INTERNATIONAL STANDARD

ISO/IEC 8613-1

Second edition
1994-12-15

Information technology — Open Document Architecture (ODA) and Interchange Format: Introduction and general principles

Technologies de l'information — Architecture des documents ouverts (ODA) et format d'échange: Introduction et principes généraux

Reference number
ISO/IEC 8613-1:1994(E)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Foreword

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

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

International Standard ISO/IEC 8613-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 18, *Document processing and related communication*, in collaboration with ITU-T. The identical text is published as ITU-T Rec. T.411.

This second edition cancels and replaces the first edition (ISO 8613-1:1989), which has been technically revised.

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

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

Annexes C, D and E form an integral part of this part of ISO/IEC 8613. Annexes A, B, F and G are for information only.

Introduction

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

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

- Introduction and general principles;
- Document structures;
- Document profile;
- Open document interchange format;
- Character content architectures;
- Raster graphics content architectures;
- Geometric graphics content architectures;
- Formal specifications.

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

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

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

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

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

- Alternative representation
- Annex on use of MHS/MOTIS
- Colour
- Conformance Testing annex
- Document Application Profile Proforma and Notation
- Security
- Streams
- Styles
- Tiled raster graphics

In addition, a number of technical corrigenda have been applied to this series of ITU-T Recommendations | International Standard.

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

- Annex A (non-integral) gives references to other standards and registers;
- Annex B (non-integral) gives the relationship of this series of ITU-T Recommendations | International Standards with other CCITT Recommendations and International Standards;
- Annex C (integral) specifies the method of recording documents in conformance with this series of ITU-T Recommendations | International Standards on volume and file structures of interchangeable storage media in conformance with international standards for volume and file structures;
- Annex D (integral) gives the principles for the assignment of ASN.1 object identifier values for this series of ITU-T Recommendations | International Standards;
- Annex E (integral) describes the use of MHS to interchange documents conforming to this series of ITU-T Recommendations | International Standards;
- Annex F (non-integral) defines a standardized proforma for the specification of document application profiles based on this series of ITU-T Recommendations | International Standards;
- Annex G (non-integral) gives directions for the development of a conformance testing methodology for this series of ITU-T Recommendations | International Standards.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN DOCUMENT ARCHITECTURE (ODA) AND INTERCHANGE FORMAT: INTRODUCTION AND GENERAL PRINCIPLES

1 Scope

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

In the context of these Recommendations | International Standards, documents are 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, raster graphics elements, and geometric graphics elements, all potentially within one document.

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

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

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

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

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

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

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

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

This ITU-T Recommendation | International Standard:

- introduces this ITU-T T.410-Series of Recommendations | ISO/IEC 8613 as a whole;
- gives the references necessary for the ITU-T T.410-Series of Recommendations | ISO/IEC 8613;
- defines terms used in the context of the ITU-T T.410-Series of Recommendations | ISO/IEC 8613;
- presents the concepts of the document architecture;
- gives an overview of the ITU-T T.410-Series of Recommendations | ISO/IEC 8613;
- describes the inter-dependencies of the ITU-T T.410-Series of Recommendations | ISO/IEC 8613;
- defines conformance to the ITU-T T.410-Series of Recommendations | ISO/IEC 8613;
- gives rules for defining document application profiles.

2 Normative references

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

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation T.412 (1993) | ISO/IEC 8613-2:1994, *Information technology – Open Document Architecture (ODA) and Interchange Format: Document structures.*
- ITU-T Recommendation T.414 (1993) | ISO/IEC 8613-4:1994, *Information technology – Open Document Architecture (ODA) and Interchange Format: Document profile.*
- ITU-T Recommendation T.415 (1993) | ISO/IEC 8613-5:1994, *Information technology – Open Document Architecture (ODA) and Interchange Format: Open Document Interchange Format.*
- ITU-T Recommendation T.416 (1993) | ISO/IEC 8613-6:1994, *Information technology – Open Document Architecture (ODA) and Interchange Format: Character content architectures.*
- ITU-T Recommendation T.417 (1993) | ISO/IEC 8613-7:1994, *Information technology – Open Document Architecture (ODA) and Interchange Format: Raster graphics content architectures.*
- ITU-T Recommendation T.418 (1993) | ISO/IEC 8613-8:1994, *Information technology – Open Document Architecture (ODA) and Interchange Format: Geometric graphics content architectures.*

2.2 Paired Recommendations | International Standards equivalent in technical content

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

  ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1).*
- CCITT Recommendation X.209 (1988), *Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1).*

  ISO/IEC 8825:1990, *Information technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*
- CCITT Recommendation X.411 (1992), *Message Handling Systems: Message Transfer System: Abstract Service Definition and Procedures.*

  ISO/IEC 10021-4:1990, *Information technology – Text Communication – Message Oriented Text Interchange Systems (MOTIS) – Part 4: Message Transfer System: Abstract Service Definition and Procedures.*
- CCITT Recommendation X.420 (1992), *Message Handling Systems: Message Transfer System: Interpersonal Messaging System.*

  ISO/IEC 10021-7:1990, *Information technology – Text Communication – Message Oriented Text Interchange Systems (MOTIS) – Part 7: Interpersonal Messaging System.*

2.3 Additional references

- ISO 2022:1986, *Information processing – ISO 7-bit and 8-bit coded character sets – Code extension techniques.*
- ISO/IEC 8613-10:1991, *Information processing – Text and office systems – Office Document Architecture (ODA) and Interchange Format – Part 10: Formal specifications.*
- ISO/IEC 8632-1:1992, *Information technology – Computer graphics – Metafile for the storage and transfer of picture description information – Part 1: Functional specification.*
- ISO/IEC 8632-3:1992, *Information technology – Computer graphics – Metafile for the storage and transfer of picture description information – Part 3: Binary encoding.*
- ISO 8879:1986, *Information processing – Text and office systems – Standard Generalized Markup Language (SGML).*
- ISO 9069:1988, *Information processing – SGML support facilities – SGML Document Interchange Format (SDIF).*
- ISO 9293:1987, *Information processing – Volume and file structure of flexible disk cartridges for information interchange.*
- ISO/IEC 9541-1:1991, *Information technology – Font information interchange – Part 1: Architecture.*
- ISO 9660:1988, *Information processing – Volume and file structure of CD-ROM for information interchange.*

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.1 active position: The point at which the action specified by the next character is to be effected.

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 alternative description: A description that represents a basic object that is intended to be used by the recipient in lieu of the primary description of that basic object when the primary description cannot be processed.

3.4 alternative subtree: An alternative basic object description in conjunction with its associated content portion descriptions.

3.5 aspect ratio: The ratio of the dimension of a pel array in the direction of the pel path to the dimension in the direction of the line progression.

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

3.7 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.

3.8 authenticity: The property that the claimed data source can be verified to the satisfaction of the recipient.

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

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

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

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

3.13 basic measurement unit; BMU (abbreviation): 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.

3.14 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.

3.15 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.

3.16 block: A basic layout component that corresponds to a rectangular area within a frame or a page.

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

3.18 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 a layout object.

3.19 bottom right corner: The corner of a layout object that is most progressed both in the horizontal and vertical directions of a layout object.

3.20 centred:

a) 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.

b) 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 as a value in the sub-parameter "alignment" of the parameter "position" and in the attribute "block alignment".

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

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

3.23 character image: The human perceptible rendering of a character on a presentation medium.

NOTE – The term *glyph* in ISO/IEC 9541 is equivalent to the term *character image* in this Specification.

3.24 character orientation: The direction of the character base line relative to the characters.

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

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

3.27 character spacing (for constant spacing fonts only): The distance between the position points of successive character images when the inter-character space equals zero.

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

3.29 colour component: One of the dimensions of a colour space.

3.30 colour gamut: The range of colours a given system is capable of reproducing.

3.31 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.

3.32 component: An object or an object class.

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

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

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

3.36 confidentiality: The property that information is not made available or disclosed to unauthorized individuals, entities or processes.

NOTE – This property is limited here to preventing unauthorized semantic knowledge of a document or specified parts of it.

3.37 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.

3.38 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, a content portion description or a protected part description.

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

3.40 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.

3.41 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.

3.42 content editing process: The process that creates new content or modifies previous content.

3.43 content element: A basic element of the content of a document.

3.44 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.

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

3.46 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.

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

3.48 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 Selected Graphic Rendition (SGR) in character content architectures and Set Line Type in geometric graphics content architectures.

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

3.50 Cyan Magenta Yellow (Black); CMY(K) (abbreviation) colour space: Colour space based on the subtractive colour mixture of Cyan (C), Magenta (M) and Yellow (Y) primaries with the optional inclusion of Black (K) as indicated by the parentheses.

3.51 DAP-n data stream: An ODA data stream in which the data elements are in accordance with a particular document application profile named *n*, defined in accordance with this Specification.

3.52 data integrity: The property that data has not been altered or destroyed.

3.53 data structure: A set of data items and the relationship between 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 components, the styles, or the content portions concerned.

3.54 description: A constituent that corresponds to a structural element.

NOTE – A basic object may have several descriptions when alternative descriptions are used.

3.55 descriptor: A data structure representing the document profile, an object class description, a layout style, a presentation style, an object description, or a protected part description.

3.56 device space: A colour space where the coordinates are those used by a particular device in the measurement or rendition of colour.

3.57 digital signature: A form of seal associated with a specified part of a document which provides proof of uniqueness of the identity of the originator (source) who applied the seal; it supports non-repudiation of origin of the sealed (signed) part.

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

3.59 document application profile: The specification of a combination of features defined in ITU-T Rec. T.410-Series | ISO/IEC 8613, intended to form a subset to fulfil the requirements of an application.

3.60 document architecture:

a) Rules for defining the structure of documents, in terms of collections of constituents and the attributes of which these are composed.

b) 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 together with presentation styles and/or layout styles.

3.61 document architecture class: The rules for defining the structure and representation of documents in formatted form, processable form or formatted processable form.

3.62 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, protected parts but excludes the document profile.

3.63 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.

3.64 document class description: The specification of a document class.

3.65 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.

3.66 document layout root: The composite object of the specific layout structure at the highest level of the hierarchy.

3.67 document logical root: The composite object of the specific logical structure at the highest level of the hierarchy.

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

3.69 editing: The carrying out of operations associated with creation and modification of the structure and/or the content of a document.

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

3.71 end-aligned:

a) 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.

b) 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.

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

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

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

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

3.76 file: A named collection of information.

3.77 file section: For a file recorded over more than one volume, that part of the file that is recorded on any one volume.

NOTE – The term *file* in ISO 9293 is equivalent to the term *file section* in this Specification.

3.78 file space: The region within which a file section may be recorded on a volume.

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

3.80 fingerprint: A short and compact code that may be computed in order to characterize some specified information, with the property that it is not practicable to construct different information which would yield the same output.

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

3.82 font size: The height of the character images in a font.

3.83 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.

3.84 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.

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

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

3.87 generic content portion: A content portion associated with an object class.

3.88 generic content portion description: A content portion description associated with an object class description.

3.89 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.

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

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

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

NOTE – Geometric graphics elements include those describing primitive geometric shapes such as points, arcs, lines.

3.93 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).

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

NOTE – Three types of graphic elements are distinguished in the ITU-T Rec. T.410-Series | ISO/IEC 8613: graphic characters, raster graphics elements, and geometric graphics elements.

3.95 hard line terminator: A line terminator that is intended not to be removed in a reformatting process.

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

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

3.98 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.

3.99 implicit layout category: The layout category of a logical object which is used in the absence of an explicit specification of a layout category for that object.

3.100 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.

3.101 initial point:

a) 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 ITU-T Rec. T.416 | ISO/IEC 8613-6).

b) 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 ITU-T Rec. T.417 | ISO/IEC 8613-7).

3.102 integrity: Used here synonymously with data integrity.

3.103 intended recipient: A recipient of a document that is expected to receive or have access to the document.

3.104 interchange: The process of providing a document to a receiving person or device, by means of data communication or by exchange of storage media.

3.105 interchange data element: A data structure representing a constituent of a document.

3.106 interchange format: The rules for representing a document for the purpose of interchange.

3.107 interchange format class: A form of interchange format suitable to a specific application.

NOTE – In the ITU-T Rec. T.410-Series | ISO/IEC 8613, the defined classes differ by the ordering of the interchange data elements or by the coding.

3.108 interchangeable storage medium: Storage medium which can be used to interchange information by moving the medium from one information processing system to another.

3.109 inter-character space: An additional amount of spacing that is included between adjacent character images.

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

3.111 item identifier: A string of characters that precedes and is separated from the remainder of the first line of a basic component with content. An item identifier is used to identify the subsequent text.

3.112 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.

3.113 kern: The part of a character image which extends beyond its position point or escapement point.

3.114 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.

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

3.116 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.

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

NOTE – This is also referred to as formatting.

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

3.119 layout structure:

a) 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.

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

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

3.121 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.

3.122 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 degrees counterclockwise relative to the direction of the layout path.

3.123 line box: A rectangular area within which a sequence of character images are positioned.

3.124 line home position: The point within a line box that is used for positioning that line box.

3.125 line progression:

a) The direction of progression of successive line boxes within a basic layout object (character content architectures ITU-T Rec. T.416 | ISO/IEC 8613-6).

b) The direction of progression of successive lines of pels within a basic layout object (raster graphics content architectures ITU-T Rec. T.417 | ISO/IEC 8613-7).

3.126 line spacing:

a) The distance between two adjacent reference lines within a basic layout object (character content architectures ITU-T Rec. T.416 | ISO/IEC 8613-6).

b) The distance between two adjacent lines of pels within a basic layout object (raster graphics content architectures ITU-T Rec. T.417 | ISO/IEC 8613-7).

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

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

3.129 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.

3.130 logical structure:

a) 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.

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

3.131 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.

3.132 mandatory attribute: An attribute which, when applicable to a constituent, must be specified explicitly in the constituent.

3.133 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.

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

3.135 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.

3.136 non-repudiation of origin: The property that an originator can be proved to the satisfaction of a third party to be the source of a document or a specified part of it.

3.137 object: An element of the specific layout structure or of the specific logical structure.

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

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

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

3.141 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.

3.142 ODA data stream: A data stream conforming to the ITU-T Rec. T.410-Series | ISO/IEC 8613 in which the data elements representing constituents and attribute values are in accordance with ITU-T Rec. T.412, T.414 ISO/IEC 8613-2, 8613-4 and one or more of the ITU-T Rec. T.410-Series | ISO/IEC 8613 Specifications including:

- ITU-T Rec. T.416 | ISO/IEC 8613-6;
- ITU-T Rec. T.417 | ISO/IEC 8613-7;
- ITU-T Rec. T.418 | ISO/IEC 8613-8;

and any referenced standards.

3.143 Open Document Language; ODL (abbreviation): A Standard Generalized Markup Language (SGML, ISO 8879) application for representing documents conforming to ISO/IEC 8613.

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

3.145 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.

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

3.147 page coordinate system: An orthogonal coordinate 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.

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

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

3.150 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.

3.151 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 do not fully specify all specific structures that may be created.

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

3.153 pel path: The direction of progression of successive pels along a line within a basic layout object.

3.154 pel spacing: The distance between any two successive pels along a line within a basic layout object.

3.155 picture element; pel (abbreviation): The smallest graphic element that may be individually addressed within a picture (an alternative term for raster graphics element).

3.156 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.

3.157 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.

3.158 presentation: The operation of rendering the content of a document in a form perceptible to a human being.

3.159 presentation medium: The carrier of information in a form perceptible to a human being.

3.160 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.

3.161 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.

3.162 primary description: A description that represents a basic object that best represents the intent of the originator.

3.163 primary subtree: The basic object description in conjunction with its associated content portion descriptions, with the potential to be replaced by an alternative subtree.

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

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

3.166 processing: The carrying out of operations on a document, including editing, (re)formatting, presentation, filing and retrieval.

3.167 protected part: A constituent consisting of a part of the document that is sealed or enciphered, e.g. a sealed document profile or pre-enciphered document body part.

3.168 raster graphics element: An alternative term for a picture element (pel).

3.169 recipient: Any application or user receiving or having access to a document.

3.170 record: A sequence of bytes treated as a unit of information stored on a file.

standard for volume and file structure: An international standard that specifies the format of the recorded structures that contain descriptive information about the volume and the files recorded on an interchangeable storage medium.

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

3.172 reference colour space: Basic colour space within ODA relative to which relationships to colour spaces used in document and content architectures are specified.

NOTE – The reference colour space is defined by the Commission Internationale de l'Eclairage (CIE) Standard Colorimetric System (XYZ), 1931.

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

3.174 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.

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

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

3.177 resource-document: A generic-document containing one or more object class descriptions and their associated generic content descriptions referred to by one or more object class descriptions of another document.

3.178 retrieval: The recovery of previously filed information.

3.179 Red Green Blue; RGB (abbreviation) colour space: A colour space based on red, green and blue stimuli or primaries.

NOTES

1 The RGB values are not to be confused with the CIE $\overline{rgb}$ as defined in CIE colorimetry.

2 RGB values are negative in certain areas outside the gamut defined by the RGB primaries.

3.180 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 degrees counterclockwise relative to the direction of the layout path.

3.181 scaled measurement unit; SMU (abbreviation): 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.

3.182 seal (noun): Data associated with a specified part of a document by an originator, which a recipient (privileged recipient) may use to verify the integrity and authenticity of the specified part.

3.183 seal (verb): To associate a seal with a specified part of a document.

3.184 security domain: The set of resources subject to a single security policy.

3.185 security label: A marking of a document, which specifies the handling of the document according to the security policy in force.

3.186 security policy: The set of rules that specify the procedures and services required to maintain the intended level of security of a set of resources.

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

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

3.189 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.

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

3.191 spacing ratio: The ratio of line spacing to pel spacing.

3.192 specific layout structure: A set of layout objects and associated content portions.

3.193 specific logical structure: A set of logical objects and associated content portions.

