INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

T.505

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(11/94)

TERMINALS FOR TELEMATIC SERVICES

DOCUMENT APPLICATION PROFILE PM-26 FOR THE INTERCHANGE OF ENHANCED STRUCTURE, MIXED CONTENT DOCUMENTS IN PROCESSABLE AND FORMATTED FORMS

ITU-T Recommendation T.505

(Previously “CCITT Recommendation”)

FOREWORD

The ITU-T (Telecommunication Standardization Sector) is a permanent organ of the International Telecommunication Union (ITU). The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, March 1-12, 1993).

ITU-T Recommendation T.505 was prepared by ITU-T Study Group 8 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 11th November 1994.

NOTE

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

CONTENTS

Page

1 Scope 1

2 Normative references 1

2.1 Paired Recommendations | International Standards equivalent in technical content 1

2.2 Additional references 3

3 Definitions 3

4 Relationship with other Profiles 4

5 Conformance 4

5.1 Data stream conformance 4

5.2 Implementation conformance 5

6 Characteristics supported by this document application profile 5

6.1 Overview 5

6.1.1 General 5

6.1.2 Formatted form documents 5

6.1.3 Processable form documents 5

6.1.4 Formatted processable form documents 6

6.1.5 Generic-documents 6

6.1.6 Document profile 6

6.2 Logical characteristics 7

6.2.1 Introduction 7

6.2.2 Overview of the logical structure 7

6.2.3 Body part of the logical structure 7

6.2.4 Common content part of the logical structure 11

6.3 Layout characteristics 12

6.3.1 Overview of the layout characteristics 12

6.3.2 DocumentLayoutRoot 14

6.3.3 PageSet 14

6.3.4 Page characteristics 15

6.3.5 Body area characteristics 17

6.3.6 Header and footer area characteristics 24

6.3.7 GenericBlock and SpecificBlock 25

6.4 Document layout characteristics 25

6.4.1 Flow controls 26

6.4.2 Layout of the document content 28

6.4.3 Layout controls applicable in the absence of a generic layout structure 30

6.5 Content layout and imaging characteristics 30

6.5.1 Character content 30

6.5.2 Raster graphics content 39

6.5.3 Geometric graphics content 41

6.6 Miscellaneous features 41

6.6.1 Resource-documents 41

6.6.2 External documents 42

6.6.3 Border 42

6.6.4 Application comments 42

6.6.5 Alternative representation 42

6.6.6 Automatic numbering mechanisms 43

6.6.7 User readable comments 46

6.6.8 User visible name 46

6.7 Document management features 47

6.7.1 Document constituent information 47

6.7.2 Document identification information 47

6.7.3 Document default information 48

6.7.4 Non-basic characteristics information 48

6.7.5 Fonts list 50

6.7.6 Document management attributes 50

Page

7 Specification of constituent constraints 46
7.1 Introduction 46
7.1.1 Diagrams of relationships of logical constituents 46
7.1.2 Diagrams of relationships of layout constituents 46
7.1.3 Notation 46
7.2 Document profile constituent constraints 46
7.2.1 Macro definitions 47
7.2.2 Constituents constraints 48
7.3 Logical constituent constraints 51
7.3.1 Macro definitions 51
7.3.2 Factor constraints 51
7.3.3 Constituent constraints 52
7.4 Layout constituent constraints 58
7.4.1 Macro definitions 59
7.4.2 Factor constraints 60
7.4.3 Constituent constraints 61
7.5 Layout style constituent constraints 81
7.5.1 Macro definitions 81
7.5.2 Factor constraints 81
7.5.3 Constituent constraints 82
7.6 Presentation style constituent contraints 83
7.6.1 Macro definitions 83
7.6.2 Factor constraints 83
7.6.3 Constituent constraints 83
7.7 Content portion constituent constraints 85
7.7.1 Macro definitions 85
7.7.2 Factor constraints 85
7.7.3 Constituent constraints 85

