   bundle-part   [2]   IMPLICIT SEQUENCE OF SEQUENCE {
      bundle-index   INTEGER
      bundle-representation   OCTET STRING } OPTIONAL
-- octet string representing the binary
-- encoding of the CGM elements
-- MARKER TYPE
-- MARKER SIZE and MARKER COLOUR,
-- as identified in § 6.1.1.3
}

Text-Rendition   ::=   SEQUENCE {
   individual-part   [0]   IMPLICIT OCTET STRING OPTIONAL,
-- octet string representing the binary
-- encoding of any ordered set of CGM elements
-- identified in § 6.1.1.4 up to
-- parameter "text aspect source flags"

```
asf-part                                    [1]   IMPLICIT SEQUENCE {
    text-font-asf                                     ASF-Type,
    text-precision-asf                                ASF-Type,
    character-expansion-factor-asf                    ASF-Type,
    character-spacing-asf                             ASF-Type,
    text-colour-asf                                   ASF-Type
                                                        } OPTIONAL,
bundle-part                                 [2]   IMPLICIT SEQUENCE OF SEQUENCE {
    bundle-index                                      INTEGER
    bundle-representation                             OCTET STRING } OPTIONAL
                                            -- octet string representing the binary
                                            -- encoding of the CGM elements
                                            -- TEXT FONT INDEX, TEXT PRECISION
                                            -- CHARACTER EXPANSION FACTOR,
                                            -- CHARACTER SPACING and
                                            -- TEXT COLOUR,
                                            -- as identified in § 6.1.1.4

Filled-Area-Rendition                  ::=  SEQUENCE {
    individual-part                         [0]   IMPLICIT OCTET STRING OPTIONAL,
                                            -- octet string representing the binary
                                            -- encoding of any ordered set of CGM
                                            -- elements identified in § 6.1.1.5,
                                            -- up to parameter "pattern table specifications"
    pattern-table-part                      [1]   IMPLICIT SEQUENCE OF
                                                      Pattern-Table-Element OPTIONAL

Pattern-Table-Element                  ::=  -- octet string representing the binary encoding
                                            -- of the CGM elements
                                            -- PATTERN TABLE, as identified in § 6.1.1.5
    asf-part                                [2]   IMPLICIT SEQUENCE{
        interior-style-asf                            ASF-Type,
        fill-colour-asf                               ASF-Type,
        hatch-index-asf                               ASF-Type,
        pattern-index-asf                             ASF-Type
                                                        } OPTIONAL,
    bundle-part                             [3]   IMPLICIT SEQUENCE OF SEQUENCE{
        bundle-index                                  INTEGER
        bundle-representation                         OCTET STRING } OPTIONAL
                                            -- octet string representing the binary encoding
                                            -- of the CGM elements INTERIOR STYLE
                                            -- FILL COLOUR,
                                            -- HATCH INDEX and PATTERN INDEX as
                                            -- identified in § 6.1.1.5
                                                        }

Edge-Rendition                         ::=  SEQUENCE {
    individual-part                         [0]   IMPLICIT OCTET STRING OPTIONAL,
                                            -- octet string representing the binary encoding
                                            -- of any ordered set of CGM elements
                                            -- identified in § 6.1.1.6 up to parameter
                                            -- "edge aspect source flags"
    asf-part                                [1]   IMPLICIT SEQUENCE {
        edge-type-asf                                 ASF-Type,
        edge-width-asf                                ASF-Type,
        edge-colour-asf                               ASF-Type
                                                        } OPTIONAL,
    bundle-part                             [2]   IMPLICIT SEQUENCE OF SEQUENCE {
        bundle-index                                  INTEGER,
        bundle-representation                         OCTET STRING } OPTIONAL
                                            -- octet string representing the binary encoding
                                            -- of the CGM elements EDGE TYPE,
                                            -- EDGE WIDTH and EDGE COLOUR as
                                            -- identified in § 6.1.1.6
                                                        }
```

ASF-Type	::=	INTEGER { bundle(0), individual(1) }
Colour-Representations	::=	SEQUENCE {
background-colour		[0] IMPLICIT OCTET STRING OPTIONAL,
		-- *octet string representing the binary encoding*
		-- *of the CGM element BACKGROUND COLOUR*
		-- *as identified in § 6.1.1.7*
colour-table-part		[1] IMPLICIT SEQUENCE OF {
		Colour-Table-Element OPTIONAL
Colour-Table-Element	::=	-- *octet string representing the binary encoding*
		-- *of the CGM element COLOUR TABLE, as*
		-- *identified in § 6.1.1.7*
		}
Transparency-Specification	::=	OCTET STRING
		-- *octet string representing the binary encoding*
		-- *of any ordered set of the CGM elements*
		-- *identified in § 6.1.1.8*
Transformation-Specification	::=	OCTET STRING
		-- *octet string representing the binary encoding*
		-- *of any ordered set of CGM elements*
		-- *identified in § 6.1.1.9*
Region-Of-Interest	::=	CHOICE {
automatic		[0] IMPLICIT NULL,
rectangle		[1] IMPLICIT SEQUENCE { VDC-Pair, VDC-Pair }
		}
VDC-Pair	::=	OCTET STRING
		-- *octet string representing the binary encoding*
		-- *of a CGM Virtual Device Coordinate pair*
Picture-Orientation	::=	One-Of-Four-Angles
One-Of-Four-Angles	::=	INTEGER{d0(0),
		d90(1),
		d180(2),
		d270(3) }
Picture-Dimensions	::=	CHOICE {
width-controlled		[0] IMPLICIT SEQUENCE {
minimum-width		INTEGER,
preferred-width		INTEGER },
height-controlled		[1] IMPLICIT SEQUENCE {
minimum-height		INTEGER,
preferred-height		INTEGER },
area-controlled		[2] IMPLICIT SEQUENCE {
minimum-width		INTEGER,
preferred-width		INTEGER,
minimum-height		INTEGER,
preferred-height		INTEGER,
aspect-ratio-flag		INTEGER { fixed(0), variable(1) }},
automatic		[3] IMPLICIT NULL
		}
		END