3.194 start-aligned:

a) 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, for the line.

b) 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.

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

3.196 stream root category: A property of components that can be used to designate the layout objects into which the content associated with logical objects shall be placed.

3.197 stream root sub category: A property of components that can be used to designate the subordinate layout objects into which the content associated with logical objects shall be placed. The designated subordinate layout objects shall all be subordinates of a single layout object that has the property of being a stream root category layout object.

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

3.199 tabulation alignment: A layout or imaging process that causes the sequence of character images for a 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.

3.200 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.

3.201 tile: One element of a two dimensional array of non-overlapping rectangular regions of a pel array.

3.202 text unit: A data structure representing a content portion description.

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

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

3.205 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.

3.206 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.

3.207 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).

3.208 variable spacing: The characteristics of a font wherein the distance between the position point and the escapement point for different character images may be different.

3.209 volume: A dismountable physical unit of storage medium, for example a flexible disk cartridge.

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

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following definitions apply.

ASN.1	Abstract Syntax Notation One
BMU	Basic Measurement Unit
BNF	Backus-Naur-Form
CF	Character Formatted
CGM	Computer Graphics Metafile
CIE	Commission Internationale de l'Eclairage
CIELAB	Commission Internationale de l'Eclairage lightness-hue-chroma CIE 1976 $L^*a^*b^*$ colour space
CIELUV	Commission Internationale de l'Eclairage lightness-hue-chroma CIE 1976 $L^*u^*v^*$ colour space
CMY	Cyan Magenta Yellow
CMY(K)	Cyan Magenta Yellow Black
DAPPN	Document Application Profile Proforma Notation
DIS	Disallowed
ECMA	European Computer Manufacturer's Association
FDA	Formatted Document Architecture
FODA	Formal Specification of the Open Document Architecture
FPDA	Formatted Processable Document Architecture
ID	Identifier
IMCL	Information Modeling by Composition Language
IPM	Interpersonal Message
IPMS	Interpersonal Message System
LUT	Colour LookUp Table
MHS	Message Handling System
mm	Millimeter
MOTIS	Message Oriented Text Interchange System
MTS	Message Transfer System
MUL	Multiple (Repetition)
NIST	National Institute of Standards and Technology
NOVR	No Override

ODA	Open Document Architecture
ODIF	Open Document Interchange Format
ODL	Open Document Language
PDA	Processable Document Architecture
pel	Picture Element
PERM	Permitted
PMUL	Permitted Multiple (Optional Repetition)
RGB	Red Green Blue
REQ	Required
SDIF	SGML Document Interchange Format
SGML	Standard Generalized Markup Language
SMPTE	Society of Motion Picture and Television Engineers
SMU	Scaled Measurement Unit
tpi	Tracks Per Inch
tpmm	Tracks Per Millimeter

5 Conventions

The following conventions are used within this Recommendation | International Standard. In this clause, the word parameter(s) is to be interpreted to apply to parameter(s), sub-parameter(s), sub-sub-parameter(s), etc. to any level.

5.1 Names of attributes and parameters

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

- ...expressed by the attribute "subordinates". This attribute...;
- ...two parameters "horizontal dimension"...;
- ...by a sub-parameter "fixed dimension"....

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

5.2 Names of attribute and parameter values

If the values of attributes and parameters are terminal symbols, they are referenced by giving the name of the attribute or parameter value in single quotation marks. For example:

- ...value is 'opaque';
- ...this parameter allows a sender to select a recto or a verso presentation of the page by specifying the value 'recto' or 'verso', respectively.

5.3 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 blocks means layout objects containing blocks, or layout object classes containing blocks;

b) frame component means an object of type frame, or a class to which objects of type frame belong.

The term component description may be qualified in the same way.

5.4 Introduction of terms

Italics are 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 clause 3.

5.5 Representation of degrees

A lower case d is used for the purpose of signifying that the associated value specifies degrees. For example, the notation d0 specifies zero degrees and d180 specifies 180 degrees.

6 General concept of ODA

6.1 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 audio, can coexist within a document;
- the intentions of a document originator with respect to the logical and layout structure of a document can be transmitted to a recipient.

It also supports security aspects such as

- protection against unauthorized semantic knowledge of parts of a document;
- detection of discrepancies between the claimed and actual source and content of parts of the document.

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

- *Interchange* – The process of providing a document to a receiving person or device, by means of data communication or by exchange of storage media.
- *Editing* – The carrying out of operations associated with creation and modification of the structure and/or the content of a document.
- *Formatting* – The carrying out of operations to determine the layout of a document, i.e. the appearance of its content on a presentation medium.
- *Presentation* – 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 formatted 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.

6.2 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 chapters);
- 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).

6.2.1 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 may 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.

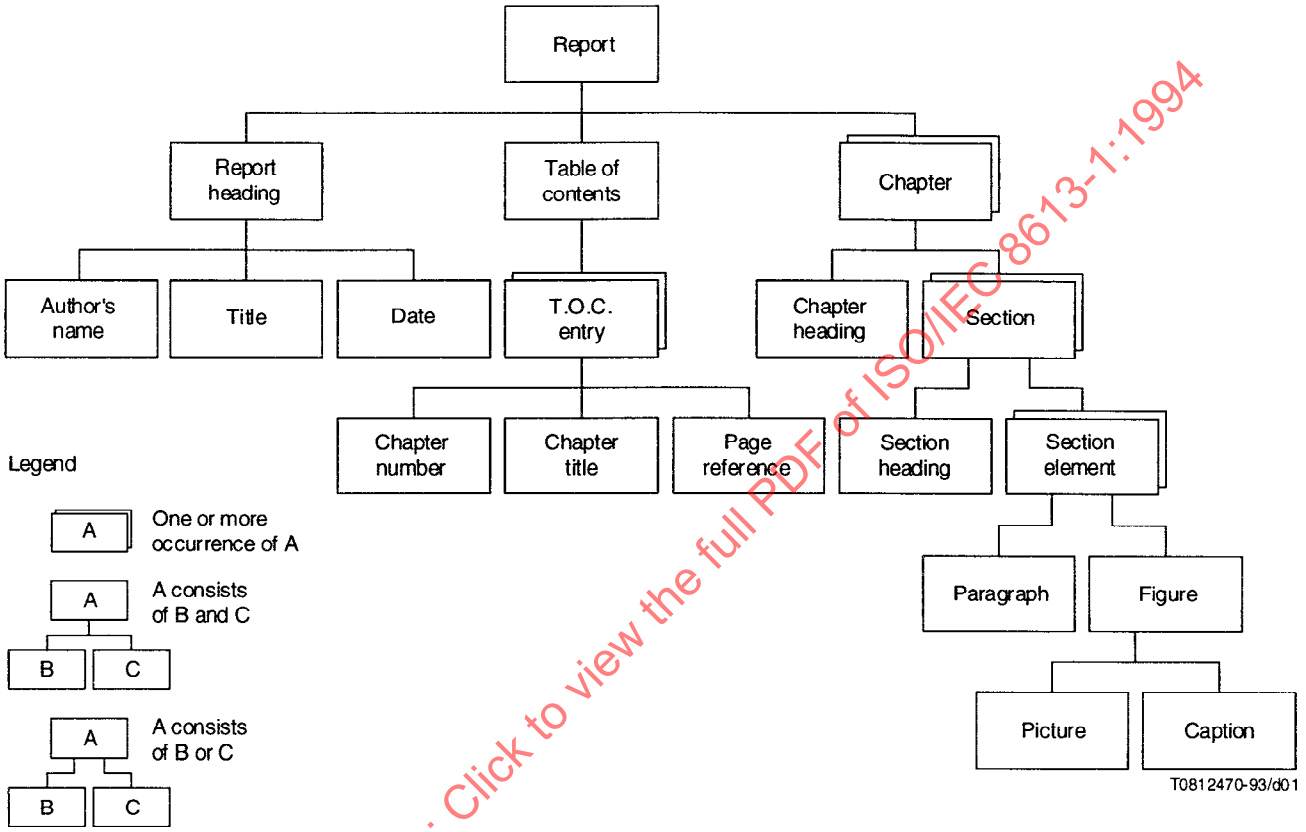

Figure 1 – Example of a logical view of a document

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 on the presentation medium and, if it is a composite object, containing either one or more frames or one or more blocks or, if it is a basic object, containing one or more content portions of the document content.
- *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 by a document application profile using the object class mechanism (see 6.2.6).

6.2.2 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 collected within a constituent type called a content portion.

The relationships between basic logical objects, basic layout objects and content portions are:

- 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 zero or more 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.

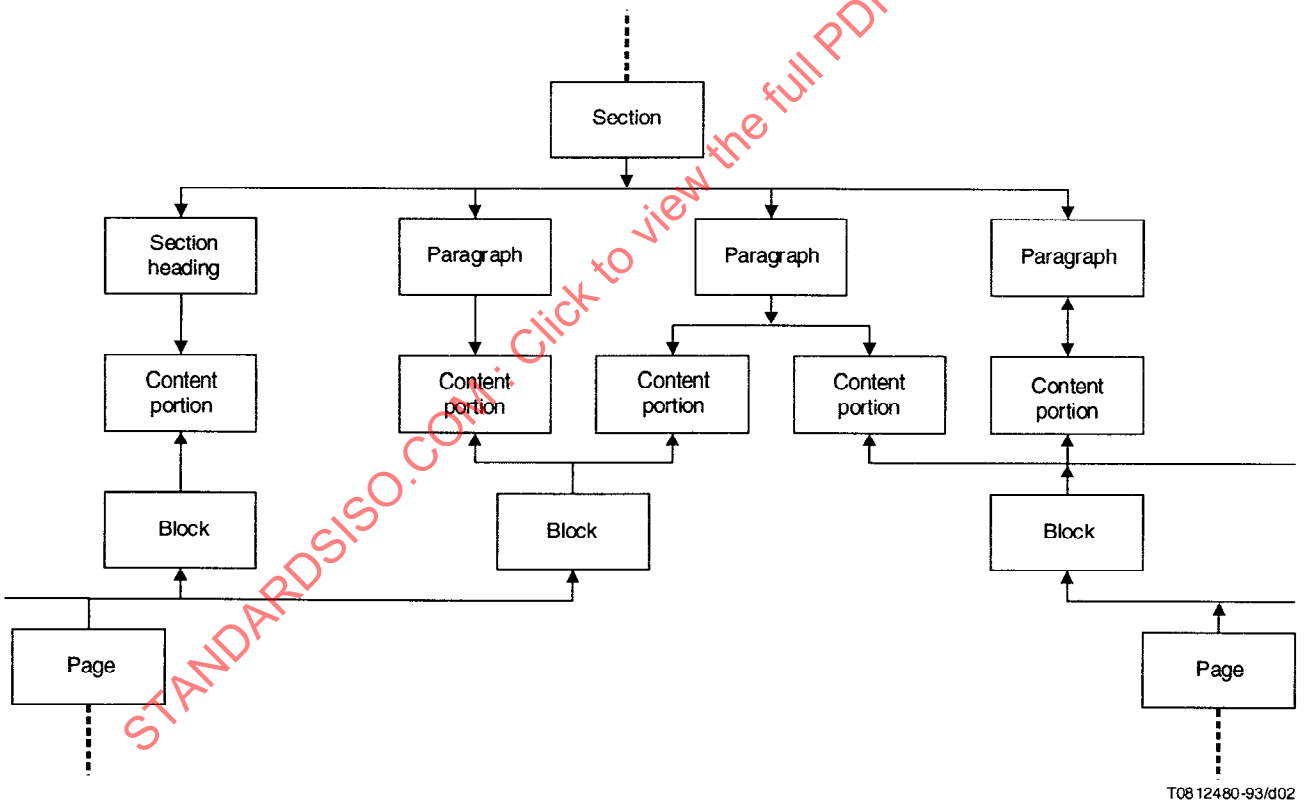

Figure 2 – Possible correspondence between logical and layout objects

6.2.3 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.

Definitions of content architectures are given in other Specifications of the ITU-T Rec. T.410-Series | ISO/IEC 8613. These include:

- ITU-T Rec. T.416 | ISO/IEC 8613-6: Character content architectures;
- ITU-T Rec. T.417 | ISO/IEC 8613-7: Raster graphics content architectures;
- ITU-T Rec. T.418 | ISO/IEC 8613-8: Geometric graphics content architectures.

6.2.4 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, a content portion or a protected part). 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 a 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. In general, 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 "concatenation", the attribute "offset", the attribute "separation" (basic 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 "character 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.

6.2.5 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. In addition to the presentation attributes, 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.

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

6.2.6 Specific 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 can 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 are 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 contains a generic logical structure and/or a generic layout structure. In addition, all document class descriptions shall include a document profile and may include layout styles, presentation styles and generic content portion descriptions.

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 is used as a set of rules from which specific logical objects and structures are derived during the editing process. The generic layout structure is 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 in Figure 3 which shows a page layout with frames for a header, a footer and two columns of body text.

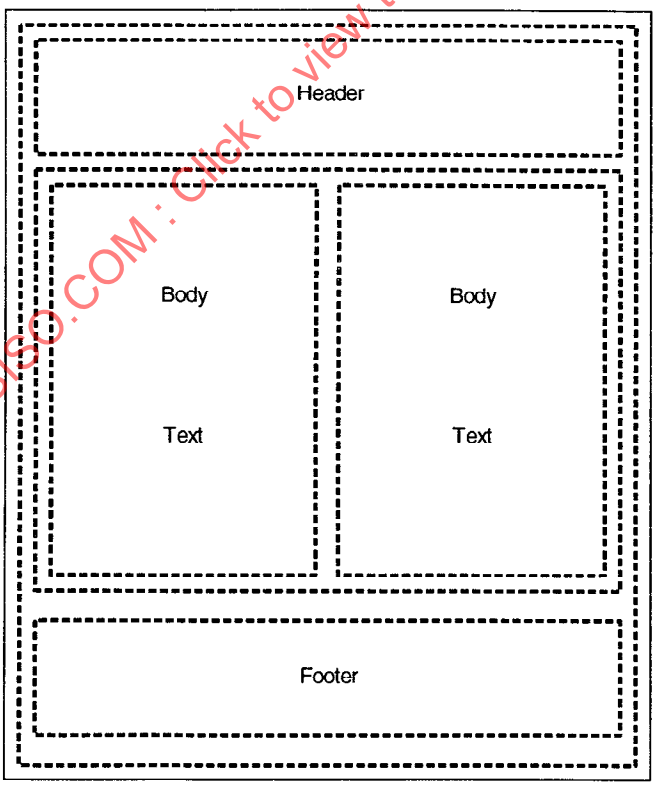

Figure 3 – Example of a page layout

6.2.7 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 some of which are and the remainder of which may be 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.

6.2.8 Generic-document

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

6.2.9 Protected parts

Protected parts enable protection to be provided for parts of a document offering confidentiality, integrity, authenticity and non-repudiation of origin.

Protected parts cannot provide for the protection of a document as a whole. Treatment of the complete document as an object is outside the scope of this series of Recommendations | International Standards.

Parts of a document may be protected before the layout process in their processable form, or after the layout process in formatted or formatted processable forms.

6.3 Document processing model

6.3.1 Relationships of ODA to document processing

The ITU-T Rec. T.410-Series | ISO/IEC 8613 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 ITU-T Rec. T.410-Series | ISO/IEC 8613.

A basic model of document processing is summarized in this subclause (see Figure 4). 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. The order of the processes is not intended to imply that they are performed sequentially in an actual implementation.

6.3.2 Editing process

The document editing process is concerned with creating a new document or modifying an existing one. The document architecture provides data structures for representing the document resulting from this process.

While document creation and modification can 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 may 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.

6.3.3 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.

A formatted processable form document which is processed by an editing process is considered to have reverted to its processable form, because editing changes in the processable form usually result in concomitant changes to the formatted form. The extent to which the original formatted form of the edited document is preserved is regarded as a locally defined agreement between the editing and layout processes and is not defined by the ITU-T Rec. T.410-Series | ISO/IEC 8613.

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

6.3.4 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 paper or on a screen. A document interchanged in accordance with the ITU-T Rec. T.410-Series | ISO/IEC 8613 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 ITU-T Rec. T.410-Series | ISO/IEC 8613 and is regarded as a locally defined process that depends on the presentation device used.

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

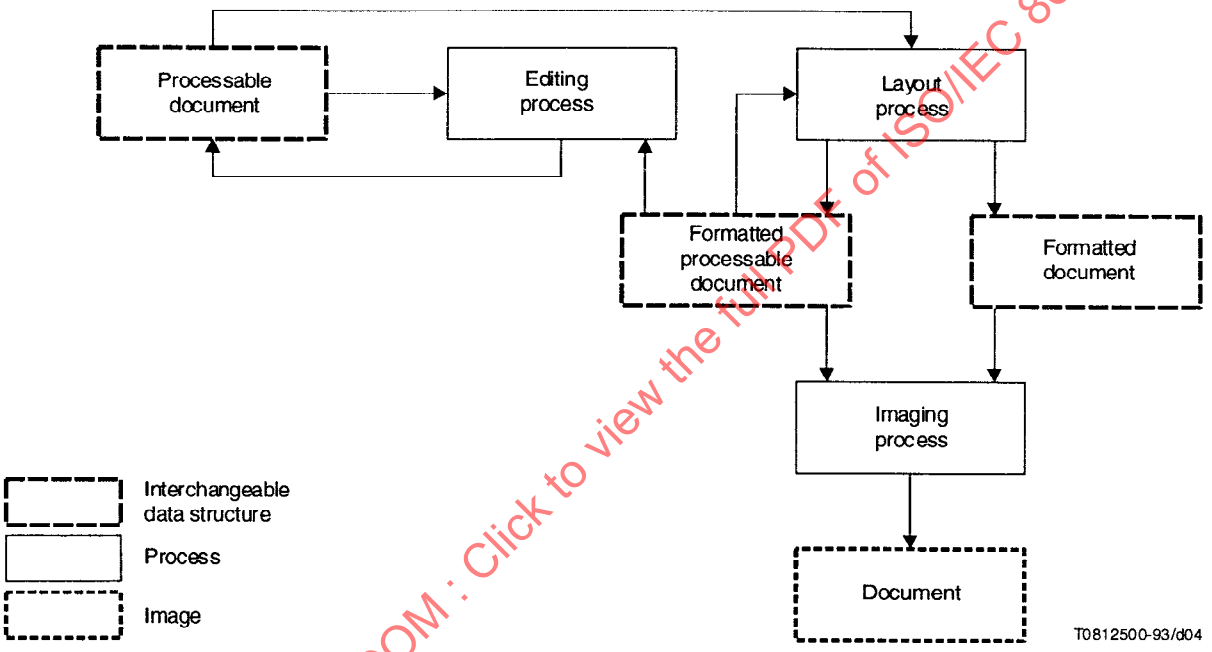

Figure 4 – Basic document processing model

7 Overview of the ITU-T Rec. T.410-Series | ISO/IEC 8613

The Specifications of the ITU-T Rec. T.410-Series are numbered T.411, T.412 and T.414 through T.418. At present there is no T.413. The Parts of ISO/IEC 8613 are numbered 8613-1, 8613-2, 8613-4 through 8613-8 and 8613-10. At present, there is no 8613-3 or 8613-9.