8 Interchange format 86
8.1 Interchange format 86
8.2 Document application profile object identifier 86
8.3 Encoding of application comments 87
8.4 Data lengths 87

Annex A – Amendments and technical corrigenda 87
A.1 Amendments 87
A.1.1 Amendments to the base standard 87
A.1.2 Proposed changes to standards due to defects 87
A.2 Technical corrigenda 87
A.2.1 Technical corrigenda to this Recommendation 87
A.3 Versions of the ODA Specification 87

Annex B – Recommended practices 88
B.1 Transfer methods for ODA 88
B.1.1 Conveyance of ODA over CCITT Recommendation X.400-1984 88
B.1.2 Conveyance of ODA over FTAM 88
B.1.3 Conveyance of ODA over DTAM 88
B.1.4 Conveyance of ODA over flexible disks 88
B.2 Font reference 88
B.3 ISO/IEC 8632 (CGM) constraints for this document application profile 89
B.3.1 Delimeter elements 89
B.3.2 Metafile descriptor elements 90
B.3.3 Picture descriptor elements 90
B.3.4 Control elements 91
B.3.5 Graphical primitive elements 91
B.3.6 Attribute elements 91
B.3.7 External elements 93

Annex C – Bibliography 94

SUMMARY

This Recommendation specifies an ODA Document Application Profile (DAP) named PM-26.

This Profile specifies an interchange format for the transfer of structured documents between equipments designed for word or document processing. Such documents may contain character, raster graphics and geometric graphics content.

The documents that can be interchanged using this Profile range from simple documents to highly structured technical reports, articles and typeset documents such as brochures. This Profile provides a comprehensive level of features for the transfer of documents between these systems.

A document structured in accordance with this Profile is represented for interchange by the Open Document Interchange Format (ODIF), as defined in Recommendation T.415.

INTRODUCTION

This Recommendation specifies an ODA Document Application Profile (DAP) named PM-26. The purpose of this Profile is to facilitate the interworking of applications interchanging documents based on ODA, T.410 Series of CCITT Rec. | ISO/IEC 8613. This Profile is suitable for interchanging documents in formatted form, processable form or formatted processable form and has been defined in accordance with CCITT Rec. T.411 | ISO/IEC 8613-1. The format of this Profile is in accordance with the standardized proforma and notation defined in Annex F of CCITT Rec. T.411 | ISO/IEC 8613-1.

DOCUMENT APPLICATION PROFILE PM-26 FOR THE INTERCHANGE OF ENHANCED STRUCTURE, MIXED CONTENT DOCUMENTS IN PROCESSABLE AND FORMATTED FORMS[1)]

(Geneva, 1991; amended in Geneva, 1994)

1 Scope

This Profile specifies an interchange format for the transfer of structured documents between equipments designed for word or document processing. Such documents may contain character, raster graphics and geometric graphics content.

The documents that can be interchanged using this Profile range from simple documents to highly structured technical reports, articles and typeset documents such as brochures. This Profile provides a comprehensive level of features for the transfer of documents between these systems.

This Profile allows documents to be interchanged in the following forms:

- formatted form;
- processable form;
- formatted processable form.

The architecture levels defined for these three forms have matching functionalities so that the interchange formats of a document are convertible from a processable form to any other form.

This Profile is independent of the processes carried out in an end system to create, edit or reproduce documents. It is also independent of the means to transfer documents which, for example, may be by means of communication links or storage media.

A document structured in accordance with this Profile is represented for interchange by the Open Document Interchange Format (ODIF), as defined in CCITT Rec. T.415 | ISO/IEC 8613-5.

2 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. All Recommendations and other references are subject to revision; as a general rule, all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. However, users are warned against automatically applying any more recent editions of the Recommendations and other references listed below, since the nature of the references made by this Recommendation to such documents is that it may be specific to a particular edition. Members of IEC and ISO maintain registers of currently valid International Standards and ISPs, and ITU-T maintains published editions of its current Recommendations.