8.3   *Representation of coding attributes*

Geo-Gr-Coding-Attributes		{ 2 8 1 8 3 }
DEFINITIONS	::=	BEGIN
EXPORTS		Geo-Gr-Coding-Attributes;
Geo-Gr-Coding-Attributes	::=	SET { }
		-- *no geometric graphics coding attributes*
		-- *are defined in this Recommendation*
		END

8.4 *Representation of non-basic features and non-standard defaults*

```
Geo-Gr-Profile-Attributes           { 2 8 1 8 4 }

DEFINITION                          ::=  BEGIN

        EXPORTS                          Geo-Gr-Presentation-Feature,
                                         Geo-Gr-Coding-Attributes,
                                         Geo-Gr-Content-Defaults;

        IMPORTS                          Encoding-Announcer, Line-Rendition, Marker-Rendition,
                                         Text-Rendition, Filled-Area-Rendition, Edge-Rendition,
                                         Colour-Representations, Transparency-Specification,
                                         Transformation-Specification, Region-Of-Interest,
                                         Picture-Orientation, Picture-Dimensions, ASF-Type,
                                         VDC-Pair, One-Of-Four-Angles
                                             FROM Geo-Gr-Presentation-Attributes;
                                                                         -- see § 8.2

Geo-Gr-Presentation-Feature         ::=  NULL

                                         -- no non-basic values are expected to be defined
                                         -- for the geometric graphics presentation attributes
                                         -- in this Recommendation

Geo-Gr-Coding-Attribute             ::=  NULL

                                         -- no non-basic values are expected to be defined for
                                         -- the geometric graphics coding attributes
                                         -- in this Recommendation

Geo-Gr-Content-Defaults             ::=  SET {

        encoding-announcer               [0]   IMPLICIT Encoding-Announcer OPTIONAL,
        line-rendition                   [1]   IMPLICIT Line-Rendition OPTIONAL,
        marker-rendition                 [2]   IMPLICIT Marker-Rendition OPTIONAL,
        text-rendition                   [3]   IMPLICIT Text-Rendition OPTIONAL,
        filled-area-rendition            [4]   IMPLICIT Filled-Area-Rendition OPTIONAL,
        edge-rendition                   [5]   IMPLICIT Edge-Rendition OPTIONAL,
        colour-representations           [6]   IMPLICIT Colour-Representations OPTIONAL,
        transparency-specification       [7]   IMPLICIT Transparency-Specification OPTIONAL,
        transformation-specification     [8]   IMPLICIT Transformation-Specification OPTIONAL,
        region-of-interest               [9]            Region-Of-Interest OPTIONAL,
        picture-orientation              [10]  IMPLICIT Picture-Orientation OPTIONAL,
        picture-dimensions               [11]           Picture-Dimensions OPTIONAL }

                                         END
```

9 Content layout process

This section describes a content layout process for basic logical objects associated with content architectures of type geometric graphics.

Its purpose is to aid understanding of the semantics of the presentation attributes by describing the required results of such a process. However, it is not intended to specify any process that might be carried out in a particular implementation to achieve these results.

9.1 *Introduction*

9.1.1 *Purpose*

The content layout process describes a process of laying out the geometric graphics content into an allocated area. This area is referred to as the available area and is determined by the document layout process described in the Recommendation T.412.

The purpose of the content layout process is to convert content associated with basic logical components into content associated with basic layout objects.

The content layout process results in the creation of a basic layout object into which the content shall be positioned. The dimensions of the basic layout object are returned to the document layout process which determines the precise position of that basic layout object within the available area.

9.1.2 *Available area*

The content layout process is constrained by the available area. The maximum dimensions that a basic layout object can take are constrained by the dimensions of the available area.

During the layout of content associated with a basic logical component into a basic layout object, the following cases can occur:

a)   the formatted processable content fits into the dimensions of the available area;

b)   the formatted processable content does not fit into the dimensions of the available area; in this case, a new available area is required.

9.1.3 *Presentation attributes*

The content layout process takes into account the presentation attributes applying to the basic logical object with which the content is associated. The content layout process takes also into account the region of interest that may depend on CGM elements in the content portion.

The presentation attributes applying to the content layout process can be specified in the generic layout structure and presentation styles. The values of these presentation attributes are determined according to the defaulting rules specified in the Recommendation T.412.

9.1.4 *Geometric graphics content architecture classes*

The content layout process is specified for basic logical objects associated with the formatted processable form geometric graphics content architecture class. The content layout process does not modify the form of the content.

9.1.5 *Layout of the content*

For the geometric graphics content architecture class, one case of laying out the content of basic logical objects into layout objects is possible:

–   single basic logical object to single basic layout object: the content associated with a single basic logical object can be laid out into a single basic layout object and is the only content associated with this basic layout object.

9.2 *Content layout process for formatted processable content architecture class*

Determination of the dimensions of the basic layout object depends on the value of the presentation attribute "picture dimensions" (the four possible cases are illustrated in Figures 4/T.418 to 7/T.418):

a)   The presentation attribute "picture dimensions" specifies a value for the parameter "width controlled".

In this case the width of the picture will be within the range specified by the originator.

The determination of the basic layout object dimensions is constrained by the range of allowed widths given by the value of the parameter "width controlled", the dimensions of the available area and the aspect ratio of the region of interest.