7.1 ITU-T Rec. T.411 | ISO/IEC 8613-1: Introduction and general principles

This Specification provides information about the ITU-T Rec. T.410-Series | ISO/IEC 8613 as a whole by way of an introductory description of the document architecture, an overview of each of the Specifications and a description of their inter-dependencies. References necessary for all Specifications are given, and terms used throughout all Specifications are defined. Conformance to the ITU-T Rec. T.410-Series | ISO/IEC 8613 is specified and rules for defining document application profiles are given.

7.2 ITU-T Rec. T.412 | ISO/IEC 8613-2: Document structures

ITU-T Rec. T.412 | ISO/IEC 8613-2 defines document architecture concepts which may 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 may be produced or interpreted.

ITU-T Rec. T.412 | ISO/IEC 8613-2 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 ITU-T Rec. T.410-Series | ISO/IEC 8613 and a suggested notation for representing them are included.

7.3 ITU-T Rec. T.414 | ISO/IEC 8613-4: Document profile

ITU-T Rec. T.414 | ISO/IEC 8613-4 defines the document profile that provides information concerning the handling of the document. This is accomplished by means of attributes (for example, title, authors), 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 document profile is intended for a recipient (person) and/or device (for example, keywords). Some attribute values can have been automatically generated by a process (for example, size).

7.4 ITU-T Rec. T.415 | ISO/IEC 8613-5: Open document interchange format

ITU-T Rec. T.415 | ISO/IEC 8613-5 defines the format of the data stream used to interchange office documents structured in accordance with ITU-T Rec. T.412 | ISO/IEC 8613-2.

The ODIF data stream is described in terms of a set of data structures, called interchange data elements, which represent the constituents (document profile, object descriptions, object class descriptions, presentation styles, layout styles, content portion descriptions and protected parts) of a document. The formats of the interchanged data elements according to ODIF are defined using the Abstract Syntax Notation One (ASN.1) specified in CCITT Rec. X.208 | ISO/IEC 8824.

ISO/IEC 8613-5 also specifies a clear text language, known as the Open Document Language (ODL), that is used to represent and process documents structured in accordance with ISO/IEC 8613-2. The use of ODL is applicable to ISO/IEC 8613 only.

ODL uses the Standard Generalized Markup Language (SGML) specified in ISO 8879. It consists of a standard set of SGML names and markup conventions for representing the constituents and attributes of a document.

The SGML Document Interchange Format (SDIF) specified in ISO 9069 may be used for interchange of ODA documents represented in ODL between conforming SGML systems.

Both data structure (ODIF) and language (ODL) representations have been standardized in order to meet the document representation requirements of distinct application environments.

7.5 ITU-T Rec. T.416 | ISO/IEC 8613-6: Character content architectures

ITU-T Rec. T.416 | ISO/IEC 8613-6 applies to documents that are structured according to the architecture defined in ITU-T Rec. T.412 | ISO/IEC 8613-2 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.

7.6 ITU-T Rec. T.417 | ISO/IEC 8613-7: Raster graphics content architectures

ITU-T Rec. T.417 | ISO/IEC 8613-7 applies to documents that are structured according to the architecture defined in ITU-T Rec. T.412 | ISO/IEC 8613-2 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, bitmap encoding or other multi-valued encodings.

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.

7.7 ITU-T Rec. T.418 | ISO/IEC 8613-8: Geometric graphics content architectures

ITU-T Rec. T.418 | ISO/IEC 8613-8 applies to documents that are structured according to the architecture defined in ITU-T Rec. T.412 | ISO/IEC 8613-2 that include geometric graphics content, consisting of a descriptive representation of picture description information as an ordered set of drawing elements such as lines, arcs, polygons, attributes for these drawing elements, elements that structure the content portion, and elements that among other things specify the precision of attribute and element values and control the imaging of the picture using the Computer Graphic Metafile (CGM) and its binary encoding defined in ISO/IEC 8632-1 and ISO/IEC 8632-3, 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 terms of its structure, presentation attributes, the relevant CGM parameters, the content layout process, control functions and coding attributes.

7.8 ISO/IEC 8613-10: Formal specifications

The use of ISO/IEC 8613-10 is applicable to ISO/IEC 8613 only. This text provides a formal specification of the concepts, structures, attributes and control functions expressed in natural language in other Parts of ISO/IEC 8613. These include:

- ISO/IEC 8613-2: Document structures;
- ISO/IEC 8613-4: Document profile;
- ISO/IEC 8613-6: Character content architectures;
- ISO/IEC 8613-7: Raster graphics content architectures;
- ISO/IEC 8613-8: Geometric graphics content architectures.

FODA uses a language called Information Modeling by Composition Language (IMCL) to describe ODA document structures, the document profile and the content architectures in terms of abstract information constructs. IMCL is also defined in ISO/IEC 8613-10.

8 Inter-dependencies of the Specifications

If there is a requirement to interchange documents or generic-documents, it is necessary to use the following Specifications together:

- ITU-T Rec. T.411 | ISO/IEC 8613-1: Introduction and general principles;
- ITU-T Rec. T.412 | ISO/IEC 8613-2: Document structures;
- ITU-T Rec. T.414 | ISO/IEC 8613-4: Document profile;
- ITU-T Rec. T.415 | ISO/IEC 8613-5: Open document interchange format.

Additionally, it will be necessary to use one or more of the remaining Specifications, depending on the particular type of content to be interchanged. These include:

- ITU-T Rec. T.416 | ISO/IEC 8613-6: Character content architectures;
- ITU-T Rec. T.417 | ISO/IEC 8613-7: Raster graphics content architectures;
- ITU-T Rec. T.418 | ISO/IEC 8613-8: Geometric graphics content architectures.

If there is a requirement to interchange just the document profile, it is necessary to use the following Specifications together:

- ITU-T Rec. T.411 | ISO/IEC 8613-1: Introduction and general principles;
- ITU-T Rec. T.414 | ISO/IEC 8613-4: Document profile;
- ITU-T Rec. T.415 | ISO/IEC 8613-5: Open document interchange format.

9 Conformance

Conformance to the ITU-T Rec. T.410-Series | ISO/IEC 8613 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:

- no value is specified for the document profile attribute "document application profile";
- the document profile attribute "document application profile" specifies the identifier of a document application profile.

A document application profile shall be specified only if it is identified by an ASN.1 object identifier. This includes document application profiles defined in International Standards or ITU-T Recommendations, or registered by registration authorities; see Annexes B, C and D of CCITT Rec. X.208 | ISO/IEC 8824. ISO/IEC 8824 as published is not in Parts.

In the absence of the specification of a document application profile, the data stream representing the document shall conform to the specifications in this series of Recommendations | International Standards.

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 ITU-T Rec. T.410-Series | ISO/IEC 8613 if it conforms to the requirements above and to the additional constraints specified in the specified document application profile.

10 Document application profile

A document application profile is the specification of a combination of features that are defined in various Specifications of the ITU-T Rec. T.410-Series | ISO/IEC 8613. It is identified by a unique ASN.1 object identifier obtained in accordance with the rules in ISO/IEC 8824.

In order to define a valid combination, the features shall be selected according to the rules given in 10.2.

A document application profile shall include

- one or more document architecture classes;
- one or more content architecture classes;
- the mandatory document profile attributes and optionally one or more permitted document profile attributes;
- an interchange format class.

The document architecture features can be broken down into

a) three classes:
   1) formatted document architecture (FDA);
   2) processable document architecture (PDA);
   3) formatted processable document architecture (FPDA);

b) for each class its
   1) constituents;

c) for each constituent its
   1) attributes;

d) for each attribute its
   1) classification (mandatory, non-mandatory, defaultable);
   2) permissible values divided into basic, non-basic values;
   3) 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: formatted character, processable character, and formatted processable character). The content architecture features are:

e) for each content architecture class its
   1) presentation attributes;
   2) coding attributes;
   3) control functions;

f) for each presentation attribute, coding attribute and control function its
   1) permissible values divided into basic, non-basic values;
   2) default value.

The features of the document profile are its

g) attributes;

h) for each attribute its
   1) classification (mandatory, non-mandatory);
   2) permissible values.

The interchange formats that are permitted by the ITU-T Rec. T.410-Series | ISO/IEC 8613 are

i) the open document interchange format (ODIF) divided into
   1) class A;
   2) class B;

j) and the SGML document interchange format (SDIF) which is applicable to ISO/IEC 8613 only.

Annex F (informative) defines a standardized Proforma and a standardized Proforma notation that may be used for specifying document application profiles in a consistent manner.

10.1 General principles for defining a document application profile

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

A document application profile shall not allow the use of attributes for purposes beyond those defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613. That is, a document application profile shall not modify in any way the semantics of the attributes defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613.

10.2 Rules for defining a document application profile

The rules for defining a document application profile consist of rules for defining document architecture classes, content architecture classes, document profile attributes, and for selecting an interchange format class.

10.2.1 Rules for defining a document architecture class

ITU-T Rec. T.412 | ISO/IEC 8613-2 specifies three document architecture classes. These are formatted document architecture class, processable document architecture class and formatted processable document architecture class.

For each of these classes, ITU-T Rec. T.412 | ISO/IEC 8613-2 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 objects and object classes are classified as mandatory or optional. The document architecture class also defines which attributes are applicable to those objects and object classes and ITU-T Rec. T.412 | ISO/IEC 8613-2 defines all permissible values and a standard default value for each defaultable attribute.

A document application profile defines sets of document architecture class constraints 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 profile.

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

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

a) A set of document architecture class constraints shall pertain to a particular document architecture class, that is, the set of document architecture class constraints shall make use of only those document structures, objects and object classes that pertain to the specified document architecture class.

b) A set of document architecture class constraints shall specify which document structures pertain to that set of document architecture class constraints. Structures pertaining to the corresponding document architecture class that are mandatory shall be specified as mandatory in the set of document architecture class constraints. Structures specified as optional in the document architecture class may be specified as optional or mandatory in the set of document architecture class constraints or they may be disallowed from the document architecture class specification.

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

d) A set of document architecture class constraints shall specify which objects and object classes pertain to that class. Objects and object classes that are mandatory for a particular structure shall be specified as mandatory in the set of document architecture class constraints. Objects and object classes that are specified as optional may be specified as optional or mandatory in the set of document architecture class constraints or they may be disallowed from the document architecture class specification.

e) The document application profile shall specify for each document architecture class any restrictions that are applicable to the document structures. 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 application profile shall specify, in the case of formatted document architecture class, whether the pages are to be composite or basic.

g) The document application profile shall specify all nominal page sizes used in the document. In the case of each nominal page size not defined in ITU-T Rec. T.412 | ISO/IEC 8613-2, the document application profile shall also specify the dimensions of the assured reproduction area, and its position on the nominal page.

   In the case of nominal page sizes defined in ITU-T Rec. T.412 | ISO/IEC 8613-2, the document application profile shall not specify the dimensions of the assured reproduction area, or its position on the nominal page. These shall be the values defined in ITU-T Rec. T.412 | ISO/IEC 8613-2.

h) The document application profile shall specify, in the case of formatted or processable document architecture classes, whether only one content portion or multiple content portions may be associated with basic objects.

i) For each object or object class used, the document application profile shall specify for each document architecture class which attributes are applicable. These shall include the appropriate minimum set of attributes pertaining to each object type as defined in ITU-T Rec. T.412 | ISO/IEC 8613-2.

j) For each permitted attribute, the document application profile shall specify the basic, default and non-basic (if any) values that are applicable. These values shall be taken from the range of permissible values specified in the attribute definitions in ITU-T Rec. T.412 | ISO/IEC 8613-2.

k) The document application profile may classify attributes that are designated as being defaultable or non-mandatory in ITU-T Rec. T.412 | ISO/IEC 8613-2 as being mandatory for each document architecture class. The classification of mandatory attributes shall not be changed.

l) The document application profile shall specify for each document architecture class which attributes may be included in the attribute "default value lists" and shall specify the object types for which a default value list may be specified. ITU-T Rec. T.412 | ISO/IEC 8613-2 gives a definition of the use of this attribute.

10.2.2 Rules for defining a content architecture class

Each specification of the ITU-T Rec. T.410-Series | ISO/IEC 8613 that caters for particular content types defines one or more content architecture classes that corresponds to that content type. The number of content architecture classes defined depends 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, if applicable;
- the types 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. Similarly, the content architecture class definition specifies the permissible values and a recommended default value for the control functions (where applicable).

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

A document application profile defines sets of architecture class constraints and content elements. The sets of architecture class constraints specify the presentation attributes, control functions (if applicable) and coding attributes, and their values, that pertain to the content elements. The document application profile 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 set of content architecture class constraints shall be specified. For example, a document application profile cannot make use of two different sets of content architecture class constraints pertaining to the formatted character content architecture class (CF).

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

a) A set of content architecture class constraints shall pertain to a particular content architecture class, that is, the presentation attributes, content elements, control functions (if applicable), types of coding and coding attributes specified by the content architecture shall 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, processable and formatted processable for character content architecture classes), the sets of content architecture class constraints shall be consistent. For example, the features used in the set of content architecture class constraints of formatted form shall be, when applicable, identical to those used in the set of content architecture class constraints 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 may be specified in a set of content architecture class constraints.

d) The document application profile shall 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 shall be taken from the range of permissible values specified in the corresponding content architecture class.

NOTE 1 – It is recommended that the default values used are those specified in other Specifications of the ITU-T Rec. T.410-Series | ISO/IEC 8613. These include:

- ITU-T Rec. T.416 | ISO/IEC 8613-6: Character content architecture classes;
- ITU-T Rec. T.417 | ISO/IEC 8613-7: Raster graphics content architecture classes;
- ITU-T Rec. T.418 | ISO/IEC 8613-8: Geometric graphics content architecture classes.

e) The document application profile shall specify which set or sets of content elements are applicable. These shall be taken from the permissible sets specified in the corresponding content architecture class. The type or types of coding allowable shall also be specified.

NOTE 2 – 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 document application profile.

When using a document application profile, the ITU-T Rec. T.410-Series | ISO/IEC 8613 allows the interchange of documents containing content architecture classes that are not defined in this Recommendation | International Standard. The ITU-T Rec. T.410-Series | ISO/IEC 8613 does not define how such content architecture classes should be specified except that the interface between the content architecture and the document architecture should be defined as specified in ITU-T Rec. T.412 | ISO/IEC 8613-2. The only restriction imposed on the use of content architecture classes defined outside the ITU-T Rec. T.410-Series | ISO/IEC 8613 is that they are not allowed to be used if no document application profile identifier is indicated in the document profile (see clause 9).

10.2.3 Rules for defining a document profile

ITU-T Rec. T.414 | ISO/IEC 8613-4 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 of a document application profile are given below.

a) The document application profile may specify any document profile attribute defined in ITU-T Rec. T.414 | ISO/IEC 8613-4. It shall not specify attributes not defined in ITU-T Rec. T.414 | ISO/IEC 8613-4.

b) The classification of mandatory attributes shall not be changed. That is, the document application profile shall specify the mandatory document profile attributes as defined in ITU-T Rec. T.414 | ISO/IEC 8613-4.

c) The document application profile may classify attributes that are designated as non-mandatory in ITU-T Rec. T.414 | ISO/IEC 8613-4 as being mandatory for that profile.

d) The document application profile shall specify attribute values taken from the range of permissible values defined in ITU-T Rec. T.414 | ISO/IEC 8613-4.

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

f) The document application profile shall not modify the semantics of the absence of attributes from those semantics specified in ITU-T Rec. T.414 | ISO/IEC 8613-4.

10.2.4 Rules for selecting the interchange format class

ITU-T Rec. T.415 | ISO/IEC 8613-5 defines the valid interchange format classes that may 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 shall be specified concerning the use of an interchange format class in a document application profile.

NOTE – This requirement does not preclude applications in a mixed office and publishing environment, in which documents could be interchanged using class A, class B or SDIF. Such interchange is effected by defining two document application profiles that differ only in their interchange format class; for example, by creating a single specification document with individual ASN.1 object identifiers for class A and SDIF interchange. The use of SDIF is applicable to ISO/IEC 8613 only. Interchange class B is provided only for use with a document application profile for the interchange of Group 4 facsimile documents as documented in CCITT Rec. T.503.