2.1 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation T.411 (1988), *Open Document Architecture (ODA) and interchange format – Introduction and general principles*.

  ISO 8613-1:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 1: Introduction and general principles.*

- CCITT Recommendation T.411 Annex F (1991), *Open Document Architecture (ODA) and interchange format: Introduction and general principles – Annex F: Document Application Profile Proforma and Notation*.

  ISO 8613-1 Add.1: *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 1: Introduction and general principles – Addendum 1: Document Application Profile Proforma and Notation.*

---

1) Recommendation T.505 is technically alyned with ISO/IEC ISP 11181-1:1993.

– CCITT Recommendation T.412 (1988), *Open Document Architecture (ODA) and interchange format – Document structures*.

ISO 8613-2:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 2: Document structures.*

– CCITT Recommendation T.414 (1988), *Open Document Architecture (ODA) and interchange format – Document profile.*

ISO 8613-4:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 4: Document profile.*

– CCITT Recommendation T.415 (1988), *Open Document Architecture (ODA) and interchange format – Open Document Interchange Format (ODIF)*.

ISO 8613-5:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 5: Office Document Interchange Format (ODIF).*

– CCITT Recommendation T.416 (1988), *Open Document Architecture (ODA) and interchange format – Character content architectures.*

ISO 8613-6:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 6: Character content architecture.*

– CCITT Recommendation T.417 (1988), *Open Document Architecture (ODA) and interchange format – Raster graphics content architectures*.

ISO 8613-7:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 7: Raster graphics content architecture.*

– CCITT Recommendation T.418 (1988), *Open Document Architecture (ODA) and interchange format – Geometric graphics content architectures*.

ISO 8613-8:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 8: Geometric graphics content architecture.*

– 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)*.

– ITU-T Recommendation T.502 (1994), *Document application profile PM-11 for the interchange of simple structure, character content documents in processable and formatted forms*.

ISO/IEC ISP 10610-1:1993, *Information technology – International Standardized Profile FOD11 – Open document format: Simple document structure – Character content architecture only – Part 1: Document Application Profile (DAP).*

– ITU-T Recommendation T.506 (1993), *Document application profile PM-36 for the interchange of extended document structures and mixed content documents in processable and formatted forms*.

ISO/IEC ISP 11182-1:1993, *Information technology – International Standardized Profile FOD36 – Open document format: Extended document structure – Character, raster graphics and geometric graphics content architectures – Part 1: Document Application Profile (DAP).*

– CCITT Recommendation T.50 (1992), *International Reference Alphabet (IRA) – Information technology – 7-bit coded character set for information interchange*.

ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange.*

– CCITT Recommendation T.51 (1992), *Latin based coded character sets for telematic services.*

ISO 6937:1994, *Information technology – Coded graphic character set for the communication of texts using the Latin alphabet*.

2.2 Additional references

– CCITT Recommendation T.4 (1988), *Standardization of group 3 facsimile apparatus for document transmission*.

- CCITT Recommendation T.6 (1988), *Facsimile coding schemes and coding control functions for group 4 facsimile apparatus*.
- ISO 2022:1986, *Information processing – ISO 7-bit and 8-bit coded character sets – Code extension techniques.*
- ISO 2375:1985, *Data processing – Procedure for registration of escape sequences.*
- ISO/IEC 7350:1991, *Information technology – Registration of repertoires of graphic characters from ISO/IEC 10367.*
- ISO/IEC 8632:1992, *Information technology – Computer graphics – Metafile for the storage and transfer of picture description information*
  - *Part 1: Functional specification*.
  - *Part 3: Binary encoding.*
- ISO 8859-1:1987, *Information processing – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No 1*.
- ISO 9293:1987, *Information processing – Volume and file structure of flexible disk cartridges for information interchange.*
- ISO/IEC 9541:1991, *Information technology – Font information interchange.*
- ISO/IEC TR 10000-1:1992, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 1: Framework.*
- ISO/IEC TR 10000-2:1992, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 2 Taxonomy of OSI profiles.*
- CCITT Recommendation T.400 (1988), *Introduction to document architecture, transfer and manipulation.*
- ISO/IEC ISP 11181-1:1993, *Information technology – International Standardized Profile FOD26 – Open Document Format : Enhanced document structure – Character, raster graphics and geometric graphics content architectures – Part 1: Document Application Profile (DAP).*