The dimensions of the basic layout object shall be determined such that: the basic layout object fits into the available area; the aspect ratio of the basic layout object is the same as that of the region of interest; and the width of the basic layout object has a value that is within the range of allowed widths. The width of the basic layout object shall in addition be determined such that the deviation from the value of "preferred width", specified by the parameter "width controlled" is as small as possible.

b)   The presentation attribute "picture dimensions" specifies a value for the parameter "height controlled".

In this case the height of the picture will be within the range specified by the originator.

The determination of the basic layout object dimensions is constrained by the range of allowed heights given by the value of the parameter "height controlled", the dimensions of the available area and the aspect ratio of the region of interest.

The dimensions of the basic layout object shall be determined such that: the basic layout object fits into the available area; the aspect ratio of the basic layout object is the same as that of the region of interest; and the height of the basic layout object has a value that is within the range of allowed heights. The height of the basic layout object shall in addition be determined such that the deviation from the value of "preferred height", specified by the parameter "height controlled" is as small as possible.

c)   The presentation attribute "picture dimensions" specifies a value for the parameter "area controlled".

In this case the dimensions of the picture will be within the range specified by the originator. In particular, this can be used to ensure that a picture will have a fixed size.

The determination of the basic layout object dimensions is constrained by the range of allowed heights and widths given by the value of the parameter "area controlled", the dimensions of the available area and, depending on the value of the sub-parameter "aspect ratio flag" of the parameter "area controlled", by the aspect ratio of the region of interest.
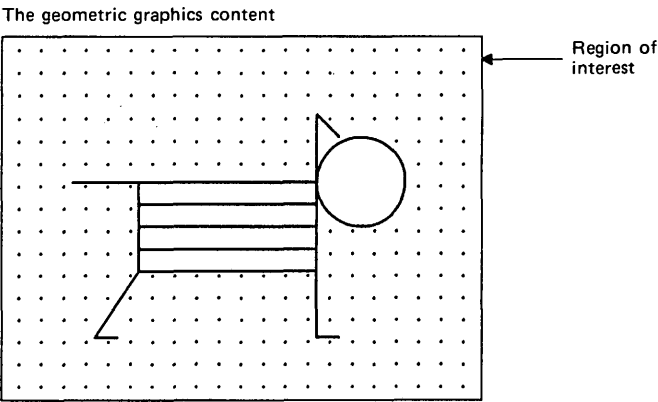
The dimensions of the basic layout object shall be determined such that: the basic layout object fits into the available area; the width of the basic layout object has a value that is within the range of allowed widths; and the height of the basic layout object has a value that is within the range of allowed heights. If the value of the sub-parameter "aspect ratio flag" is 'fixed' there is the further constraint to the basic layout object dimensions, that the aspect ratio of the basic layout object shall be the same as that of the region of interest. Both the width and height of the basic layout object shall additionally be chosen such, that their deviations from their preferred values, specified by the parameter "area controlled" are both as small as possible.

d)   The presentation attribute "picture dimensions" specifies a value for the parameter "automatic".

In this case the picture dimensions are automatically adjusted to the page layout.

The determination of the basic layout object dimensions is constrained by the dimensions of the available area and the aspect ratio of the region of interest.

The dimensions of the basic layout object shall be determined such that: the basic layout object fits into the available area; the width of the basic layout object is given the same value as the dimension of the available area in that direction; and the height of the basic layout object is determined such, that the aspect ratio of the basic layout object dimensions is the same as that of the region of interest.

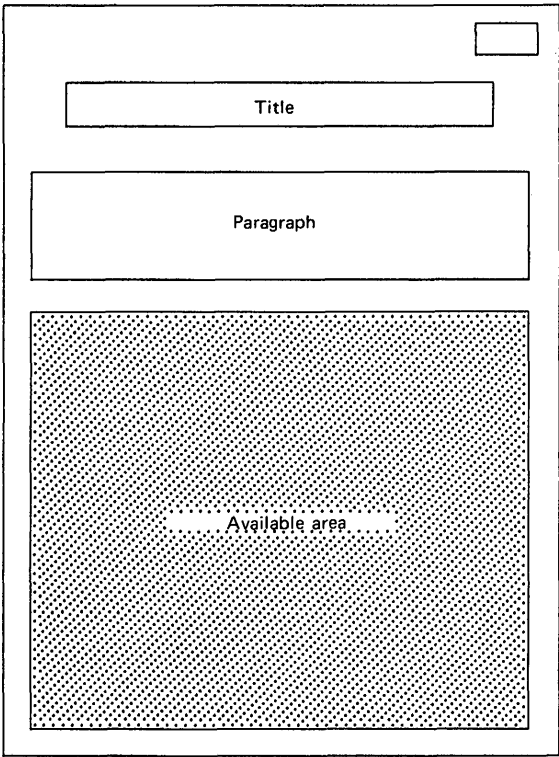If the given constraints cannot be met, then no dimensions of the basic layout object are determined.

If the SCALING MODE is 'metric', it is required that the values of the parameter set "picture size" is the equivalent value in SMUs of the specified metric size, taking into account the document profile attribute "unit scaling".

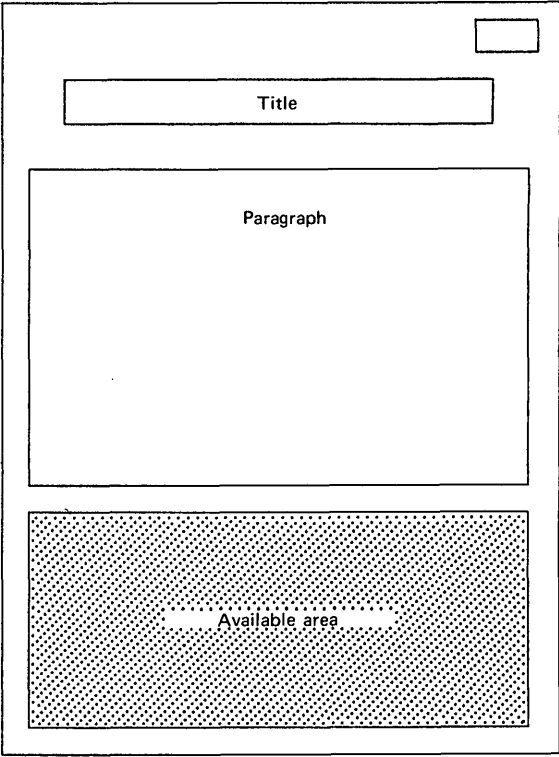The dimensions of a basic layout object are restricted to integral multiples of 1 SMU.