Annex A

References to other standards and registers

(This annex does not form an integral part of this Recommendation | International Standard)

The following Recommendations and International Standards are given for information. They are not required for the application of this series of Recommendations | International Standard:

A.1 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation X.200 (1988), *Reference Model of Open Systems – Open Systems Interconnection – Basic Reference Model.*

  ISO 7498:1984, *Information processing systems – Open Systems Interconnection – Basic Reference Model.*

A.2 Additional references

- ISO 216:1975, *Writing paper and certain classes of printed matter – Trimmed sizes – A and B series.*
- ISO/IEC TR 10183
- *ISO International register of character sets to be used with escape sequences.*
- *ISO International register of graphic character subrepertoires.*
- ECMA 101:1988, *Open 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.*
- CCITT Recommendation T.61:1984, *Character repertoire and coded character sets for the international Teletex service.*
- CCITT Recommendation T.73:1984, *Document interchange protocol for the telematic services.*

  NOTE – CCITT Recommendation T.73 is no longer valid.
- CCITT Recommendation T.400:1988, *Introduction to document architecture, transfer and manipulation.*
- CCITT Recommendation T.503:1988, *A document application profile for the interchange of Group 4 facsimile documents.*

Annex B

Relationships with other standards

(This annex does not form an integral part of this Recommendation | International Standard)

B.1 Transfer standards

Documents structured and represented according to the ITU-T Rec. T.410-Series | ISO/IEC 8613 can be transferred using the functions defined in the following standards and other similar standards.

- ISO 8378-1:1986, *Information processing – Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 7958 ftprad, 3.8 tpmm (96 tpi), on both sides – Part 1: Dimensional, physical and magnetic characteristics.*

- ISO 8571:1988, *Information processing systems – Open System Interconnection – File Transfer, Access and Management.*

- ISO 8630-1:1987, *Information processing – Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 13262 ftprad, on 80 tracks on each side – Part 1: Dimensional, physical and magnetic characteristics.*

- ISO 8860-1:1987, *Information processing – Data interchange on 90 mm (3.5 in) flexible disk cartridges using modified frequency modulation recording at 7958 ftprad, on 80 tracks on each side – Part 1: Dimensional, physical and magnetic characteristics.*

- ISO/IEC 10021:1990, *Information technology – Text Communication – Message Oriented Text Interchange Systems (MOTIS).*

The interchange format defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613 constitutes an application context as defined in:

- ISO 8822:1988, *Information processing systems – Open Systems Interconnection – Connection oriented presentation service definition.*

The abstract syntax of the application context is defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613-5. ISO/IEC 8613-5 also specifies that an interchanged document can be represented by the interchange format defined in:

- ISO 9069:1988, *Information processing – SGML support facilities – SGML Document Interchange Format (SDIF).*

The use of SDIF is applicable to ISO/IEC 8613 only.

B.2 Other standards

For document interchange on magnetic disk the following part of ISO 7487 should be used:

- ISO 7487-1:1985, *Information processing – Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 7958 ftprad, 1.9 tpmm (48 tpi), on both sides – Part 1: Dimensional, physical and magnetic characteristics.*

Annex C

Recording of documents in conformance with ISO/IEC 8613 on interchangeable storage media

(This annex forms an integral part of this International Standard)
This annex is applicable to ISO/IEC only.

C.1 Introduction

This annex specifies the method of recording documents in conformance with ISO/IEC 8613 on volume and file structures of interchangeable storage media in conformance with international standards for volume and file structures, for the purpose of interchanging such documents between users of information processing systems.

Specifically this annex defines how to record documents in conformance with ISO/IEC 8613 when:

- a flexible disk cartridge is employed as an interchangeable storage medium with its volume and file structure in conformance with ISO 9293;
- a CD-ROM is employed as an interchangeable storage medium with its volume and file structure in conformance with ISO 9660.

C.2 Conformance

Recorded document conformance is defined in terms of storage representation conformance, which is specified by the mode of representation of a conforming data stream on an interchangeable storage medium.

A recorded document shall be in conformance with this annex, when the representation of its data stream on a volume and file structure satisfies the requirements specified in C.4.

A prerequisite of such conformance is conformance of the interchangeable storage medium to a standard for volume and file structure. The standard for volume and file structure used in conjunction with this annex is subject to agreement between the originator and the recipient of the medium.

C.3 References

This annex refers to the following documents:

- ISO 9293:1987, *Information processing – Volume and file structure of flexible disk cartridges for information interchange.*
- ISO 9660:1988, *Information processing – Volume and file structure of CD-ROM for information interchange.*

C.4 Procedure for recording documents on interchangeable storage media

C.4.1 Representation and encoding of documents

A document recorded on an interchangeable storage medium shall be in conformance with ISO/IEC 8613. The data stream comprising the recorded document shall be represented using either ODIF or ODL/SDIF as defined in parts 5, 6, 7 and 8 of ISO/IEC 8613 and in accordance with ISO 8825/IEC and ISO 9069.

A document shall be recorded according to an international standard for volume and file structure of the interchangeable storage medium.

- A document shall be recorded in the data space of a file.
- The international standard for volume and file structure may permit a file to be recorded on more than one volume. If a document is recorded over more than one volume, the document shall be recorded over more than one volume as a file in conformance with the standard for volume and file structure.
- The international standard for volume and file structure may permit a file to be recorded on no more than one volume. If a document is recorded over more than one volume, the document shall be recorded in more than one part and each part of the document shall be recorded as a separate file in conformance with the standard for volume and file structure.

  NOTE – If the document comprises more than one file, the method of identifying the sequence of the successive files is not specified by this annex.

C.4.3 Recording of the data stream

The bytes in the data space of a file shall be treated as if they were numbered with consecutive integers assigned in an ascending sequence. The numbering shall start with the first byte of the data space of a file as specified in the relevant standard for volume and file structure.

The successive octets of the data stream comprising an encoded document or part of an encoded document shall be recorded in the consecutive bytes of the data space of a file starting from the first byte of the data space.

If the international standard for volume and file structure for an interchangeable storage medium requires the recording of the file as an organized set of records, the document shall be recorded in the file as a set of records of a record type specified in the international standard for volume and file structure.

C.4.4 Specification for recording on flexible disk cartridges in conformance with ISO 9293

The successive octets of the data stream comprising an encoded document or part of an encoded document shall be recorded in the consecutive bytes of the file space of a file section starting from the first byte of the file space.

NOTES

1 For a file recorded over more than one volume, file section is that part of the file that is recorded on any one volume. The term "file" in ISO 9293 is equivalent to the term "file section" in this clause.

2 If the file comprises more than one file section, the method of identifying the sequence of the successive file sections is not specified by this annex.

3 The Name Extension field of the directory entry that identifies the file or file section is not specified by this annex and is specified by the local system that creates the file or file section. To identify files and file sections that conform to this annex, it is recommended that local systems consider using ODA for the Name Extension field, when the coded graphic character set that applies to the Name Extension field includes these graphic characters.

C.4.5 Specification for recording on CD-ROM in conformance with ISO 9660

The generic specification in C.4.2 and C.4.3 can be applied directly in this case.

NOTE – The File Name Extension field is not specified by this annex and is specified by the local system that creates the file. However, it is recommended that ODA be used as the extension for the file in conformance with this annex when the coded graphic characters permitted in a File Identifier include the Latin alphabet.

Annex D

Principles for the assignment of ASN.1 object identifier values

(This annex forms an integral part of this Recommendation | International Standard)

Values of ASN.1 object identifiers are assigned in various specifications parts of the ITU-T Rec. T.410-Series | ISO/IEC 8613. 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-TSS work "document architecture".

c) The value of the third component is 0, 1, 2, 3 or 4, identifying one of the following categories of object identifier values assigned within this area of work:

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

   2) 1 – object identifier value to be used as a part of an ASN.1 module identifier;

   3) 2 – object identifier value for the identification of a content architecture class;

   4) 3 – object identifier value for the identification of a type of coding;

   5) 4 – object identifier value for the identification of a document application profile.

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:

   1) 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 ITU-T Rec. T.415 | ISO/IEC 8613-5; in this case, there is no fifth component;

   2) if the third component is 1, 2, or 3, the fourth component identifies the ITU-T Rec. T.410-Series | ISO/IEC 8613 Specification in which the value of the fifth component is assigned:

      1) ITU-T Rec. T.411 | ISO/IEC 8613-1;

      2) ITU-T Rec. T.412 | ISO/IEC 8613-2;

      4) ITU-T Rec. T.414 | ISO/IEC 8613-4;

      5) ITU-T Rec. T.415 | ISO/IEC 8613-5;

      6) ITU-T Rec. T.416 | ISO/IEC 8613-6;

      7) ITU-T Rec. T.417 | ISO/IEC 8613-7;

      8) ITU-T Rec. T.418 | ISO/IEC 8613-8.

   3) if the value of the third component is 4, the fourth component identifies a document application profile and the fifth component identifies a version number of this document application profile.

Annex E

Use of MHS to interchange documents conforming to the ITU-T Rec. T.410-Series | ISO/IEC 8613

(This annex forms an integral part of this Recommendation | International Standard)

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	**{ See Note 2}**
... ...	{ . . . . }
... ...	{ . . . . }
... ...	{ . . . . }

NOTES

1 When using [MHS/MOTIS] to transfer documents conforming to ODA, the MTS may perform format conversion. Format conversion of ODA documents may result in loss of information. If format conversion is not appropriate, this should be indicated by the sender when submitting a message with ODA body parts to [MHS/MOTIS].

2 These document application profiles ASN.1 object identifiers are those defined for TSS | ISO/IEC. Other organizations shall use object identifiers as appropriate.

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 IPMSInformationObjects  { joint-iso-ccitt(2) mhs-motis(6) ipms(1) modules(0)
        information-
        objects(2) };

oda-body-part  EXTENDED-BODY-PART-TYPE
  PARAMETERS        OdaBodyPartParameters IDENTIFIED BY id-et-oda-param
  DATA                OdaData
  ::= id-et-oda-data
        -- Abstract syntax for ODA body part parameters shall appear in the parameter elements 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-data object identifier --

  document-architecture-class    [1]      INTEGER  {
                                          formatted  (0),
                                          processable  (1),
                                          formatted processable (2) } }
        -- Abstract syntax for ODA data shall appear in the data element of an IPM ExternallyDefinedBodyPart --

OdaData                      ::=  SEQUENCE OF Interchange-Data-Element

  id-et-oda-param OBJECT IDENTIFIER    ::= { 2 8 1 1 2 },
        -- identifies the Abstract Syntax for ODA bodypart parameters using the ASN.1 basic encoding rules --
  id-et-oda-data OBJECT IDENTIFIER     ::= { 2 8 1 1 1 }
        -- identifies the Abstract Syntax for ODA data using the ASN.1 basic encoding rules --