3 Definitions

For the purposes of this Recommendation, the following definitions apply. The definitions given in CCITT Rec. T.411 | ISO/IEC 8613-1 are applicable to this Profile.

constituent contraint names: Each constituent that may be included in a document that conforms to this Profile has been given a unique name which serves to associate that constituent with a constituent constraint defined in this Recommendation.

The convention is that full names are used (i.e., no abbreviations are used), two or more words in a name are concatenated and each word begins with a capital. Examples of constituent constraint names used in this Profile are BodyText, Footnote, RectoPage and ColumnFixed.

In clause 6 each constituent constraint provided by this Profile is italicized once at the point in the text at which the purpose of that constituent constraint is defined. This also serves to identify all the constituent constraints provided by this Profile.

The same constituent constraint names are also used in the technical specification in clause 7 so that there is a one-to-one correspondence between the use of these names in clauses 6 and 7.

Although the constituent constraint names relate to the purpose of the constituent constraints, the semantics of constituents shall not be implied from the names that are used. Also, these names do not appear in an interchanged document but a mechanism for associating constituents in a document with constituent constraints is provided (see 6.6.4). Thus in an application using this Profile, the constituents may be known to the user by different names.

4 Relationship with other Profiles

This Profile belongs to a series of hierarchically related Profiles which include PM-11 and PM-36.

The features supported by this Profile are a superset of the features supported by the Profile PM-11 and thus all data streams that are conformant to PM-11 are also conformant to this Profile, apart from the document application profile identifier.

Also the features supported by this Profile are a subset of the features supported by the Profile PM-36 and thus all data streams conformant to this Profile are also conformant to PM-36, apart from the document application profile identifier.

NOTE – This Profile is technically aligned with (but not identical to) the Specification defined in the ISO International Standardized Profile FOD26, except that FOD26 not only defines the use of the ODIF interchange format, but also the use of the SDIF interchange format.

5 Conformance

In order to conform to this Profile, a data stream representing a document shall meet the requirements specified in 5.1.

This Recommendation does not define implementation or service requirements.

5.1 Data stream conformance

The following requirements apply to the encoding of data streams which conform to this Profile:

- the data stream shall be encoded in accordance with the ASN.1 encoding rules defined in CCITT Rec. X.209 | ISO/IEC 8825;
- the data stream shall be structured in accordance with the interchange format defined in clause 8;
- the document, as represented by the data stream after resolution of any external references, shall be structured in accordance with one of the documents architecture classes as defined in 6.1 and shall contain all mandatory constituents specified for that class; other constituents may be included, provided that they are permitted for that class, as specified in clause 7;
- each constituent shall contain all those attributes specified as required for that constituent in this Profile; other attributes may be specified provided that they are permitted for that constituent;
- the attribute values specified shall be within the range of permissible values specified in this Profile;
- the encoded document shall be constructed in accordance with the abstract document architecture defined in CCITT Rec. T.412 | ISO/IEC 8613-2;
- the document shall be structured in accordance with the characteristics and constraints specified in clause 6.

5.2 Implementation conformance

This subclause states the requirements for implementations claiming conformance to this Profile.

A conforming receiving implementation shall be capable of receiving any data streams conforming to this Profile structured in accordance with ODIF. Receiving usually, but not always, involves recognizing and further processing the data stream elements.

A receiving system which claims conformance to this document application profile shall be capable of handling data streams which are conformant to document application profiles that are subordinate to this document application profile within the taxonomy described in clause 4, (i.e. PM-11).

6 Characteristics supported by this document application profile

This clause describes the characteristics of documents which may be represented by data streams conforming to this Profile. This clause also describes how these characteristics are represented in terms of constituent constraints.