The presentation attribute "picture orientation" may rotate the region of interest. The rotated region of interest is used for the calculation of the basic layout object dimensions.

The geometric graphics content

Region of interest

Assumed page layout A

Title

Paragraph

Available area

Alternate assumed page layout B
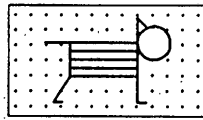

Title

Paragraph

Available area

T0802660-89

FIGURE 4/T.418

**Diagrams used to illustrate the process of determining
the basic layout object dimensions**

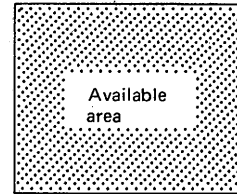Value of presentation attribute "picture dimensions": automatic

(Page layout A)

● *Initial constraints*

Aspect ratio
of region interest

Width of
available area
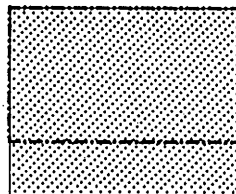
Dimensions of
available area

● *Allowable picture dimensions*

Height = width of available
area/aspect ratio

Width = width of available area

● *Basic layout object dimensions determined*

Dimensions of the basic
layout object

Dimensions of the
available area

● *Basic objects laid out, positioned and imaged*

Note — In this example, the positioning
of these basic layout objects assumes
normal fill order, the attribute "block
alignment" has value 'centered' and a
certain separation between two consecutive
blocks.

Available
area for next
basic object
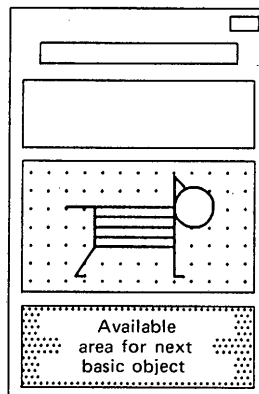
T0802670-89

FIGURE 5/T.418

**Layout process for the presentation attribute "picture dimensions"
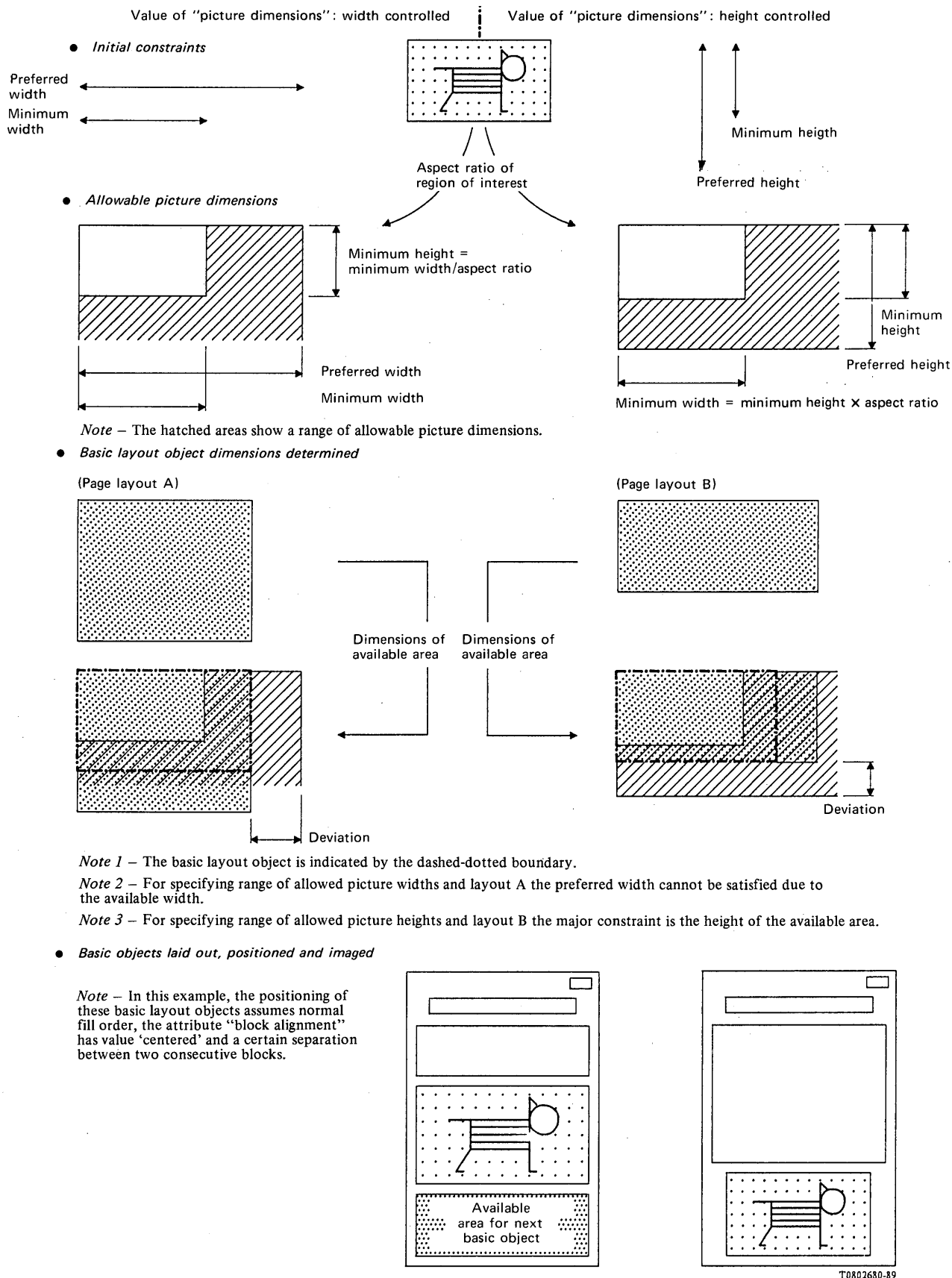when a value is specified for the parameter "automatic"**