END     -- of IPMSExtendedBodyPartTypeOda --
```

Annex F

Document Application Profile Proforma and Notation

(This annex does not form an integral part of this Recommendation | International Standard)

F.1 Introduction

This annex defines a standardized Proforma for the specification of document application profiles. It also defines a recommended notation for describing the technical specifications of document application profiles.

The use of both Proforma and notation is intended to

- aid clarity and remove ambiguity in the definition of document application profiles;
- ensure that all necessary information is included in the document application profile and is correctly specified;
- aid comparison between document application profiles;
- aid verification that a particular document application profile conforms to the rules defined in clause 10 for the specification of document application profiles.

F.2 Proforma

This subclause contains two parts: a document application profile proforma outline in F.2.1 and a description of the material that should be contained in the clauses of any particular document application profile in F.2.2.

F.2.1 Document application profile proforma outline

0 Introduction

1 Scope and field of application

2 References

3 Definitions and abbreviations

4 Relationships to other document application profiles

5 Conformance

6 Characteristics supported by this document application profile

- 6.1 Overview
- 6.2 Logical constituents
- 6.3 Layout constituents
- 6.4 Document layout control
- 6.5 Content layout and imaging control
- 6.6 Miscellaneous features
- 6.7 Document management features

7 Specification of constituent constraints

- 7.1 Document profile constraints
- 7.2 Logical constituent constraints
- 7.3 Layout constituent constraints
- 7.4 Layout style constraints
- 7.5 Presentation style constraints
- 7.6 Content portion constraints
- 7.7 Additional usage constraints

8 Interchange Format

F.2.2 Document application profile clause description

Where possible, a document application profile should follow the terminology defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613 and the naming convention used in F.3.6 (Table F.1) and F.3.7 (Table F.2).

F.2.2.1 Clauses 0-3

Clauses 0 to 3 are intended to be supplied in accordance with TSS | ISO/IEC Rules for the drafting and presentation of Recommendations | International Standards.

F.2.2.2 Clause 4

Clause 4 describes how the document application profile relates to other document application profiles; for example in its hierarchical relationship with other document application profiles.

F.2.2.3 Clause 5

Clause 5 details conformance requirements, specified in terms of data stream representations of documents conforming to the document application profile. This clause should also contain further information concerning the use of the document application profile in applications.

F.2.2.4 Clause 6

The purpose of clause 6 is to specify, in natural language form, the constituent constraints that are formally defined in clause 7 of a document application profile and that may be used to represent documents in accordance with the document application profile. The specification in clause 6 should be complete. That is, it should specify all the characteristics of each constituent constraint including any usage constraints that are applicable. Such usage constraints shall be adhered to when representing documents in accordance with the document application profile.

In addition, clause 6 should contain a description of the typical use of the constraints to provide guidance as to how a data stream conforming to the document application profile is to be interpreted. This should be described in terms of a set of abstract features that are supported by the document application profile.

The subclauses given for clause 6 in F.2.1 are for guidance only since the requirements may differ depending on the document application profile. Following an overview, the logical and layout constituents are described in 6.2 and 6.3 in a document application profile respectively. The content independent layout features that are supported by the document application profile are described in 6.4. This describes the controls available for positioning content within pages and frames, for example, offsets between the content and frame boundaries, separation between blocks of content and page breaks.

The layout and presentation features that pertain to each content type that is supported by the document application profile are described in 6.5. Miscellaneous features supported by the document application profile that are not covered elsewhere, such as automatic page or paragraph numbering, are described in 6.6. Document management features that are supported are described in 6.7.

F.2.2.5 Clause 7

This clause contains a definition of the constituents that should be used to represent a document that conforms to the document application profile. These definitions shall be in accordance with the notation defined in subclause F.3. In clause 7, 7.1 should contain the document profile constraints, 7.2 the logical constituent constraints, 7.3 the layout constituent constraints, 7.4 the layout style constraints, 7.5 the presentation style constraints and 7.6 the content portion constraints. All of the additional usage constraints not covered in 7.1 to 7.6 should be contained in 7.7. The constraints contained in 7.7 should be specified in the appropriate natural language and/or using the same formal specification techniques as are used in the Formal Specification of the ITU-T Rec. T.410-Series | ISO/IEC 8613. Such constraints could also be included as comments in affected constituent constraints.

It is recommended that 7.1 to 7.6 in clause 7 each be divided into three further parts:

7.x.1 Macro definitions

7.x.2 Factor constraints

7.x.3 Constituent constraints

7.x.1 and 7.x.2 contain, respectively, any macro definitions (see F.3.3.1) and factor constraints (see F.3.3.2) that pertain to the constituent constraints defined in 7.x.3.

F.2.2.6 Clause 8

This clause specifies the interchange classes selected for the document application profile. It is possible to specify, in a document application profile description based on the ITU-T Rec. T.410-Series, that ODIF is used for document interchange. It is possible to specify, in a document application profile description based on ISO/IEC 8613, that ODIF or ODL/SDIF (or both) are used for the interchange of documents. The use of ODL/SDIF is applicable to ISO/IEC 8613 only.

F.3 Proforma and notation

This subclause provides production rules for the specification of the constituent constraints allowed in a document application profile.

F.3.1 gives an overview of how clause 7 of the document application profile is formally defined according to the notation and production rules specified in this annex.

The production rules are defined according to a context free grammar using a Backus-Naur-Form (BNF) which makes use of the symbols defined in F.3.2.1. The terminal symbols used in the notation are defined in F.3.2.2.

F.3.3 contains the description and format of the production rules used for the specification of the set of constituent constraint definitions which constitutes clause 7 of a document application profile. This includes a macro facility described in F.3.3.1. For convenience the production rules are listed in full in F.3.4. F.3.5 provides an example of the recommended style of layout for document application profile specifications.

F.3.6 contains a table of the ITU-T Rec. T.410-Series | ISO/IEC 8613 attribute and (sub-)parameter names and their associated Document Application Profile Proforma Notation data types. This table provides a set of unique names that will facilitate the construction of clause 7 of a document application profile that is consistent with clause F.3. F.3.7 provides a similar table for the character content architecture control functions, that is, a list of abbreviated control function names and their associated Document Application Profile Proforma Notation data types.

F.3.1 Overview

A document application profile defined in accordance with this annex consists of a set of constituent constraint definitions. The section numbers and headers in clause 7 of a Document Application Profile Proforma do not form part of the notation.

A constituent constraint definition consists of a number of attribute constraint definitions for attributes that are either specified explicitly for the constituent or are applicable to the constituent (for example, layout directives in the case of object descriptions). Thus a document application profile specification consists of a set of constituent constraint definitions relating to the document profile, object and object class descriptions, layout styles, presentation styles and content portion descriptions.

A constituent constraint definition specifies which attributes shall be specified in instances of that constituent, which attributes should be optionally specified and which attributes are disallowed.

Also, a constituent constraint definition specifies which values should be unconditionally specified for each attribute as it specifies the basic values that are applicable. The non-basic values that are allowed (if any) and any default values which are other than the recommended values in the ITU-T Rec. T.410-Series | ISO/IEC 8613 are specified in the constituent constraint definition relating to the document profile.

When defining a document application profile using this Specification Proforma, it is necessary to take into account

a) The rules for defining document application profiles as specified in clause 10 of this Specification. In particular, attributes specified as mandatory in the ITU-T Rec. T.410-Series | ISO/IEC 8613 shall be specified as required in the corresponding constituent constraint definitions in the document application profile. Attributes classified as defaultable or non-mandatory in the ITU-T Rec. T.410-Series | ISO/IEC 8613 should be specified as permitted or required in the corresponding document application profile constituent constraint definitions.

b) A document application profile specification may disallow the occurrence of a defaultable attribute in a constituent. An application shall take into account all defaultable attributes that are applicable to each constituent in the document, irrespective of whether the document application profile allows the occurrence of those attributes or not.

c) A document application profile specification does not repeat requirements already specified in the ITU-T Rec. T.410-Series | ISO/IEC 8613 although these requirements automatically apply to the documents conforming to the document application profile. As a consequence, a document application profile specification does not provide information relating to the internal consistency of a document and does not include any additional information that will ensure the internal consistency of a document.

F.3.2 Description of the notation

F.3.2.1 Symbols used in the meta-language

The symbols listed and defined below are used to define the production rules contained in F.3.3 which specify the format of a document application profile specification.

::= is used to specify that the string of symbols on the right side is to be substituted for the non-terminal symbol on the left side.

| is used to separate alternatives.

< > is used as a pair of symbols to delimit a non-terminal symbol.

-- is used as a paired pair of symbols to delimit a comment string.

NOTE – A comment string is used for describing symbols (non-terminals) which are not formally defined by further production rules but are defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613; e.g. **{ attribute-name }** *-- a name that identifies an attribute from the ITU-T Rec. T.410-Series | ISO/IEC 8613, see F.3.6 (Table F.1) --*

{ } is used as a pair of symbols to delimit a syntactical unit.

[ ] is used as a pair of symbols to delimit an optional syntactical unit, i.e. which may be present or absent.

... is used following a syntactical unit to indicate that the syntactical unit may be repeated, e.g. **[<parameter>]...**

' ' is used as a pair of symbols to delimit a terminal symbol.

A syntactical unit is either a non-terminal symbol or a sequence of symbols enclosed in either curly brackets **{ }** or square brackets **[ ]**.

These symbols are also used in document application profile defined production rules used for the definition of allowed values for attributes such as "bindings" and "generator for subordinates". However, an exception to this is that the apostrophe symbols used to delimit a terminal symbol are not used in document application profile defined production rules.

F.3.2.2 Terminal symbols used in the notation (Reserved Names)

The following is a list of the terminal symbols used in the notation defined in F.3.3. It does not include those terminal symbols in ITU-T Rec. T.412 | ISO/IEC 8613-2 which are used in some of the constraint specifications.

GENERIC	announces attributes specified for object classes.
SPECIFIC	announces attributes specified for objects.
SPECIFIC_AND_GENERIC	announces attributes specified for object classes, objects or both.
REQ	announces attributes or parameters that shall be specified for the associated constituent or attribute.
PERM	announces defaultable and non-mandatory attributes or parameters that may be specified for the associated constituent or attribute. If the attribute/parameter is defaultable, then the attribute/parameter shall have a value within the specified value range, either explicitly specified in the constituent/attribute or derived using the defaulting rules specified in ITU-T Rec. T.412 | ISO/IEC 8613-2. If the attribute/parameter is non-mandatory, then the constraint on the value is applicable only if the attribute/parameter is specified in the constituent/attribute.
DIS	announces defaultable attributes or parameters that shall not be specified for the associated constituent or attribute. The value of that attribute/parameter obtained by using the defaulting rules specified in ITU-T Rec. T.412 | ISO/IEC 8613-2 shall be within the specified value range.

MUL	announces a repetitive group of attributes or parameters.
PMUL	announces an optionally repetitive group of attributes or parameters.
FACTOR	announces a common set of constraints.
DEFINE	announces a string substitution macro.
CASE	announces a conditional selection mechanism that depends on another attribute value.
OF	announces a list of possible attribute values in the CASE conditional selection mechanism.
ANY_VALUE	is used to denote that any attribute or (sub-)parameter value may be specified that is permitted in the ITU-T Rec. T.410-Series | ISO/IEC 8613.
OBJECT_CLASS_ID_OF	is used to specify any object class identifier from the set of instances of a particular constituent constraint.
OBJECT_ID_OF	is used to specify any object identifier from the set of instances of a particular constituent constraint.
STYLE_ID_OF	is used to specify any style identifier from the set of instances of a particular style constraint.
SUB_ID_OF	is used to specify, implicitly, a subordinate object identifier from the set of instances of a particular constituent constraint.
CONTENT_ID_OF	is used to specify, implicitly, a content portion identifier from the set of instances of a particular constituent constraint.
VIRTUAL	is used to specify that the attribute value is specified in another constraint.
NOVR	is used to specify that an attribute or parameter value derived from an object class shall not be overridden in corresponding objects.
SUPERIOR	is used when referencing an attribute, in a conditional selection mechanism, to specify the first superior constituent constraint containing the attribute.
VOID	is used when referencing an attribute to specify the case when the attribute is absent.
ANY_STRING	is used to specify any character string allowed by the ITU-T Rec. T.410-Series | ISO/IEC 8613 consistent with the document application profile.
ANY_INTEGER	is used to specify any integer value allowed by the ITU-T Rec. T.410-Series | ISO/IEC 8613.
ASN.1	is used to announce an ASN.1 object identifier value.
EXCEPT	is used to specify a range of values that shall not be used.
CHARACTER	is used to specify any graphic character allowed by the document application profile.
RASTER	is used to specify any raster content allowed by the document application profile.
GEOMETRIC	is used to specify any geometric content allowed by the document application profile.
ESC	is used to specify an escape sequence allowed by the document application profile.
+	is used to indicate one or more occurrences of an attribute value.
{ }	is used as a pair of symbols to delimit a syntactical unit.
[ ]	is used as a pair of symbols to delimit an optional syntactical unit.
( )	is used as a pair of symbols to delimit parameters in functions and macros.
< >	is used as a pair of symbols to delimit a non-terminal symbol.
-- --	is used as a paired pair of symbols to delimit a comment in a document application profile. Comments shall not be nested.

;	is used to terminate a simple attribute specification.
|	is used to separate alternatives.
:	is used to separate attribute or (sub-)parameter values in a case specification from the associated attribute or (sub-)parameter specification.
,	is used to separate elements of a set.
..	is used to specify a range of integers or characters.
#	is used to announce a parameter or a control function name.
" "	is used as a pair of symbols to delimit a character string. If a double quotation mark is to be used within the character string, then it shall be announced with another double quotation mark.
'space'	is used to separate elements of a sequence and terminal symbols in general.
...	is used to indicate that an element may be repeated.
$	is used to announce a macro substitution.
=	is used to specify equality.
>	is used to specify greater than.
<	is used to specify less than.
>=	is used to specify greater than or equal to.
<=	is used to specify less than or equal to.
/	is used to separate character code table column and row positions in an escape sequence constraint specification.
\	is used, within character strings, to delimit non-printable strings.

In addition to the above a space is used as a terminal symbol to indicate the presence of one or more spaces separating integer values in the specification <asn.1-object-id>.

F.3.3 The document application profile specification

A document application profile specification consists of intermixed macro definitions and constraint definitions. The format of the macro definitions and the method of substitution is described in F.3.3.1. F.3.3.2 defines the format of the constraint definitions and F.3.3.3 defines the format of the individual attribute specifications which make up a constraint definition.

Thus the format of the document application profile specification is as follows:

<document-application-profile> := { <macro-definition> | <constraint-definition> }...

F.3.3.1 Format of macro definition

The macro facility provides a shorthand mechanism for the notation. For each macro definition of the form DEFINE (S, "..."), corresponding text or additional macros are substituted for all strings of the form: $S where S is a unique macro name identifying the macro. The string substitution macros shall be fully expanded to a defined single string of characters lexically inserted into the corresponding part of the notation.

The format of a macro definition is as follows :

<macro-definition> **::= 'DEFINE' '(' <macro-name> ',' <macro-string> ')'**

<macro-name> -- *A unique name that identifies a document application profile macro name. It consists of a sequence of one or more characters. The first character shall be a letter ('a'..'z', 'A'..'Z'). The remaining characters shall be letters ('a'..'z', 'A'..'Z'), digits ('0'..'9') or hyphens ('-'). Uppercase and lowercase letters are not significant but their use throughout a document application profile should be consistent.* --

<macro-string> **::= <printable-string>**

Macros shall be defined before their macro names are used. Macros may be repeated for readability but should not be redefined.

EXAMPLE 1 (in recommended style)

$S is expanded to the string "abc" with the macro definitions:

```
DEFINE (S, "$T")

DEFINE (T, "abc")
```

EXAMPLE 2 (in recommended style)

With the following two macro definitions:

```
DEFINE (CHAPTER, "
        <chapter>       ::= SEQ (<title> <body>)
                            | <body>
                            ;
        <title>         ::= " "Chapter1" "
                            ;
$BODY
")

DEFINE (BODY, "
        <body>          ::= SEQ (<section> ... <ending>)
                            ;
")
```

$CHAPTER would be expanded to :

```
<chapter>               ::= SEQ (<title> <body>)
                            | <body>
                            ;
<title>                 ::= "Chapter1"
                            ;
<body>                  ::= SEQ (<section> ... <ending>)
                            ;
```

F.3.3.2 Format of constraint definitions

A constraint definition consists of a document application profile constituent constraint or a factor constraint.

A document application profile constituent constraint definition specifies the constraints on the presence and values of attributes that are applicable to that particular constraint.

A factor constraint definition specifies constraint information that is common to one or more of the document application profile constituent constraints.

The format of the constraint definition is as follows:

```
<constraint-definition>         ::= <constraint-name> '{' <constraint> '}'
```

F.3.3.2.1 Constituent and factor names

Document application profile constituent names and factor names are used to identify document application profile constituent constraints and factor constraints respectively within a document application profile specification.

The format of a constraint name is as follows:

```
<constraint-name>               ::= <constituent-name> | <factor-name>
```

A document application profile constituent name consists of one or two fields separated by the colon character ":". The first field applies to the constituent itself. The second field (if used) identifies factor constituent constraints that specify additional constraints imposed on that constituent.

A factor name is structured similarly to a document application profile constituent name. The first field in the name is used to identify the factor constituent constraint. The second field (if used) identifies another factor constituent constraint which contains additional constraints that are applicable to that factor constituent constraint.

<constituent-name> ::= **<c-name> [<factor-reference>]**

<c-name> -- *A unique name that identifies a document application profile constituent constraint. It consists of a sequence of one or more characters. The first character shall be an uppercase letter ('A'..'Z'). The remaining characters shall be uppercase letters ('A'..'Z'), lowercase letters ('a'..'z'), digits ('0'..'9') or hyphens ('-'). --*

<factor-reference> ::= **':' <f-name>**

<factor-name> ::= **'FACTOR' <f-name> [<factor-reference>]**

<f-name> -- *A unique name that identifies a document application profile factored constraint. It consists of a sequence of one or more characters. The first character shall be an uppercase letter ('A'..'Z'). The remaining characters shall be uppercase letters ('A'..'Z'), digits ('0'..'9') or hyphens ('-'). --*

F.3.3.2.2 Constraint specifications

If a constraint specification relates to an object or object class, the constraint specification is divided into sections (up to three sections). These are the generic part, the specific part and the specific_and_generic part. They respectively define the constraints that are applicable to object classes, to objects, and to both objects and object classes for which the constraint specification applies.

If the constraint specification relates to some other constituent of the document, for example, the document profile, a style or a content portion, then the constraint specification consists of an attribute expression.

Each section consists of an attribute expression specification.

The format of the constraint specification is as follows:

<constraint> ::= **<section>... | <attribute-expr-a>**

<section> ::= **{ 'GENERIC' | 'SPECIFIC' | 'SPECIFIC_AND_GENERIC' } ':'**
<attribute-expr-a>

F.3.3.3 Attribute expression specifications

An attribute expression specification is used to separate a list of attributes and to allow a choice between groups of attributes.

<attribute-expr-a> ::= **<attribute-expr-b> ['|' <attribute-expr-b>]...**

<attribute-expr-b> ::= **<attribute-expr-c> [',' <attribute-expr-c>]...**

An attribute expression type c may take one of two forms :

- case attribute value range specification;
- simple attribute value range specification.

The format of an attribute expression type c specification is as follows:

<attribute-expr-c> ::= **<case-attribute> | <simple-attribute>**

F.3.3.3.1 Case attribute value range specifications

This form is used when the attribute specification within a constraint definition depends upon the value of another attribute (or parameter) in the same or another constraint definition.

This specification begins with the terminal symbol CASE. The specification then consists of the field <reference-type> which may be either a functional reference or a direct reference. The terminal symbol SUPERIOR is used to specify the first superior constituent constraint containing the attribute. The <attribute-reference> non-terminal is a reference to an attribute (or parameter) in the same or another constituent definition.

The attribute or parameter reference and terminal symbol OF, is followed by a list of case specifications. Each case specification consists of an attribute (or parameter) value pertaining to the attribute indicated in the field <attribute-reference> and an associated attribute specification which corresponds to that value. The terminal symbol VOID is used to specify the case where the attribute is not present. Each attribute value specification is in the format defined in F.3.3.3.

The format is as follows:

```
<case-attribute>              ::= 'CASE' <reference-type> 'OF'
                                  '{' <single-case-attribute>... '}'
<reference-type>              ::= <functional-reference> | <direct-reference>
<functional-reference>        ::= <function> '(' <direct-reference> ')'
<function>                    ::= 'SUPERIOR'
<direct-reference>            ::= <c-name> '(' <attribute-reference> ')'
<attribute-reference>         ::= <attribute-name> [ '#' <parameter-name> ]...
<single-case-attribute>       ::= <case-value> ':' <attribute-expr-a>
<case-value>                  ::= <attribute-value-except> | 'VOID'
```

EXAMPLE 3 (in recommended style)

```
DEFINE (FDA,          " 'formatted' ")
DEFINE (FPDA          " 'formatted-processable' ")
DEFINE (FC            "ASN.1 { 2 8 2 6 0 }")
CASE Document-profile (Document-architecture-class) OF {
        { $FDA }:
          REQ         Presentation-attributes {
            REQ       #content-architecture-class { $FC }
                                    }
        { $FPDA }:
          REQ         Presentation-attributes {
            REQ       #content-architecture-class { ANY_VALUE }
                                  }
                                              }
```

In this example, the value of the parameter "content architecture class" for the attribute "Presentation attributes" depends on the document architecture class to which the document belongs.

F.3.3.3.2 Simple attribute value range specifications

A simple attribute value range specification begins with one of the following keywords:

REQ (required) – This means that the attribute shall always be specified for the constituent, i.e. the attribute is mandatory.

PERM (permitted) – This means that a defaultable or non-mandatory attribute may but need not be specified for the constituent. If the attribute is defaultable, then the attribute shall have a value within the specified value range, either explicitly specified for the constituent or derived using the defaulting rules specified in ITU-T Rec. T.412 | ISO/IEC 8613-2. If the attribute is non-mandatory, then the constraint on the value is applicable only if the attribute is specified for the constituent.

DIS (disallowed) – This means that a defaultable attribute shall not be specified for the associated constituent. The value of the attribute derived using the defaulting rules specified in ITU-T Rec. T.412 | ISO/IEC 8613-2 shall be within the specified value range.

If an attribute is not explicitly specified and the attribute is defaultable, then the attribute value is derived using the defaulting rules specified in ITU-T Rec. T.412 | ISO/IEC 8613-2.

The keyword REQ, PERM or DIS is followed by the attribute name and the attribute specification. The keyword EXCEPT maybe inserted between attribute value specifications to restrict a range of values. Each attribute value specification may consist of a parameter expression, a keyword such as ANY_VALUE, content information or an attribute value expression optionally preceded by the keyword NOVR. The keyword NOVR indicates that the attribute derived from an object class shall not be overridden in the corresponding object. An attribute value specification may also be empty to allow for the empty parameter list.

The format of the simple attribute specification is as follows:

```
<simple-attribute>          ::= { 'REQ' | 'PERM' | 'DIS' } <attribute-name>
                                <attribute-value-except>

<attribute-name>            -- a name that identifies an attribute from the ITU-T Rec. T.410-Series |
                               ISO/IEC 8613, see also Table F.1 in F.3.6 --

<attribute-value-except>    ::= '{' <attribute-value> '}'
                                [ 'EXCEPT' '{' <attribute-value> '}' ]

<attribute-value>           ::= -- empty --
                                | <parameter-expr-a>
                                | <keyword>
                                | <content-information>
                                | [ 'NOVR' ] <attribute-value-expr>
```

Attribute names consist of a sequence of one or more characters. The first character shall be an uppercase letter ('A'..'Z'). All the other characters shall be lowercase letters ('a'..'z'), digits ('0'..'9') or hyphens ('-').

These names are as specified in the ITU-T Rec. T.410-Series | ISO/IEC 8613 with the exception that where the attribute name consists of more than one word, the spaces between the words are replaced by hyphens. F.3.6 (Table F.1) lists all the attribute names and their associated data types.

F.3.3.4 Parameter expression specifications

A parameter (or sub-parameter) expression specification allows choices between groups of parameters and allows repetition of groups of parameters. The comma symbol "," is used to separate parameters and the vertical bar symbol "|" is used to separate groups of parameters. The keywords MUL and PMUL are used to indicate repetition and optional repetition of a group of parameters. This is necessary for some applications of the definition of the attribute "Bindings" in document application profiles and other cases which allow sets of parameters. Each parameter name in a parameter expression shall be preceded by either the REQ, PERM or DIS keyword.

REQ (required) – This means that the parameter shall always be specified for the constituent, i.e. the parameter is mandatory.

PERM (permitted) – This means that a defaultable or non-mandatory parameter may but need not be specified for an attribute. If the parameter is defaultable, then the parameter shall have a value within the specified value range, either explicitly specified for the parameter or derived using the defaulting rules specified in ITU-T Rec. T.412 | ISO/IEC 8613-2. If the parameter is non-mandatory, then the constraint on the value is applicable only if the parameter is specified for the attribute.

DIS (disallowed) – This means that a defaultable parameter shall not be specified for the associated attribute. The value of the parameter derived using the defaulting rules specified in ITU-T Rec. T.412 | ISO/IEC 8613-2 shall be within the specified value range.

The format of the parameter expression specification is as follows:

```
<parameter-expr-a>          ::= <parameter-expr-b> [ '|' <parameter-expr-b> ]...