6.1 Overview

6.1.1 General

This Profile supports the interchange of documents in the following forms:

- processable form, which facilitates the revision of a document by a recipient;
- formatted form, which facilitates the reproduction of a document as intended by the originator;
- formatted processable form, which facilitates the reproduction of a document as intended by the originator or facilitates the revision of a document by a recipient.

In addition, this Profile supports the interchange of:

– generic-documents;
– document profile.

The constituents that make up these five classes of data stream are defined in 6.1.2 to 6.1.6. Constituents defined as *required* shall occur in any data stream that conforms to this Profile. Constituents listed as *optional* may or may not be present in the data stream depending on the requirements of the particular data stream.

The constituents that make up a complete document that is conformant to this Profile include all those referenced and contained in resource-documents and external-documents, if any (see 6.6.1 and 6.6.2).

6.1.2 Formatted form documents

Required constituents:

– a document profile;
– layout object descriptions representing a specific layout structure.

Optional constituents:

– layout object class descriptions representing a factor generic layout structure;
– presentation styles;
– content portion descriptions.

6.1.3 Processable form documents

Required constituents:

– a document profile;
– logical object class descriptions representing a complete or partial generic logical structure;
– logical object descriptions representing a specific logical structure.

Optional constituents:

– layout object class descriptions representing a complete generic layout structure;
– layout styles;
– presentation styles;
– content portion descriptions.

In the case of processable form documents, when the generic layout structure is not present, additional restrictions are placed on the layout directives that may be included in layout styles. These restrictions are defined in 6.4.3.

6.1.4 Formatted processable form documents

Required constituents:

– a document profile;
– logical object class descriptions representing a complete or partial generic logical structure;
– logical object descriptions representing a specific logical structure;
– layout object class descriptions representing a complete generic layout structure;
– layout object descriptions representing a specific layout structure.

Optional constituents:

– layout styles;
– presentation styles;
– content portion descriptions.

6.1.5 Generic-documents

A generic-document consists of one of the following sets of constituents:

a) – a document profile;
– logical object class descriptions which represent a complete or partial generic logical structure;
– layout styles whose presence is optional;
– presentation styles whose presence is optional;

– generic content portions whose presence is optional.

b) – a document profile;
– layout object class descriptions which represent a complete generic layout structure or factor set;
– presentation styles whose presence is optional;
– generic content portions whose presence is optional.

c) – a document profile;
– logical object class descriptions which represent a complete or partial generic logical structure;
– layout object class descriptions which represent a complete generic layout structure;
– layout styles whose presence is optional;
– presentation styles whose presence is optional;
– generic content portions whose presence is optional.

6.1.6 Document profile

This form of document contains a document profile only.

6.2 Logical characteristics

6.2.1 Introduction

This subclause defines the constituent constraints for logical components which are provided by this Profile to represent the characteristics of documents containing logical component descriptions.

Different constituent constraints may be used to represent and distinguish parts of a document that have different logical characteristics. This subclause describes the general characteristics and typical uses of the constituent constraints that are provided.

The descriptions of the logical characteristics represented by each of the constituent constraints is provided for guidance only. It is the responsibility of the user to determine how a document is to be represented using the constituents provided. Adherence to these guidelines can enhance the mutual understanding of a document by an originator and a recipient.

6.2.2 Overview of the logical structure

From the logical point of view, the document consists of two parts, namely a body part and a common part.

The body part represents the main content of a document and is intended to be reproduced in the body area of the pages that make up the document.

The common part represents common content that is to be placed in reserved header and footer areas on each page of a document. Header and footer content are independently optional and so may be included in an interchanged document only if required.

6.2.3 Body part of the logical structure

6.2.3.1 DocumentLogicalRoot

DocumentLogicalRoot is a constituent constraint representing the top level in the document logical structure. Its immediate subordinates consist of a sequence of one or more constituent constraints of the type Passage.