FIGURE 6/T.418

Layout process for the presentation attribute "picture dimensions"
when a value is specified for the parameter
"width controlled" or "height controlled"

Value of presentation attribute "picture dimensions": area controlled
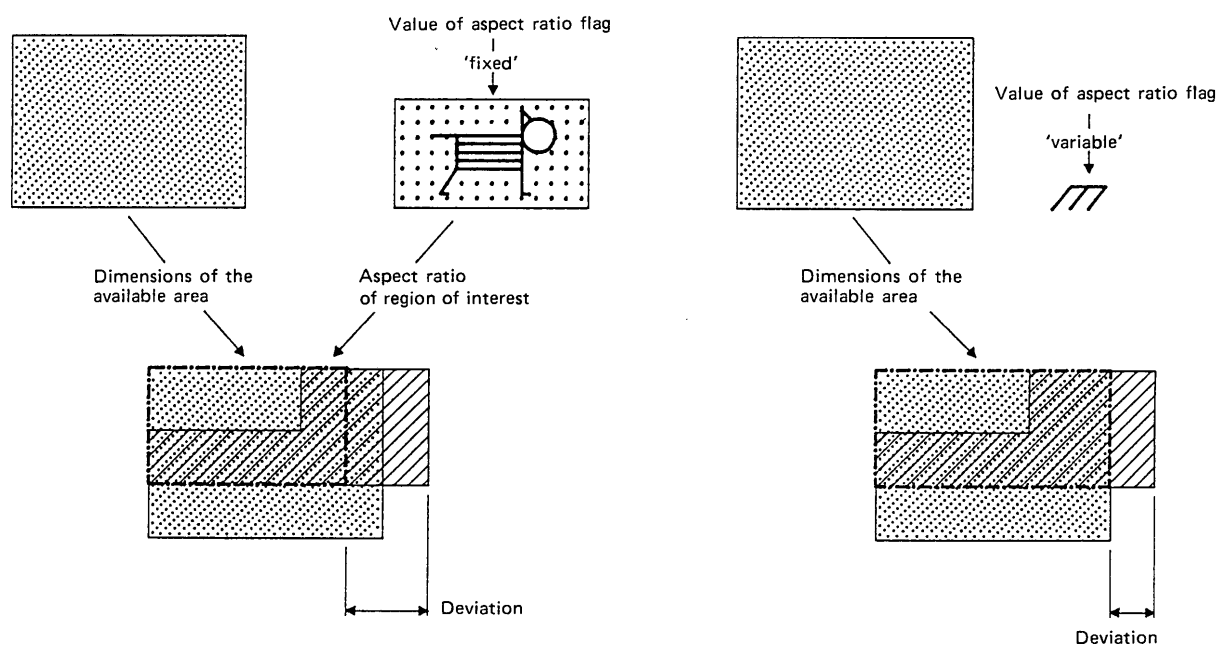
● *Initial constraints*

Preferred width

Minimum width

Minimum height

Prefered height
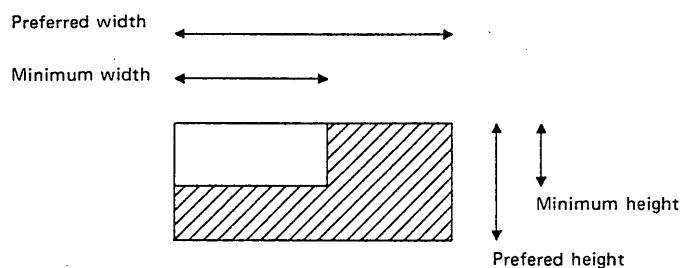
Note – The hatched area shows a range of allowable picture dimensions.

● *Allowable picture dimensions*

The allowable picture dimensions are completely determined by the initial constraints.

● *Basic layout dimensions determined (page layout A is used)*

Value of aspect ratio flag
|
'fixed'

Value of aspect ratio flag
|
'variable'

Dimensions of the
available area

Aspect ratio
of region of interest

Dimensions of the
available area

Deviation

Deviation

Note – The basic layout objects is indicated by the dashed-dotted boundary.

● *Basic objects positioned, laid out and imaged*

Note – In this example, the positioning of these basic layout objects assumes normal fill order, the attribute "block alignement" has value 'centered' and a certain separation between two consecutive blocks.
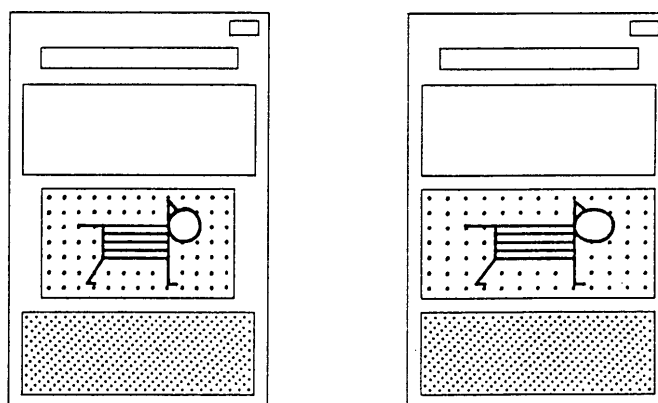
T0802690-89

FIGURE 7/T.418

**Layout process for the presentation attribute "picture dimensions"
when a value is specified for the parameter "area controlled"**

10   Content imaging process

This section describes a content imaging process for basic layout objects associated with content architectures of type geometric graphics.