<parameter-expr-b>          ::= <parameter-expr-c>
                                | <parameter-expr-c> <parameter-expr-c>...
                                | <parameter-expr-c> { ',' <parameter-expr-c> }...

<parameter-expr-c>          ::= <case-parameter>
                                <simple-parameter>
                                | 'REQ' '{' <parameter-expr-a> '}'
                                | 'PERM' '{' <parameter-expr-a> '}'
                                | 'MUL' '{' <parameter-expr-a> '}'
                                | 'PMUL' '{' <parameter-expr-a> '}'
```

A parameter expression type c may be a case parameter or a simple parameter for single parameter specifications. The case form of specification is used when the parameter specification depends on the value of another attribute (or parameter) in the same or another constraint definition. (See F.3.3.3.1 for a description of the case mechanism.)

F.3.3.4.1 Case parameter value range specifications

The format of the case parameter specification is as follows:

```
<case-parameter>            ::= 'CASE' <reference-type> 'OF' '{' <single-case-parameter> '}'...

<single-case-parameter>     ::= <case-value> ':' <attribute-value-except>
```

F.3.3.4.2 Simple parameter value range specifications

In the case of a simple parameter value specification the format is as follows:

```
<simple-parameter>          ::= { 'REQ' | 'PERM' | 'DIS' } '#' <parameter-name>
                                <attribute-value-except>

<parameter-name>            -- a name that identifies a (sub-)parameter from the ITU-T Rec. T.410-Series |
                               ISO/IEC 8613, see F.3.6 (Table F.1) --
```

(Sub-)parameter names consist of a sequence of one or more characters. The first character shall be a lowercase letter ('a'..'z'). All the other characters shall be lowercase letters ('a'..'z'), digits ('0'..'9') or hyphens ('-').

These names are as specified in the ITU-T Rec. T.410-Series | ISO/IEC 8613 with the exception that where the parameter name consists of more than one word, the spaces between the words are replaced by hyphens. F.3.6 (Table F.1) lists all the parameter names and their associated data types.

EXAMPLE 4 (in recommended style)

```
REQ         Object-type { 'composite-logical-object' },
REQ         Position {
  REQ       #fixed-position {
    REQ     #horizontal-position { ANY_INTEGER },
    REQ     #vertical-position { ANY_INTEGER }
                        }
                }
```

F.3.3.5 Keywords

The keywords allowed for an attribute or parameter value are ANY_VALUE and VIRTUAL. ANY_VALUE denotes that the attribute or parameter may take any value permitted for it in the ITU-T Rec. T.410-Series | ISO/IEC 8613. VIRTUAL is used in factor constraints to denote that the attribute or parameter value will be determined in a constraint definition that references that factor constraint.

The format of the keyword constraint is as follows:

```
<keyword>                   ::= 'ANY_VALUE' | 'VIRTUAL'
```

EXAMPLE 5 (in recommended style)

```
PERM Dimensions                     { ANY_VALUE }
```

F.3.3.6 Content information constraint

A content information constraint is used to define the allowed values for the attribute "Content information". This type of constraint has two parts for content information structured in accordance with the character content architecture and a single terminal symbol for raster and geometric content information.

The format of the content information constraint is as follows:

```
<content-information>       ::= 'CHARACTER' [ '{' <control-function-list> '}' '...' ]
                                | 'RASTER'
                                | 'GEOMETRIC'

<control-function-list>     ::= <control-function> [ '|' <control-function> ]...

<control-function>          ::= '#' <control-function-name> [ '{' <attribute-value> '}' ]

<control-function-name>     -- any abbreviated control function name specified in ITU-T Rec. T.416 |
                               ISO/IEC 8613-6, see F.3.7 (Table F.2) --
```

EXAMPLE 6 (in recommended style)

```
DEFINE (CONTRLFNS,"
        { #CR
        | #SGR { [ 'cancel' | 'increased-intensity' | 'normal-intensity' ]... }
        | #SLS { ANY_INTEGER }
        }...")

PERM Content-information    { CHARACTER $CONTRLFNS },
```

F.3.3.7 Attribute value expression specifications and production rules

An attribute value expression defines the allowed range of values (that is both basic and non-basic values) that may be specified for that particular attribute.

NOTE – The document profile constraint specification defines the non-basic attribute values allowed and also defines the default values to be used for those attributes for which a non-standard default value is specified.

The attribute value expression specification allows one or more value types to be specified by means of sequences, choices, sets and repetitions of value expressions. Elements of a sequence construction are separated by a space. Elements of a set construction are separated by a comma. Elements of a choice construction are separated by a vertical bar, i.e. "|". Elements that are repeated are suffixed with an ellipsis, i.e. "...". Elements enclosed in square brackets, i.e. "[" and "]" are optional. Elements are grouped using curly brackets, i.e. "{" and "}". In the cases where an attribute value type allows the specification of a set of production rules to define the allowed values, each production rule in the set shall be terminated by a semicolon ";". The value type specification consists of one of fourteen different types of constraint as defined below.

The format of the attribute expression specification is as follows:

```
<attribute-value-expr>      ::= <production-rule>...
                                <value-expr-a>

<production-rule>           ::= non-terminal-symbol> '::=' <value-expr-a> ';'

<non-terminal-symbol>       -- A document application profile non-terminal symbol. It consists of a sequence of
                               one or more characters enclosed in angle brackets ('<' and '>'). The first character
                               shall be a lowercase letter ('a'..'z'). The remaining characters shall be lowercase
                               letters ('a'..'z'), digits ('0'..'9') or hyphens ('-'). Within a given set of production rules,
                               each non-terminal symbol referenced shall be defined once and only once and
                               each non-terminal symbol defined shall be referenced at least once. --

<value-expr-a>              ::= <value-expr-b> ['|' <value-expr-b> ]...

<value-expr-b>              ::= <value-expr-c>
                                | <value-expr-c> <value-expr-c>...
                                | <value-expr-c> { ',' <value-expr-c> }...

<value-expr-c>              ::= '{' <value-expr-a> '}' [ '...' ]
                                | '[' <value-expr-a> ']' [ '...' ]
                                | <value-type> [ '...' ]

<value-type>                ::= <non-terminal-symbol>
                                | <enumerated-type>
                                | <integer-value>
                                | <real-value>
                                | <character-string>
                                | <escape-sequence>
                                | <asn.1-object-id>
                                | <reference-to-object-id>
                                | <reference-to-object-class-id>
                                | <reference-to-style-id>
                                | <reference-to-subordinates>
                                | <reference-to-content-portions>
                                | <expr-function>
                                | <invocation-control-function>
```

F.3.3.7.1 Enumerated data type constraint

The enumerated data type constraint is used when an attribute or parameter consists of an enumerated data type as defined in the ITU-T Rec. T.410-Series | ISO/IEC 8613. The specification consists of an enumerated data type value.

The format of the enumerated data type constraint is as follows:

```
<enumerated-type>           -- any enumerated data value specified in the ITU-T Rec. T.410-Series |
                               ISO/IEC 8613, enclosed in single quotes, see F.3.6 (Table F.1) --
```

EXAMPLE 7 (in recommended style)

```
REQ     Layout-path     { 'd0'
                        | 'd90'
                        | 'd270'
                        }
```

F.3.3.7.2 Integer value constraint

An integer value constraint may be one of four different types of specification. These are: an integer, a relational specification, the specification of an inclusive range between two integers, and the keyword ANY_INTEGER. ANY_INTEGER is used to specify any integer value allowed by the ITU-T Rec. T.410-Series | ISO/IEC 8613.

The format of the integer value constraint is as follows:

```
<integer-value>          ::= <integer>
                             | <relational-operator> <integer>
                             | <integer> '..' <integer>
                             | 'ANY_INTEGER'
<integer>                -- any integer constant --

<relational-operator>    ::= '>' | '<' | '>=' | '<='
```

Note – The relational operator "< >" may be implemented using the 'EXCEPT' mechanism (see F.3.3.3.2).

EXAMPLE 8 (in recommended style)

```
PERM Dimensions      {
  PERM               #horizontal-dimension {
    PERM               #fixed-dimension { <= 9240 }
                                        }

  PERM               #vertical-dimension {
    PERM               #fixed-dimension { 20..12400 }
                                        }
                     }
```

F.3.3.7.3 Real value constraint

The real value constraint is used when an attribute or parameter consists of a real number specification.

The format of the real value constraint is as follows:

```
<real-value>             ::= <real>

<real>                   ::= -- any real number --
```

EXAMPLE 9 (in recommended style)

```
PERM       #colour-specification {
  PERM     #cmyk-colour {
    PERM   #c-value { 0.1200..0.3200 }
    PERM   #m-value { 0.3800..0.4700 }
    PERM   #y-value { 0.5100..0.6800 }
    PERM   #k-value { 0.700..0.8100 }
                    }
                              }
```

F.3.3.7.4 Character string constraint

A character string constraint is used to define the allowed character strings for an attribute or parameter. 'ANY_STRING' is used to specify any string allowed by the ITU-T Rec. T.410-Series | ISO/IEC 8613 consistent with the document application profile.

The format of the character string constraint is as follows:

```
<character-string>       ::= <printable-string>
                             | '\' <octet>...'\'
                             | 'ANY_STRING'

<printable-string>       -- any string of printable characters enclosed in double quotes. A double quote in the
                            string is announced by another double quote --

<octet>                  ::= <integer> '/' <integer>

                         -- where each integer is in the range 00..15 --
```

F.3.3.7.5 Escape sequence constraint

This form of constraint is used when an attribute or parameter value consists of an escape sequence used for control purposes. Escape sequences are described in terms of the ESC terminal symbol followed by one or more characters defined by row and column positions in a character code table and optionally followed by a single invocation control function.

The format of the escape sequence constraint is as follows:

```
<escape-sequence>              ::= 'ESC' <octet>... [ <invocation-control-function> ]

<invocation-control-function>  ::= { 'SI' | 'SO' | 'LS0' | 'LS1' | 'LS1R' | 'LS2' | 'LS2R' | 'LS3' | 'LS3R' |
                                   'SS2' | 'SS3' }
```

EXAMPLE 10 (in recommended style)

```
-- Non-basic document characteristics --
PERM Presentation-features  {
  PERM                  #character-presentation-features {
    PERM                #graphic-character-sets { ESC 02/15 04/06, ESC 07/12 }
      -- Greek in G3 (Designation and Invocation) --
                                                          }
                        }
```

F.3.3.7.6 ASN.1 object identifier constraint

An ASN.1 object identifier constraint is used to define the allowed ASN.1 object identifiers for an attribute or parameter. This type of constraint allows a sequence of integers separated by spaces, as the possible attribute or parameter value.

The format of the ASN.1 object identifier is as follows:

```
<asn.1-object-id>              ::= 'ASN.1' '{' <integer>...'}'
```

EXAMPLE 11 (in recommended style)

```
DEFINE (FC,          "ASN.1 { 2 8 2 6 0 }")
DEFINE (PC,          "ASN.1 { 2 8 2 6 1 }")
DEFINE (FPC,         "ASN.1 { 2 8 2 6 2 }")

REQ  Content-architecture-class          { $FC | $PC | $FPC }
```

F.3.3.7.7 References to objects, object classes, styles, subordinates and content portions

The references to objects, object classes, styles, subordinates, and content portions are used when an attribute or parameter consists of an identifier of an object, object class, layout or presentation style or the sequence of numeric strings that implicitly identify subordinates or content portions. The constraint specification consists of a keyword followed by a document application profile constituent name corresponding to one of the possible constituent constraints.

The format of the reference to objects, object classes, content portions, and styles constraint is as follows:

```
<reference-to-object-id>          ::= 'OBJECT_ID_OF' '(' <c-name> ')'

<reference-to-object-class-id>    ::= 'OBJECT_CLASS_ID_OF' '(' <c-name> ')'

<reference-to-style-id>           ::= 'STYLE_ID_OF'  '(' <c-name> ')'

<reference-to-subordinates>       ::= 'SUB_ID_OF' '(' <c-name> ')' [ '+' ]

<reference-to-content-portions>   ::= 'CONTENT_ID_OF' '(' <c-name> ')' [ '+' ]
```

EXAMPLE 12 (in recommended style)

```
REQ Object-class          { OBJECT_CLASS_ID_OF (PageSet) },
PERM Presentation-style   { STYLE_ID_OF (PresStyle-1) },
```

F.3.3.7.8 Expression functions

This form of value type is used to specify the constraints for object identifier expressions, string expressions and construction expressions.

The format of an expression function is as follows:

```
<expr-function>          ::= <expr-token-word-0>
                           | <expr-token-word-1a> <one-parameter-a>
                           | <expr-token-word-1b> <one-parameter-b>
                           | <expr-token-word-2a> <two-parameter-a>
                           | <expr-token-word-2b> <two-parameter-b>

<one-parameter-a>        ::= <value-type>
                           | '{' <value-expr-a> '}'

<one-parameter-b>        ::= '(' <value-expr-a> ')'

<two-parameter-a>        ::= '(' <class-or-type> ',' <object-reference> ')'

<two-parameter-b>        ::= '('<value-expr-a> ')' '(' <value-expr-a> ')'

<class-or-type>          ::= <value-expr-c> | <object-type>

<object-type>            ::= 'DLAR' | 'DOCUMENT_LAYOUT_ROOT' | 'PAGESET' | 'PAGE' | 'FRAME' |
                             'BLOCK' | 'DLOR' | 'DOCUMENT_LOGICAL_ROOT' | 'CLO' |
                             'COMPOSITE_LOGICAL' | 'BLO' | 'BASIC_LOGICAL'

<object-reference>       ::= <value-expr-c> | '(' <value-exp-c> ')'
```

The expression token words are those words used as terminal symbols in ITU-T Rec. T.412 | ISO/IEC 8613-2. The association of these words is as follows:

```
<expr-token-word-0> ::=
        { 'CURR-OBJ' | 'CURRENT-OBJECT' }

<expr-token-word-1a>    ::=
        { 'OPT' | 'OPT REP' | 'REP ' }

<expr-token-word-1b> ::=
        { 'AGG'       |  'AGGREGATE'        |  'CHO'        |  'CHOICE'
        'DEC'         |  'DECREMENT'        |  'INC'        |  'INCREMENT'
        'L-ALPHA'     |  'LOWER-ALPHA'      |  'L-ROM'      |  'LOWER-ROMAN'
        'MK-STR'      |  'MAKE-STRING'      |  'ORD'        |  'ORDINAL'
        'PREC'        |  'PRECEDING'        |  'PREC-OBJ'   |  'PRECEDING-OBJECT'
        'SEQ'         |  'SEQUENCE'         |  'SUP'        |  'SUPERIOR'
        'SUP-OBJ'     |  'SUPERIOR-OBJECT'  |  'U-ALPHA'    |  'UPPER-ALPHA'
        'U-ROM'       |  'UPPER-ROMAN' }

<expr-token-word-2a> ::=

        { 'CURR-INST' |  'CURRENT-INSTANCE' }

<expr-token-word-2b> ::=

        { 'B_REF'     |  'BINDING_REFERENCE' }
```

F.3.3.7.8.1 Object identifier expression constraint

An object identifier expression constraint is an expression function used to define the allowed object identifier expressions for an attribute or parameter. This type of constraint is specified by means of a set of production rules using the meta symbols defined in F.3.2.1. The exception to this rule is that the apostrophe symbols "'" and "'" used to delimit terminals in the Document application profile proforma are not used in a document application profile specification.

The following restrictions also apply :

a) The start symbol shall be the non-terminal <object-id-expr>.

b) The terminal symbols are restricted to those defined in ITU-T Rec. T.412 | ISO/IEC 8613-2 together with 'OBJECT_CLASS_ID_OF', 'OBJECT_ID_OF', '(', ')' and any document application profile constituent name.

c) The set of production rules shall be defined such that they always generate an object identifier expression which is in accordance with the rules defined in ITU-T Rec. T.412 | ISO/IEC 8613-2 and each production rule in the set shall be terminated by a semicolon ";".

EXAMPLE 13 (in recommended style)

```
PERM  Same-layout-object  {
        PERM                    #to-logical-object {
          <object-id-expr>      ::= PREC-OBJ (CURR-OBJ);
                                                  },
        PERM                    #to-layout-object { 'page' }
                        }
```

F.3.3.7.8.2 String expression constraint

A string expression constraint is an expression function used to define the allowed string expressions for an attribute or parameter. This type of constraint is specified by means of a set of production rules using the meta symbols defined in F.3.2.1. The exception to this rule is that the apostrophe symbols "'" and "'" used to delimit terminals in the Document application profile proforma are not used in a document application profile specification.

The following restrictions also apply:

a) The start symbol shall be the non-terminal <string-expr>.

b) The terminal symbols are restricted to those defined in ITU-T Rec. T.412 | ISO/IEC 8613-2 together with 'OBJECT_CLASS_ID_OF', 'OBJECT_ID_OF', '(', ')', 'ANY_STRING', 'ANY_INTEGER', any document application profile constituent name and any Document application profile defined character string enclosed in quotation marks, i.e. "...".

c) The set of production rules shall be defined such that they always generate a string expression which is in accordance with the rules defined in ITU-T Rec. T.412 | ISO/IEC 8613-2 and each production rule in the set shall be terminated by a semicolon ";".

d) The production rule for integer value constraint in F.3.3.7.2 is used in place of the non-terminal <numeric literal> production rule in ITU-T Rec. T.412 | ISO/IEC 8613-2.

EXAMPLE 14 (in recommended style)

```
DEFINE (PAGENUMBERSTRING, "

        <string-expr>           ::= [ ANY_STRING ]
                                    INC (B_REF (PREC (CURR-OBJ)) (" "PGnum" "))
                                    [ ANY_STRING ]; " )