The automatic numbering schemes that apply to constituent constraints of the types NumberedSegment and Footnote may be initialised on the DocumentLogicalRoot.

6.2.3.2 Passage

Passage is a constituent constraint that represents the first level of logical subdivision of a document. It may be used to indicate a logical grouping of subordinate parts of a document that are to be regarded as an entity for reading or that have common layout and presentation characteristics.

Passages are typically used to represent:

– the contents to be placed on the title page of a report;
– the front matter in the table of contents or foreword;
– the main matter of the document;

– the back matter, consisting of appendices, glossary or index.

The automatic numbering schemes that apply to subordinate constituent constraints of the types NumberedSegment and Footnote may be initialised on a Passage.

The immediate subordinates of a Passage consist of an optional arbitrary ordered sequence of one or more of the following constituent constraint types:

– Paragraph;
– BodyGeometric;
– BodyRaster;
– BodyText.

These may be optionally followed by one or more constituent constraints of the type NumberedSegment.

A Passage shall have at least one of the above constituent constraint types as a subordinate.

A document may contain several different class definitions of the type Passage, each of which defines the common characteristics of sets of Passages within the document such as their allowed subordinates or layout properties. For example, a class of Passages may be defined which always begin on a new page set.

6.2.3.3 NumberedSegment

NumberedSegment is a constituent constraint that represents a logical subdivision within a document that is distinguished by an identifier. The logical subdivision may be a subdivision of a Passage or another higher level NumberedSegment. The logical subdivision may also have some common layout characteristics.

The automatic numbering schemes that apply to subordinate constituent constraints of the types NumberedSegment and Footnote may be initialised on a Passage.

The immediate subordinates of a NumberedSegment consist of the constituent constraint Number, whose presence is mandatory and serves to carry the identifier of the NumberedSegment. This is followed by an optional arbitrary ordered sequence of one or more of the following constituent constraints:

– Paragraph;
– BodyGeometric;
– BodyRaster;
– BodyText.

These are optionally followed by a sequence of one or more constituent constraints of the type NumberedSegment. Hence a document may contain any number of nested levels of the NumberedSegment constituents.

A NumberedSegment is typically used to represent entities such as chapters, sections, nested sub-sections and appendices which contain an identifier that serves to distinguish that entity for human comprehension.

A document may contain any number of different class definitions of NumberedSegment which define the common characteristics of sets of NumberedSegments, such as their allowed subordinates and layout properties.

Class definitions of NumberedSegments may be recursively defined. In this case only one call of NumberedSegment may be specified, and the <simple-expr> construction in the macro USENUMBERSTRINGS in the bindings attribute of this class shall use the optional ORD construction only. If the recursive class definitions are used for NumberedSegment, the following constraints shall also apply. For all levels which reference recursively defined classes:

– numbering format shall be the same;
– no initial value other than 1 or re-initialisation of the numbering is possible;
– it is not possible to continue the numbering across Passages.

If class definitions are not recursive in this way, there shall be one and only one class definition for NumberedSegements corresponding to each level of numbering within each Passage. Class definitions may be shared between NumberedSegments belonging to different Passages but they shall then be used at the same level.

6.2.3.4 Number

Number is a constituent constraint that represents the identifier of a NumberedSegment to which it is subordinate. This identifier allows the NumberedSegment to be distinguished within the document for machine processing or human comprehension.

A Number defines a basic logical constituent which contains a content generator which, when evaluated, produces the identifier referred to above. This evaluation takes place during the layout process.

The identifiers are structured and consist of a sequence of one or more numerals that allow NumberedSegments at the same or different levels in a document structure to be uniquely distinguished. The numerals may be represented by Arabic or Roman numerals or by their alphabetic equivalent in lower or upper case characters (the number 1 is represented by "A", etc.). Each numeral in an identifier is distinguished by means of separator characters such as spaces and full stops (the character "period"); a typical example is "6.2.3.4".

NOTE – The separator may be an empty string.