Its purpose is to aid understanding of the semantics of the presentation attributes by describing the required results of such a process. However, it is not intended to specify any process that might be carried out in a particular implementation to achieve these results.

10.1   *Introduction*

The content imaging process is only concerned with the layout structures, the presentation styles and the content of basic layout components conforming to this Recommendation.

The content imaging process is applicable to the formatted processable form geometric graphics content architecture class.

10.2   *Content imaging process for formatted processable form content architecture class*

This section describes how the various shared presentation attributes and CGM elements determine the image of the content.

The imaging process is divided into two parts:

a)   initialization;

b)   imaging.

10.2.1   *Initialization of the imaging process*

At the start of the imaging of a geometric graphics content portion the imaging process is set to a default state. The default state of the imaging process is defined by the presentation attributes specifying CGM defaults (see § 6.1.1) except those parameters overwritten by CGM elements explicitly specified in the geometric graphics content portion.

The imaging process uses bundle representations, pattern representations and colour representations specified by the bundle specification, pattern table specification and colour table specification parameter values, respectively, of the geometric graphics presentation attributes "line rendition", "marker rendition", "text rendition", "filled area rendition", "edge rendition" and "colour representations" in conjunction with the defaults for these representations defined in Table 2/T.418, 3/T.418 and 4/T.418. After being set to its default state, the imaging process proceeds as if the CGM default have been explicitly specified in the content portion by CGM elements.

10.2.2   *Imaging*

Within geometric graphics content, geometric graphics elements are positioned in a Virtual Device Coordinate Space using a Virtual Device Coordinate system. For each basic layout object a part of the Virtual Device Coordinate Space specified by the region of interest is imaged according to the geometric graphics positioning principles (see § 5). No part of the graphical image which extends beyond the boundaries of the basic layout object is imaged.

The imaging process ignores the elements SCALING MODE as the required dimensions, and hence the aspect ratio, of the basic layout object have already been determined appropriately by the layout process.

In the case that the layout texture of the basic layout object specifies the combination "colourless, transparent" the picture descriptor element BACKGROUND COLOUR is ignored.

The support of external and escape elements is not required. A valid interpretation is to ignore them. The use of private unregistered escapes is not supported in open systems interchange.

Negative values of parameters of CGM elements are allowed. However, a valid interpretation is to ignore them; i.e. the use of private unregistered values is not supported in open systems interchange.

The image of the graphic elements is as specified by ISO 8632-1.

11   Definition of geometric graphics content architecture classes

There is only one geometric graphics content architecture class. This content architecture class provides for formatted processable form content. It is characterized as follows:

Content architecture class:	Formatted processable form
CGM elements:	All defined in ISO 8632-1 and ISO 8632-3,
Type of coding:	As defined in ISO 8632-3,
Geometric graphics presentation attributes:	All listed in § 6,
CGM attributes:	All CGM attributes.

ANNEX A

(to Recommendation T.418)

(Informative)

Summary of ASN.1 object identifiers

Values of ASN.1 object identifiers are assigned in various sections of this Recommendation. These assignments are summarized in Table A-1/T.418.

TABLE A-1/T.418

Summary of ASN.1 object identifiers

ASN.1 object identifier value	Description	Paragraph
{ 2 8 2 8 0 }	Value of attribute "content architecture class"	6.4.1
{ 2 8 3 8 0 }	Value of attribute "type of coding"	7.1
{ 2 8 1 8 2 }	Identifies module Geometric graphics presentation attributes	8.2
{ 2 8 1 8 3 }	Identifies module Geometric graphics coding attributes	8.3
{ 2 8 1 8 4 }	Identifies module Non-basic features and non-standard defaults	8.4

ANNEX B

(to Recommendation T.418)

(Informative)

Recommendations for the development of geometric graphics content architecture levels in document application profiles

This Annex provides an example of the definition of a geometric graphics content architecture level. This content architecture level has been defined in accordance with the rules specified in Recommendation T.411:

- GFP-0 is an example of a content architecture level belonging to the formatted processable form content architecture class. GFP-0 is mainly based on the minimum capabilities suggested in ISO 8632-1 and ISO 8632-3.

Note - The document application profile may have to specify additional rules for the use of these content architecture levels in particular applications. In particular, it may specify constraints to the values of parameters of CGM elements.

B.1 *Geometric graphics content architecture level GFP-0*

B.1.1 *Content architecture class*

GFP-0 is a content architecture level derived from the formatted processable form content architecture class.

B.1.2 *Presentation attributes*

Attribute	Basic values	Non-basic values	Default values
Geometric graphics encoding announcer	see Table B-1/T.418	none	standard default value
Line rendition	see Table B-2/T.418	none	standard default value
Marker rendition	see Table B-3/T.418	none	standard default value
Text rendition	see Table B-4/T.418	none	standard default value
Filled area rendition	see Table B-5/T.418	none	standard default value
Edge rendition	see Table B-6/T.418	none	standard default value
Colour specification	see Table B-7/T.418	none	standard default value
Transparency specification	see Table B-8/T.418	none	standard default value
Transformation specification	see Table B-9/T.418	none	standard default value
Region of interest specification	rectangle, automatic	none	standard default value
Picture orientation	0°,90°,180°,270°	none	standard default value
Picture dimensions	width controlled, height controlled, area controlled, automatic	none	standard default value

Tables B-1/T.418 to B-9/T.418 provide the BASIC values of the parameters of the presentation attributes specifying CGM default.