PERM  Content-generator         { $PAGENUMBERSTRING },
```

F.3.3.7.8.3 Construction expression constraint

A construction expression constraint is an expression function used to define the allowed values for the attribute "Generator for subordinates". This type of constraint is specified by means of a set of production rules using the meta symbols defined in F.3.2.1. The exception to this rule is that the apostrophe symbols "'" and "'" used to delimit terminals in the Document application profile proforma are not used in a document application profile specification.

The following restrictions also apply :

a) The start symbol shall be the non-terminal <construction-expr>.

b) The terminal symbols are restricted to those defined in ITU-T Rec. T.412 | ISO/IEC 8613-2 together with 'OBJECT_CLASS_ID_OF' and any document application profile constituent name.

c) The set of production rules shall be defined such that they always generate a construction expression which is in accordance with the rules defined in ITU-T Rec. T.412 | ISO/IEC 8613-2 and each production rule in the set shall be terminated by a semicolon ";".

EXAMPLE 15 (in recommended style)

```
DEFINE (POSS-A, "

        <construction-expr>     ::= <object-class>
                                  | CHO (<object-class>...)
                                  | OPT <object-class>;

        <object-class>          ::= OBJECT_CLASS_ID_OF (A) ; ")

REQ  Generator-for-subordinates         { $POSS-A }
```

EXAMPLE 16 (in recommended style)

```
DEFINE (ITERANY--A-B-C, "

        <construction-expr>     ::= <construction-term>
                                  | <construction-type> ;

        <construction-term>     ::= <construction-factor>
                                  | OPT <construction-factor>
                                  | REP <construction-factor>
                                  | OPT REP <construction-factor>;

        <construction-type>     ::= SEQ (<construction-term>...)
                                  | AGG (<construction-term>...)
                                  | CHO (<construction-term>...);

        <construction-factor>   ::= OBJECT_CLASS_ID_OF (A)
                                  | OBJECT_CLASS_ID_OF (B)
                                  | OBJECT_CLASS_ID_OF (C)
                                  | <construction-type> ; ")

REQ  Generator-for-subordinates     { $ITERANY--A-B-C },
```

F.3.3.7.9 Invocation control function constraint

This form of value type constraint is used when an attribute or parameter value consists of an invocation sequence. Invocation control function constraint specification consists of a list of possible invocation control functions allowed by the ITU-T Rec. T.410-Series | ISO/IEC 8613. The format of the invocation control function constraint is as follows:

```
<invocation-control-function>   ::= { 'S0', 'LS0', 'S1', 'LS1', 'LS1R', 'LS2', 'LS2R', 'LS3', 'LS3R', 'SS2', 'SS3' }
```

EXAMPLE 17 (in recommended style)

```
PERM Presentation-attributes    {
  PERM                              #character-attributes {
    PERM                            #graphic-character-sets { S1 }
                                                }
                                }
```

F.3.4 Summary of production rules

```
<document-application-profile>  ::= { <macro-definition> | <constraint-definition> }...

<macro-definition>              ::= 'DEFINE' '(' <macro-name> ',' <macro-string> ')'

<macro-name>                    -- A unique name that identifies a document application profile macro name. It
                                   consists of a sequence of one or more characters. The first character shall be a
                                   letter ('a'..'z', 'A'..'Z'). The remaining characters shall be the letters ('a'..'z',  'A'..'Z'),
                                   digits ('0'..'9') or hyphens ('-'). Uppercase and lowercase letters are not significant
                                   but their use throughout a document application profile should be consistent. --

<macro-string>                  ::= <printable-string>

<constraint-definition>         ::= <constraint-name> '{' <constraint> '}'

<constraint-name>               ::= <constituent-name> | <factor-name>

<constituent-name>              ::= <c-name> [ <factor-reference> ]

<c-name>                        -- A unique name that identifies a document application profile constituent constraint.
                                   It  consists of a sequence of  one or more characters. The first character shall be
                                   an uppercase letter ('A'..'Z'). The remaining characters shall be uppercase letters
                                   ('A'..'Z'), digits ('0'..'9') or hyphens ('-'). --

<factor-reference>              ::= ':' <f-name>

<factor-name>                   ::= 'FACTOR' <f-name> [ <factor-reference> ]
```

```
<f-name>                    -- A unique name that identifies a document application profile factored constraint. It
                               consists of a sequence of one or more characters. The first character shall be an
                               uppercase letter ('A'..'Z'). The remaining characters shall be uppercase letters
                               ('A'..'Z'), digits ('0'..'9') or hyphens ('-'). --

<constraint>                ::= <section>...| <attribute-expr-a>

<section>                   ::= { 'GENERIC' | 'SPECIFIC' | 'SPECIFIC_AND_GENERIC' } ':' <attribute-expr-a>

<attribute-expr-a>          ::= <attribute-expr-b> [ '|' <attribute-expr-b> ]...

<attribute-expr-b>          ::= <attribute-expr-c> [ ',' <attribute-expr-c> ]...

<attribute-expr-c>          ::= <case-attribute> | <simple-attribute>

<case-attribute>            ::= 'CASE' <reference-type> 'OF' '{' <single-case-attribute>... '}'

<reference-type>            ::= <functional-reference> | <direct-reference>

<functional-reference>      ::= <function> '(' <direct-reference> ')'

<function>                  ::= 'SUPERIOR'

<direct-reference>          ::= <c-name> '(' <attribute-reference> ')'

<attribute-reference>       ::= <attribute-name> [ '#' <parameter-name> ]...

<single-case-attribute>     ::= <case-value> ':' <attribute-expr-a>

<case-value>                ::= <attribute-value-except> | 'VOID'

<simple-attribute>          ::= { 'REQ' | 'PERM' | 'DIS' } <attribute-name>
                                <attribute-value-except>

<attribute-name>            -- a name that identifies an attribute from the ITU-T Rec. T.410-Series |
                               ISO/IEC 8613, see F.3.6 (Table F.1) --

<attribute-value-except>    ::= '{' <attribute-value> '}'
                                [ 'EXCEPT' '{' <attribute-value> '}' ]

<attribute-value>           ::= -- empty --
                                | <parameter-expr-a>
                                | <keyword>
                                | <content-information>
                                | [ 'NOVR' ] <attribute-value-expr>

<parameter-expr-a>          ::= <parameter-expr-b> [ '|' <parameter-expr-b> ]...

<parameter-expr-b>          ::= <parameter-expr-c>
                                | <parameter-expr-c> <parameter-expr-c>...
                                | <parameter-expr-c> { ',' <parameter-expr-c> }...

<parameter-expr-c>          ::= <case-parameter>
                                | <simple-parameter>
                                | 'REQ' '{' <parameter-expr-a> '}'
                                | 'PERM' '{' <parameter-expr-a> '}'
                                | 'MUL' '{' <parameter-expr-a> '}'
                                | 'PMUL' '{' <parameter-expr-a> '}'

<case-parameter>            ::= 'CASE' <reference-type> 'OF' '{' <single-case-parameter>... '}'

<single-case-parameter>     ::= <case-value> ':' <attribute-value-except>

<simple-parameter>          ::= { 'REQ' | 'PERM' | 'DIS' } '#' <parameter-name>
                                <attribute-value-except>

<parameter-name>            -- a name that identifies a (sub-)parameter from the ITU-T Rec. T.410-Series |
                               ISO/IEC 8613, see F.3.6 (Table F.1) --

<keyword>                   ::= 'ANY_VALUE' | 'VIRTUAL'

<content-information>       ::= 'CHARACTER' [ '{' <control-function-list> '}' '...' ]
                                | 'RASTER'
                                | 'GEOMETRIC'
```

```
<control-function-list>      ::= <control-function> [ '|' <control-function> ]...

<control-function>           ::= '#' <control-function-name> [ '{' <attribute-value> '}' ]

<control-function-name>      -- any abbreviated control function name specified in ITU-T Rec. T.416 |
                                ISO/IEC 8613-6, see 7.3.7 (Table F.2) --

<attribute-value-expr>       ::= <production-rule>...
                                 | <value-expr-a>

<production-rule>            ::= <non-terminal-symbol> '::=' <value-expr-a> ';'

<non-terminal-symbol>        -- A document application profile non-terminal symbol. It consists of a sequence of
                                one or more characters enclosed in angle brackets ('<' and '>'). The first character
                                shall be a lowercase letter ('a'..'z'). The remaining characters shall be lowercase
                                letters ('a'..'z'), digits ('0'..'9') or hyphens ('-'). Within a given set of production rules,
                                each non-terminal symbol referenced shall be defined once and only once and
                                each non-terminal symbol defined shall be referenced at least once. --

<value-expr-a>               ::= <value-expr-b> [ '|' <value-expr-b> ]...

<value-expr-b>               ::= <value-expr-c>
                                 | <value-expr-c> <value-expr-c>...
                                 | <value-expr-c> { ',' <value-expr-c> }...

<value-expr-c>               ::= '{' <value-expr-a> '}' [ '...' ]
                                 | '[' <value-expr-a> ']' [ '...' ]
                                 | <value-type> [ '...' ]

<value-type>                 ::= <non-terminal-symbol>
                                 | <enumerated-type>
                                 | <integer-value>
                                 | <real-value>
                                 | <character-string>
                                 | <escape-sequence>
                                 | <asn.1-object-id>
                                 | <reference-to-object-id>
                                 | <reference-to-object-class-id>
                                 | <reference-to-style-id>
                                 | <reference-to-subordinates>
                                 | <reference-to-content-portions>
                                 | <expr-function>
                                 | <invocation-control-function>

<enumerated-type>            -- any enumerated data  value specified in the ITU-T Rec. T.410-Series
                                ISO/IEC 8613 enclosed in single quotes, see F.3.6 (Table F.1) --

<integer-value>              ::= <integer>
                                 | <relational-operator> <integer>
                                 | <integer> '..' <integer>
                                 | 'ANY_INTEGER'

<integer>                    -- any integer constant --

<relational-operator>        ::= '>' | '<' | '>=' | '<='

<real-value>                 ::= <real>

<real>                       -- any real constant --

<character-string>           ::= <printable-string>
                                 | '\' <octet>... '\'
                                 | 'ANY_STRING'

<printable-string>           -- any string of printable characters enclosed in double quotes. A double quote in the
                                string is announced by another double quote --

<octet>                      ::= <integer> '/' <integer>

                             -- where each integer is in the range 00..15 --
```

```
<escape-sequence>                ::= [ 'ESC' <octet>... | <invocation-control-function> ]

<invocation-control-function>    ::= { 'SI' | 'SO' | 'LS0' | 'LS1' | 'LS1R' | 'LS2' | 'LS2R' | 'LS3' | 'LS3R' |
                                     'SS2' | 'SS3' }

<asn.1-object-id>                ::= 'ASN.1' '{' <integer>... '}'

<reference-to-object-id>         ::= 'OBJECT_ID_OF' '(' <c-name> ')'

<reference-to-object-class-id>   ::= 'OBJECT_CLASS_ID_OF' '(' <c-name> ')'

<reference-to-style-id>          ::= 'STYLE_ID_OF' '(' <c-name> ')'

<reference-to-subordinates>      ::= 'SUB_ID_OF' '(' <c-name> ')' [ '+' ]

<reference-to-content-portions>  ::= 'CONTENT_ID_OF' '(' <c-name> ')' [ '+' ]

<expr-function>                  ::= <expr-token-word-0>
                                     | <expr-token-word-1a > <one-parameter-a>
                                     | <expr-token-word-1b > <one-parameter-b>
                                     | <expr-token-word-2a > <two-parameter-a>
                                     | <expr-token-word-2b > <two-parameter-b>

<one-parameter-a>                ::= <value-type>
                                     | '{' <value-expr-a> '}'

<one-parameter-b>                ::= '(' <value-expr-a> ')'

<two-parameter-a>                ::= '(' <class-or-type> ',' <object-reference> ')'

<two-parameter-b>                ::= '(' <value-expr-a> ')' '(' <value-expr-a> ')'

<class-or-type>                  ::= <value-expr-c> | <object-type>

<object-type>                    ::= 'DLAR' | 'DOCUMENT_LAYOUT_ROOT' | 'PAGESET' | 'PAGE' | 'FRAME' |
                                     'BLOCK' | 'DLOR' | 'DOCUMENT_LOGICAL_ROOT' | 'CLO' |
                                     'COMPOSITE_LOGICAL' | 'BLO' | 'BASIC_LOGICAL'

<object-reference>               ::= <value-expr-c> '(' <value-exp-c> ')'
```

The expression token words are those words used as terminal symbols in ITU-T Rec. T.412 | ISO/IEC 8613-2. The association of these words is as follows:

```
<expr-token-word-0> ::=
        { 'CURR-OBJ' | 'CURRENT-OBJECT' }

<expr-token-word-1a>    ::=
        { 'OPT' | 'OPT REP' | 'REP ' }

<expr-token-word-1b> ::=
        { 'AGG'       |  'AGGREGATE'        |  'CHO'          |  'CHOICE'
        'DEC'         |  'DECREMENT'        |  'INC'          |  'INCREMENT'
        'L-ALPHA'     |  'LOWER-ALPHA'      |  'L-ROM'        |  'LOWER-ROMAN'
        'MK-STR'      |  'MAKE-STRING'      |  'ORD'          |  'ORDINAL'
        'PREC'        |  'PRECEDING'        |  'PREC-OBJ'     |  'PRECEDING-OBJECT'
        'SEQ'         |  'SEQUENCE'         |  'SUP'          |  'SUPERIOR'
        'SUP-OBJ'     |  'SUPERIOR-OBJECT'  |  'U-ALPHA'      |  'UPPER-ALPHA'
        'U-ROM'       |  'UPPER-ROMAN' }

<expr-token-word-2a> ::=

        { 'CURR-INST' |  'CURRENT-INSTANCE' }

<expr-token-word-2b> ::=

        { 'B_REF'     |  'BINDING_REFERENCE' }
```

F.3.5 Presentation of document application profile specification

This annex does not define a presentation style for the document application profile specification. However, a consistent style of presentation should be adopted to aid readability. The following is an example of a style that may be used.

```
Layout-Object-Block {

SPECIFIC:

    REQ     Object-type                     { 'block' },
    REQ     Object-identifier               { OBJECT_ID_OF (Block1) },
    REQ     Content-portions                { CONTENT_ID_OF
                                              (Character-content-portion-1) + },
    PERM    Presentation-style              { STYLE_ID_OF (Pres-style-1) },
    PERM    Content-architecture-class      {  ASN.1 { 2 8 2 6 0 }
                                            | ASN.1 { 2 8 2 6 2 }
                                            },
    PERM    Presentation-attributes         {
      -- Character presentation attributes --
      PERM  #alignment                        { 'start-aligned' | 'justified' },
      PERM  #character-spacing                { 100 | 200 },
      PERM  #first-line-offset                { ANY_INTEGER },
      PERM  #graphic-rendition                { [ 'cancel'
                                                  | 'increased-intensity'
                                                  | 'italicized'
                                                  | 'underlined
                                                  | 'normal-intensity'
                                                  | 'not-italicized'
                                                  | 'not-underlined'
                                                ]...
                                              },
      PERM  #itemization                      { ANY_VALUE },
      PERM  #line-layout-table                {
        PMUL                                          {
          PERM  #tabulation-reference                       { ANY_VALUE },
          PERM  #tabulation-position                        { ANY_INTEGER },
          PERM  #alignment                                  { 'start-aligned'
                                                            | 'aligned-around'
                                                            },
          PERM  #alignment-string                           { ANY_STRING }
                                                      }
                                              },
    PERM    #line-spacing                   { 200 | 300 | 400 },
    PERM    #initial-offset                 {
      PERM  #horizontal-coordinate                  { 200 }
      PERM  #vertical-coordinate                    { 200 }
                                            }
                                            },
    PERM    User-readable-comments          { ANY_STRING },
    PERM    User-visible-name               { "BLOCK" },
    REQ   Position                          {
      REQ   #fixed-position                   {
        REQ   #horizontal-dimension           { ANY_INTEGER }
        REQ   #vertical-dimension             { ANY_INTEGER }
                                              }
                                            },
    REQ   Dimensions                        {
        REQ   #horizontal-dimension           {
          REQ   #fixed-dimension              { ANY_INTEGER }
                                              },
        REQ   #vertical-dimension             {
          REQ   #fixed-dimension              { ANY_INTEGER }
                                              }
                                            }
```

F.3.6 Table F.1 ITU-T Rec. T.410-Series | ISO/IEC 8613 attribute/(sub-) parameter names and their associated DAPPN data types

The association between the ITU-T Rec. T.410-Series | ISO/IEC 8613 attribute/(sub-) parameter names and DAPPN data types is specified in Table F.1. In Table F.1:

- each (sub-) parameter of an attribute is preceded by the hash symbol: "#";
- comments are delimited by a pair of hyphen symbols: "--";
- more than one occurrence of a name or data type is indicated using the ellipsis symbol: "...". If an ellipsis appears after an attribute/(sub-) parameter name, it specifies that the attribute/(sub-) parameter and all previous attribute/(sub-) parameters that are part of that declaration description and which are at the same indentation level as the immediately preceding attribute/(sub-) parameter, may be repeated. If an ellipsis appears for a data type specification declaration, it specifies that only the immediately preceding data type in the data type specification may be repeated. When more than one data type in a data type specification may be repeated, the repeatable data types are grouped using parentheses, "(" and ")", and the ellipsis immediately follows the closing parenthesis: ")";
- for <enumeration-type> data types the names of the enumerations follow the colon symbol: ":";
- in the case of attributes or (sub-) parameters that have more than one data type the possible types are separated by the comma symbol: ",";
- macros are also used as defined in F.3.3.1;
- the comment -- empty -- implies that the value of the attribute or parameter name has no corresponding value or, in the case of a control function, that the control function has no control function parameter value;
- the data type specifications do not include the <keyword> data type that is allowed for any attribute or parameter value specification;
- use of presentation attributes in the document profile shall be restricted to the specification of those non-basic attribute values which should be used in the document;
- the presentation attribute values specified in the attribute "Document-application-profile-defaults" for a particular document profile shall be used in place of the default values specified in this document for those same attributes, as needed, in order to process unambiguously the particular document instance associated with the document profile.

F.3.7 Table F.2 ITU-T Rec. T.416 | ISO/IEC 8613-6 character content information control function names and their associated DAPPN data types

The association between ITU-T Rec. T.416 | ISO/IEC 8613-6 character content information control function names and DAPPN data types is specified in Table F.2. In Table F.2:

- comments are delimited by a pair of hyphen symbols: "--" :
- for <enumeration-type> data types the names of the enumerations follow the colon symbol: ":" ;
- the comment -- empty -- implies that the control function has no control function parameter value.

Table F.1 – List of attributes and data types

Attribute/(Sub) Parameter Name	Data Type Specification
.. macro definitions ..	
DEFINE (case-i-dimensions-val,"	
#horizontal-dimension	
#fixed-dimension	-- *empty* --, <integer-value>
#vertical-dimension	
#fixed-dimension	-- *empty* --, <integer-value>
")	
DEFINE (min-max,"	
#minimum	<integer-value>
#maximum	<integer-value>
")	
DEFINE (case-ii-dimensions-val,"	
#horizontal-dimension	
#fixed-dimension	-- *empty* --, <integer-value>
#rule-a	-- *empty* --, $min-max
#rule-b	-- *empty* --, $min-max
#maximum-size	<enumeration-type>:'applies'
#vertical-dimension	
#fixed-dimension	-- *empty* --, <integer-value>
#rule-a	-- *empty* --, $min-max
#rule-b	-- *empty* --, $min-max
#maximum-size	<enumeration-type>:'applies'
")	
DEFINE (case-iii-dimensions-val,"	
#horizontal-dimension	
#fixed-dimension	<enumeration-type>:'null', <integer-value>
#vertical-dimension	
#fixed-dimension	<enumeration-type>:'null', <integer-value>
#variable-page-height	<integer-value>
")	
DEFINE (colour-exp,"	
#colour-access-mode	<enumeration-type>:'direct' 'indirect'
#direct-colour-expression	
#colour-space-id	<integer-value>
#colour-specification	<integer-value>...
#cmyk-colour	
#c-value	<integer-value>, <real-value>
#m-value	<integer-value>, <real-value>
#y-value	<integer-value>, <real-value>
#k-value	<integer-value>, <real-value>
#rgb-colour	
#r-value	<integer-value>, <real-value>
#g-value	<integer-value>, <real-value>
#b-value	<integer-value>, <real-value>
#cie-colour	
#x-value	<integer-value>, <real-value>
#y-value	<integer-value>, <real-value>
#z-value	<integer-value>, <real-value>
#colour-tolerance	
#specified-tolerance	
#tolerance-value	<integer-value>
#tolerance-space	<enumeration-type>:'CIELUV' 'CIELAB'
#unspecified-tolerance	<enumeration-type>:'infinite'
#indexed-colour-expression	
#index	<integer-value>
")	
DEFINE (colour-tbl,"	
#colour-space-id	<integer value>
#colour-table-entry	
#index	<integer-value>
#colour-coordinates	<integer-value>...
#colour-tolerance	
#specified-tolerance	
#tolerance-value	<integer-value>
#tolerance-space	<enumeration-type>:'CIELUV' 'CIELAB'
#unspecified-tolerance	<enumeration-type>:'infinite'
...	
")	

Table F.1 – List of attributes and data types *(continued)*

Attribute/(Sub) Parameter Name	Data Type Specification
DEFINE (offset-val,"	
#leading-offset	<integer-value>
#trailing-offset	<integer-value>
#left-hand-offset	<integer-value>
#right-hand-offset	<integer-value>
")	
DEFINE (separation-val,"	
#leading-edge	<integer-value>
#trailing-edge	<integer-value>
#centre-separation	<integer-value>
")	
DEFINE (edge,"	
#line-width	<integer-value>
#line-type	<enumeration-type>:'invisible' 'solid' 'dashed' 'dot' 'dash-dot' 'dash-dot-dot'
#free-space-width	<integer-value>
#line-colour	
#implementation-defined	-- *empty* --, <character-string>
#colour-expression	-- *empty* --, $colour-exp
")	
DEFINE (border-val,"	
#left-hand-edge	-- *empty* --, $edge
#right-hand-edge	-- *empty* --, $edge
#trailing-edge	-- *empty* --, $edge
#leading-edge	-- *empty* --, $edge
")	
DEFINE (medium-type-val,"	
#nominal-page-size	
#horizontal-dimension	<integer-value>
#vertical-dimension	<integer-value>
#side-of-sheet	<enumeration-type>:'unspecified' 'recto' 'verso'
#colour-of-medium	
#unspecified-colour	-- *empty* --, <character-string>
#specified-colour	-- *empty* --, $colour-exp
")	
DEFINE (bundle-part-val,"	
#bundle-index	<integer-value>
#bundle-representation	<character-string>
...	
")	
DEFINE (name,"	
#surname	<character-string>
#givenname	<character-string>
#initials	<character-string>
#generation-qualifier	<character-string>
")	
DEFINE (method-info,"	
#unique-method-info	<asn.1-object-id>
#descriptive-method-info	<character-string>
")	
DEFINE (key-info,"	
#method-information	$method-info
#additional-information	
#descriptive-information	<character-string>
#octet-string	<character-string>
")	
DEFINE (seal-data,"	
#seal-method	
#fingerprint-method	$method-info
#fingerprint-key-information	$key-info
#sealing-method	$method-info
#sealing-key-information	$key-info
#sealing-information	
#fingerprint	<character-string>
#time	<character-string>
#seal-originator	$name
#location	
#unique-location	<asn.1-object-id>
#descriptive-location	<character-string>
#seal	<character-string>
")	

Table F.1 – List of attributes and data types *(continued)*

Attribute/(Sub) Parameter Name	Data Type Specification
DEFINE (sealed-doc-bodyparts,"	
#sealing-id	<integer-value>
#sealed-constituents	
#object-class-identifiers	<reference-to-object-class-id>...
#presentation-style-identifiers	<reference-to-style-id>...
#layout-style-identifiers	<reference-to-style-id>...
#object-identifiers	<reference-to-object-id>...
#privileged-recipients	$name...
#doc-body-part-seal	$seal-data
")	
DEFINE (protected-doc-parts,"	
#protected-doc-part-id	<integer-value> <integer-value>
#priv-recipients-info	
#privileged-recipients	$name...
#encipherment-method-info	$method-info
#encipherment-key-info	$key-info
")	
DEFINE (graphic-rendition-aspect,"	<enumeration-type>:'cancel' 'increased-intensity' 'decreased-intensity' 'italicized' 'underline' 'slowly-blinking' 'rapidly-blinking' 'negative-image' 'crossed-out' 'primary-font' 'first-alternative-font' 'second-alternative-font' 'third-alternative-font' 'fourth-alternative-font' 'fifth-alternative-font' 'sixth-alternative-font' 'seventh-alternative-font' 'eighth-alternative-font' 'ninth-alternative-font' 'doubly-underlined' 'normal-intensity' 'not-italicized' 'not-underlined' 'steady' 'variable-spacing' 'positive-image' 'not-crossed-out' 'black-foreground' 'red-foreground' 'green-foreground' 'yellow-foreground' 'blue-foreground' 'magenta-foreground' 'cyan-foreground' 'white-foreground' 'select-character-foreground-colour' 'black-background' 'red-background' 'green-background' 'yellow-background' 'blue-background' 'magenta-background' 'cyan-background' 'white-background' 'select-character-background-colour' 'not-variable-spacing'...
")	
DEFINE (char-pres-atts,"	
#character-path	<enumeration-type>:'d0' 'd90' 'd180' 'd270'
#line-progression	<enumeration-type>:'d90' 'd270'
#character-orientation	<enumeration-type>:'d0' 'd90' 'd180' 'd270'
#initial-offset	
#horizontal-coordinate	<integer-value>
#vertical-coordinate	<integer-value>
#character-spacing	<integer-value>
#line-spacing	<integer-value>
#alignment	<enumeration-type>:'start-aligned' 'end-aligned' 'centred' 'justified'
#line-layout-table	
#tabulation-reference	<integer-value>
#tabulation-position	<integer-value>
#alignment	<enumeration-type>:'start-aligned' 'end-aligned' 'centred' 'aligned-around'
#alignment-character-string	<character-string>
...	
#graphic-rendition	$graphic-rendition-aspect...
#formatting-indicator	<enumeration-type>:'no' 'yes'
#character-fonts	
#primary-font-spec	
#font-size	<integer-value>
#font-identifier	<integer-value>
#first-alternative-font-spec	
#font-size	<integer-value>
#font-identifier	<integer-value>
#second-alternative-font-spec	
#font-size	<integer-value>
#font-identifier	<integer-value>
#third-alternative-font-spec	
#font-size	<integer-value>
#font-identifier	<integer-value>
#fourth-alternative-font-spec	
#font-size	<integer-value>
#font-identifier	<integer-value>

Table F.1 – List of attributes and data types *(continued)*

Attribute/(Sub) Parameter Namc	Data Type Specification
#fifth-alternative-font-spec	
#font-size	<integer-value>
#font-identifier	<integer-value>
#sixth-alternative-font-spec	
#font-size	<integer-value>
#font-identifier	<integer-value>
#seventh-alternative-font-spec	
#font-size	<integer-value>
#font-identifier	<integer-value>
#eighth-alternative-font-spec	
#font-size	<integer-value>
#font-identifier	<integer-value>
#ninth-alternative-font-spec	
#font-size	<integer-value>
#font-identifier	<integer-value>
#graphic-character-subrepertoire	<integer-value>
#itemization	
#identifier-alignment	<enumeration-type>:'no-itemization' 'start-aligned' 'end-aligned'
#identifier-start-offset	<integer-value>
#identifier-end-offset	<integer-value>
#widow-size	<integer-value>
#orphan-size	<integer-value>
#graphic-character-sets	<escape-sequence>...
#indentation	<integer-value>
#kerning-offset	
#start-edge-offset	<integer-value>
#end-edge-offset	<integer-value>
#proportional-line-spacing	<enumeration-type>:'no' 'yes'
#pairwise-kerning	<enumeration-type>:'no' 'yes'
#first-line-offset	<integer-value>
#code-extension-announcers	<escape-sequence>...
")	
DEFINE (ras-pres-atts,"	
#pel-path	<enumeration-type>:'d0' 'd90' 'd180' 'd270'
#line-progression	<enumeration-type>:'d90' 'd270'
#pel-transmission-density	<enumeration-type>:'p6' 'p5' 'p4' 'p3' 'p2' 'p1'
#initial-offset	
#horizontal-coordinate	<integer-value>
#vertical-coordinate	<integer-value>
#clipping	
#first-coordinate-pair	
#x-coordinate	<integer-value>
#y-coordinate	<integer-value>
#second-coordinate-pair	
#x-coordinate	<integer-value>
#y-coordinate	<integer-value>
#pel-spacing	<enumeration-type>:'null'
#length	<integer-value>
#pel-spaces	<integer-value>
#spacing-ratio	
#line-spacing-value	<integer-value>
#pel-spacing-value	<integer-value>
#image-dimensions	
#width-controlled	
#minimum-width	<integer-value>
#preferred-width	<integer-value>
#height-controlled	
#minimum-height	<integer-value>
#preferred-height	<integer-value>
#area-controlled	
#minimum-width	<integer-value>
#preferred-width	<integer-value>
#minimum-height	<integer-value>
#preferred-height	<integer-value>
#aspect-ratio-flag	<enumeration-type>:'fixed' 'variable'
#automatic	<enumeration-type>:'null'
")	

Table F.1 – List of attributes and data types *(continued)*

Attribute/(Sub) Parameter Name	Data Type Specification
DEFINE (geo-pres-atts,"	
#encoding-announcer	<character-string>
#line-rendition	
#individual-part	<character-string>
#asf-part	
#line-type-asf	<enumeration-type>:'bundled' 'individual'
#line-width-asf	<enumeration-type>:'bundled' 'individual'
#line-colour-asf	<enumeration-type>:'bundled' 'individual'
#bundle-part	$bundle-part-val
#marker-rendition	
#individual-part	<character-string>
#asf-part	
#marker-type-asf	<enumeration-type>:'bundled' 'individual'
#marker-size-asf	<enumeration-type>:'bundled' 'individual'
#marker-colour-asf	<enumeration-type>:'bundled' 'individual'
#bundle-part	$bundle-part-val
#text-rendition	
#individual-part	<character-string>
#asf-part	
#text-font-asf	<enumeration-type>:'bundled' 'individual'
#text-precision-asf	<enumeration-type>:'bundled' 'individual'
#character-expansion-factor-asf	<enumeration-type>:'bundled' 'individual'
#character-spacing-asf	<enumeration-type>:'bundled' 'individual'
#text-colour-asf	<enumeration-type>:'bundled' 'individual'
#bundle-part	$bundle-part-val
#filled-area-rendition	
#individual-part	<character-string>
#pattern-table-part	<character-string>...
#asf-part	
#interior-style-asf	<enumeration-type>:'bundled' 'individual'
#fill-colour-asf	<enumeration-type>:'bundled' 'individual'
#hatch-index-asf	<enumeration-type>:'bundled' 'individual'
#pattern-index-asf	<enumeration-type>:'bundled' 'individual'
#bundle-part	$bundle-part-val
#edge-rendition	
#individual-part	<character-string>
#asf-part	
#edge-type-asf	<enumeration-type>:'bundled' 'individual'
#edge-width-asf	<enumeration-type>:'bundled' 'individual'
#edge-colour-asf	<enumeration-type>:'bundled' 'individual'
#bundle-part	$bundle-part-val
#colour-representations	
#background-colour	<character-string>
#colour-table-part	<character-string>
#transparency-specification	<character-string>
#transformation-of-layout-object-specification	<character-string>
#region-of-interest	
#automatic	<enumeration-type>:'null'
#rectangle	
#first-corner	<character-string>
#second-corner	<character-string>
#picture-orientation	<enumeration-type>:'d0' d90' 'd180' 'd270'
#picture-dimensions	
#width-controlled	
#minimum-width	<integer-value>
#preferred-width	<integer-value>
#height-controlled	
#minimum-height	<integer-value>
#preferred-height	<integer-value>
#area-controlled	
#minimum-width	<integer-value>
#preferred-width	<integer-value>
#minimum-height	<integer-value>
#preferred-height	<integer-value>
#aspect-ratio-flag	<enumeration-type>:'fixed' 'variable'
#automatic	<enumeration-type>:'null'
")	
DEFINE (page-set-attributes,"	
#layout-stream-categories	<character-string>...
#layout-stream-sub-categories	<character-string>...
")	

Table F.1 – List of attributes and data types *(continued)*

Attribute/(Sub) Parameter Name	Data Type Specification
DEFINE (page-attributes,"	
#dimensions	$case-i-dimensions-val
#transparency	<enumeration-type>:'transparent' 'opaque'
#content-architecture-class	<asn.1-object-id>
#presentation-attributes	$char-pres-atts | $ras-pres-atts | $geo-pres-atts
#external-content-architecture-presentation-attributes	*-- external --*
#page-position	
#horizontal-position	*-- empty --*, <integer-value>
#vertical-position	*-- empty --*, <integer-value>
#medium-type	$medium-type-val
#presentation-style	<reference-to-style-id>
#colour	<enumeration-type>:'colourless' 'white'
#colour-of-layout-object	$colour-exp
#object-colour-table	$colour-tbl
#content-background-colour	<enumeration-type>:'content-background-transparency', $colour-exp
#content-foreground-colour	<enumeration-type>:'implementation-defined' 'content-foreground-transparency', $colour-exp
#content-colour-table	$colour-tbl
#sealing	
#sealing-status	<enumeration-type>:'no' 'yes'
#sealing-ids	<integer-value>...
#layout-stream-categories	<character-string>...
#layout-stream-sub-categories	<character-string>...
")	
DEFINE (frame-attributes,"	
#position	
#fixed-position	
#horizontal-position	*-- empty --*, <integer-value>
#vertical-position	*-- empty --*, <integer-value>
#dimensions	$case-i-dimensions-val
#transparency	<enumeration-type>:'transparent' 'opaque'
#layout-path	<enumeration-type>:'d0' 'd90' d180' 'd270'
#permitted-categories	*-- empty --*, <character-string>
#colour	<enumeration-type>:'colourless' 'white'
#colour-of-layout-object	$colour-exp
#object-colour-table	$colour-tbl
#content-background-colour	<enumeration-type>:'content-background-transparency', $colour-exp
#content-foreground-colour	<enumeration-type>:'implementation-defined' 'content-foreground-transparency', $colour-exp
#content-colour-table	$colour-tbl
#border	*-- empty --*, $border-val
#sealing	
#sealing-status	<enumeration-type>:'no' 'yes'
#sealing-ids	<integer-value>...
#layout-stream-categories	<character-string>...
#layout-stream-sub-categories	<character-string>...
")	
DEFINE (block-attributes,"	
#position	
#fixed-position	
#horizontal-position	<integer-value>
#vertical-position	<integer-value>
#dimensions	$case-i-dimensions-val
#transparency	<enumeration-type>:'transparent' 'opaque'
#presentation-attributes	$char-pres-atts, $ras-pres-atts, $geo-pres-atts
#external-content-architecture-presentation-attributes	*-- external --*
#presentation-style	<reference-to-style-id>
#colour	<enumeration-type>:'colourless' 'white'
#colour-of-layout-object	$colour-exp
#object-colour-table	$colour-tbl
#content-background-colour	<enumeration-type>:'content-background-transparency', $colour-exp
#content-foreground-colour	<enumeration-type>:'implementation-defined' 'content-foreground-transparency', $colour-exp
#content-colour-table	$colour-tbl
#border	*-- empty --*, $border-val