The third column of each table shows the relationship of the basic values to the complete set of permissible values as defined in the main body of this Recommendation. In those cases where the basic values include all possible values allowed by this Recommendation the entry in the third column is labelled 'none'. For structured parameters the basic values and the relation to the complete set of permissible values of their components are given to indicate that all possible combinations of its components form the basic values and the complete set of permissible values of a parameter respectively.

TABLE B-1/T.418

Geometric graphics encoding announcer

Parameter	Basic values	Values not in the set of basic values
VDC type	integer, real	none
Integer precision	8, 16, 24, 32	none
Real precision	(floating point format, 9, 23), (floating point format, 12, 52), (fixed point format, 16, 16), (fixed point format, 32, 32)	none
Index precision	8, 16, 24, 32	none
Colour precision	8, 16, 24, 32	none
Colour index precision	8, 16, 24, 32	none
Maximum colour index	any integer ≥ 0 and ≤ 63	all other permissible values
Colour value extent	((0, 0, 0),(255, 255, 255))	all other permissible values
Colour selection mode	indexed, direct	none
VDC integer precision	16, 24, 32	none
VDC real precision	(floating point format, 9, 23), (floating point format, 12, 52), (fixed point format, 16, 16), (fixed point format, 32, 32)	none

TABLE B-2/T.418

Line rendition

Parameter	Basic values	Values not in the set of basic values
Line width specification mode	absolute, scaled	none
Line bundle index	1, 2, 3, 4, 5	all other permissible values
Line type	1(solid), 2(dash), 3(dot), 4(dash-dot), 5(dash-dot-dot)	all other permissible values
Line width		
(if scaled)	1, 0	all other permissible values
(if absolute)	0,001 × length of longest side VDC extent	all other permissible values
Line colour		
(if indexed)	1	all other permissible values
(if direct)	foreground	all other permissible values
Line aspect source flags		
line type ASF	bundled, individual	none
line width ASF	bundled, individual	none
line colour ASF	bundled, individual	none
Line bundle specifications		
line bundle index	as for individual	all other permissible values
line bundle representation		
line type	as for individual	all other permissible values
(if scaled)	as for individual	all other permissible values
(if absolute)	as for individual	all other permissible values
Line colour		
(if indexed)	as for individual	all other permissible values
(if direct)	as for individual	all other permissible values

TABLE B-3/T.418

Marker rendition

Parameter	Basic values	Values not in the set of basic values
Marker size specification Mode	Absolute, scaled	none
Marker bundle index	1,2,3,4,5	all other permissible values
Marker type	1(dot), 2(plus), 3(asterisk), 4(circle), 5(cross)	all other permissible values
Marker size (if scaled)	1, 0	all other permissible values
(if absolute)	0,01 × length of longest side of VDC extent	all other permissible values
Marker colour (if indexed)	1	all other permissible values
(if direct)	foreground	all other permissible values
Marker aspect source flags marker type ASF marker size ASF marker colour ASF	bundled, individual bundled, individual bundled, individual	none none none
Marker bundle specifications marker bundle index	as for individual	all other permissible values
Marker bundle representation marker type	as for individual	all other permissible values
marker size (if scaled)	as for individual	all other permissible values
(if absolute)	as for individual	all other permissible values
Marker colour (if indexed)	as for individual	all other permissible values
(if direct)	as for individual	all other permissible values

TABLE B-4/T.418

Text rendition

Parameter	Basic values	Values not in the set of basic values
Font list	list containing one element: any registered font capable of representing the nationality-independent character subset of Recommendation T.50	all other permissible values
Character set list character set type	94-character sets	all other permissible values
designation sequence tail	designation sequence tail that is registered for a character set which includes the nationality-independent subset of Recommendation T.50 in the positions specified in Recommendation T.50	all other permissible values
Character coding announcer	basic 7-bit	all other permissible values
Text bundle index	1, 2	all other permissible values
Text font index	1	all other permissible values
Text precision	string, character	all other permissible values
Character expansion factor	0,7 and 1,0	all other permissible values
Character spacing	0,0	all other permissible values
Text colour (if indexed)	1	all other permissible values
(if direct)	foreground	all other permissible values
Character height	0,01 × length of the longest side of VDC extent	all other permissible values
Character orientation	any pair of VDC vectors which have non-zero length, are not collinear and are parallel to the axes of the VDC space	all other permissible values

Parameter	Basic values	Values not in the set of basic values
Text path	right, left, up, down	none
Text alignment		
horizontal alignment	normal horizontal, left, centre, right	all other permissible values
vertical alignment	normal vertical, top, base, bottom	all other permissible values
continuous horizontal alignment	n/a	all other permissible values
continuous vertical alignment	n/a	all other permissible values
Character set index	1	all other permissible values
Alternate character set index	1	all other permissible values
Text aspect source flags		
text font ASF		
text precision ASF	bundled, individual	none
character expansion factor ASF	bundled, individual	none
character spacing ASF	bundled, individual	none
text colour ASF	bundled, individual	none
	bundled, individual	none
Text bundle specifications		
Text bundle index	as for individual	all other permissible values
Text bundle representation		
Text font index	as for individual	all other permissible values
Text precision	as for individual	all other permissible values
Character expansion factor	as for individual	all other permissible values
Character spacing	as for individual	all other permissible values
Text colour		
(if indexed)	as for individual	all other permissible values
(if direct)	as for individual	all other permissible values

TABLE B-5/T.418

Filled area rendition

Parameter	Basic values	Values not in the set of basic values
Fill bundle index	1, 2, 3, 4, 5	all other permissible values
Interior style	hollow, solid, pattern, hatch, empty	none
Fill colour		
(if indexed)	1	all other permissible values
(if direct)	foreground	all other permissible values
Hatch index	1 (horizontal equally spaced parallel lines) 2 (vertical equally spaced parallel lines) 3 (positive slope equally spaced parallel lines) 4 (negative slope equally spaced parallel lines) 5 (horizontal/vertical crosshatch) 6 (positive slope/negative slope crosshatch)	all other permissible values
Pattern index	1	all other permissible values
Fill reference point	(0;0)	all other permissible values
Pattern size height vector x component	0	all other permissible values
height vector y component	height of default VDC extent	all other permissible values
width vector x component	width of default VDC extent	all other permissible values
width vector y component	0	all other permissible values
Pattern table specifications pattern table index	1	all other permissible values
nx (number of columns in pattern)	1	all other permissible values
ny (number of rows in pattern)	1	all other permissible values
local colour precision	0	all other permissible values

Parameter	Basic values	Values not in the set of basic values
colour index array (if indexed)	one cell containing colour index 1	all other permissible values
value array (if direct)	one cell containing direct colour value 'foreground'	all other permissible values
Fill aspect source flags interior style ASF	bundled, individual	none
fill colour ASF	bundled, individual	none
hatch index ASF	bundled, individual	none
pattern index ASF	bundled, individual	none
Fill bundle specifications Fill bundle index	as for individual	all other permissible values
Fill bundle representation interior style	as for individual	all other permissible values
Fill colour (if indexed)	as for individual	all other permissible values
(if direct)	as for individual	all other permissible values
Hatch index	as for individual	all other permissible values
Pattern index	as for individual	all other permissible values

TABLE B-6/T.418

Edge rendition

Parameter	Basic values	Values not in the set of basic values
Edge width specification mode	absolute, scaled	none
Edge visibility	off	all other permissible values
Edge bundle index	1, 2, 3, 4, 5	all other permissible values
Edge type	1(solid), 2(dash), 3(dot), 4(dash-dot), 5(dash-dot-dot)	all other permissible values
Edge width (if scaled)	1,0	all other permissible values
(if absolute)	0,001 × length of longest side of VDC extent	all other permissible values
Edge colour (if indexed)	1	all other permissible values
(if direct)	foreground	all other permissible values
Edge aspect source flags edge type ASF edge width ASF edge colour ASF	bundled, individual bundled, individual bundled, individual	none none none
Edge bundle specifications edge bundle index	as for individual	all other permissible values
Edge bundle representation edge type	as for individual	all other permissible values
edge width (if scaled)	as for individual	all other permissible values
(if absolute)	as for individual	all other permissible values
Edge colour (if indexed)	as for individual	all other permissible values
(if direct)	as for individual	all other permissible values

TABLE B-7/T.418

Colour representations

Parameter	Basic values	Values not in the set of basic values
Background colour	background	all other permissible values
Colour table specifications starting index	0,1	all other permissible values
colour list	any list containing one or two (only if starting colour index = 0) direct colour values each either direct colour values each either 0 or foreground if replacing colour table entry 1	all other permissible values

TABLE B-8/T.418

Transparency specification

Parameter	Basic values	Values not in the set of basic values
Transparency	on	all other permissible values
Auxiliary colour (if indexed)	n/a	all other permissible values
(if direct)	n/a	all other permissible values

TABLE B-9/T.418

Transformation specification

Parameter	Basic values	Values not in the set of basic values
VDC extent	any pair of virtual device coordinates defining a rectangle	none
Clip indicator	off, on	none
Clip rectangle	any pair of virtual device coordinates defining a rectangle	none

B.1.3 *Content portion attributes*

No content portion attributes other that the document architecture attributes "type of coding" and "content information" are defined in this Recommendation.

ANNEX C

(to Recommendation T.418)

(Informative)

Basic differences between character primitives in the geometric graphics and the content of a basic component structured according to the character content architectures defined in Recommendation T.416

This Annex identifies the basic differences between character strings regarded as graphical primitives and character content.

Within the geometric graphics content architecture (GGCA) the graphical primitive elements which may be used to describe a picture, include a text primitive. The text primitive permits the inclusion of textual information in geometric graphics pictures.

The most important basic differences between the text within the two different content architectures are as follows:

a) within GGCA each primitive is conceptually independent of any other within the same content portion. Within the character content architecture (CCA) all content is a single text string;

b) within GGCA a text primitive may be positioned anywhere in the VDC Space. Within CCA the positioning of characters is sequential in nature and, therefore, the position at which a character is imaged is derived from the position at which the predecessor was imaged;

c) within GGCA the size of the basic layout object is not affected by the text within the content portion. Within CCA the size of the basic layout object is totally dependent on the amount of text within the content portion;

d) within GGCA the orientation of a text primitive may be set to any angle relative to the Virtual Device Coordinate system first axis. Within CCA the character path must be at 0°, 90°, 180° or 270° relative to the horizontal axis of the layout object;

e) within GGCA the character size is specified by the CGM attribute elements CHARACTER HEIGHT and CHARACTER EXPANSION FACTOR and is conceptually independent of the font. allows the aspect ratio of the character to be changed. Within CCA the character size is defined by the font;

f) within GGCA a text primitive is a geometric element and may undergo geometric transformation such as scaling and arbitrary rotation. Within CCA no such functionality exists;

g) within GGCA text primitives the effect of embedded control functions are not standardized (eg: <CR>, <LF>, <HT>). These control functions may occur but there is no definition of their semantics. Within CCA such control functions have a standardized effect;

h) within GGCA there is a character attribute which permits the under-character space to be set or changed without constraint to a particular string of characters or a given line length. This permits the justification of either monospaced or proportionally spaced fonts. Within CCA, such justification can be specified as a requirement in the logical text, but resolves to embedded control functions working in unit of SMU in the laid out (formatted) text.