Further details of the structure and generation of the identifiers are given in 6.6.6.2.

6.2.3.5 Paragraph

Paragraph is a constituent constraint that is a subdivision of a Passage or NumberedSegment. It is typically used to represent the grouping of parts of a document that deals with a single theme or topic. These parts may consist of character, raster graphics and geometric graphics content.

The immediate subordinates of a Paragraph consist of an arbitrary ordered sequence of one or more of the following constituent constraints:

- BodyText;
- BodyRaster;
- BodyGeometric;
- Footnote.

Content from any subordinate basic text objects within a paragraph may be run on one from another (that is, to continue on the same line) by use of the Concatenation feature (see 6.4.2.5). Alternatively, content from subordinates of a paragraph may be separated one from another to give white space between them, using Separation (see 6.4.2.2). This may be used to give an effect similar to that achieved with empty lines of text. Use of empty text lines to achieve white space between areas of text or other content may lead to unintended blank areas adjacent to the leading edge of layout objects (e.g., at page breaks) whereas the use of Separation avoids this.

Constituents of the type BodyText may be concatenated to form a continuous stream of character content which is laid out as a single unit. Sequences of constituents of the types BodyText and Footnote may be concatenated to represent a stream of character content with embedded footnotes. Multiple embedded footnotes, which may be consecutive without intervening text, may be included in the content.

Another typical use of a Paragraph is to represent a group of document parts that have common layout characteristics. An example is a graphical illustration with associated text which is to be laid out in a particular frame.

6.2.3.6 BodyText, BodyRaster and BodyGeometric

BodyText, *BodyRaster* and *BodyGeometric* are constituent constraints which represent the lowest level of logical subdivision of a document. These constituent constraints are subdivisions of Passages, NumberedSegments and Paragraphs. They allow the layout and presentation requirements of different parts of a document to be specified.

These are basic logical constituents that directly refer to content portions that contain character, raster graphics and geometric graphics content respectively. BodyText in the specific logical structure shall refer to one or more content portions each containing processable, formatted or formatted processable character content. BodyRaster and BodyGeometric shall only refer to a single content portion containing formatted processable raster graphics content or formatted processable geometric graphics content respectively.

Constituents of these types in the generic logical structure may refer to generic content. This provides the means of defining common content within the body part of a document.

6.2.3.7 Footnote

Footnote is a constituent constraint that is a subdivision of a Paragraph and is used to represent footnotes within a document.

A footnote is an amount of content that is logically associated with a particular part of the document body but which is intended to be read and laid out separately from its associated part of the document. Typically, a footnote consists of a footnote identifier, which is embedded within the document body, and the footnote itself, which is laid out elsewhere.

A Footnote defines a composite logical constituent whose immediate subordinates consist of the constituent constraint FootnoteReference, which represents the footnote identifier, followed by the constituent constraint FootnoteBody, which represents the footnote itself. Both of these subordinates are mandatory.

6.2.3.8 FootnoteReference

FootnoteReference defines a constituent constraint that is used to represent a footnote reference within the body of a document.

FootnoteReference defines a basic logical constituent that contains a content generator which when evaluated produces a character string which constitutes the footnote reference referred to above.

The generated character string consists of a label with optional prefix and suffix character strings. The label is used to uniquely identify a particular footnote and may consist of a number which is represented in the form of Arabic or Roman numerals or by an alphabetic equivalent. The number may be automatically generated so that its value is incremented for each successive footnote. Alternatively, the label may consist of a user defined character string.

In a sequence of footnotes, automatic and user defined labels may be freely mixed (giving for example the sequence 1, 2*, 3, 4). If the label consists of a user-defined character string, the automatically generated number sequence is not incremented.

An example of a footnote reference is '(2)' where '(' and ')' are user defined prefix and suffix strings respectively and '2' is the automatically generated label. Another example is 'note^{5}' where '5' is the label and 'note' is a prefix string which also contains the control function PLU to enable the label to be represented in the form of a superscript. In this case, a suffix string containing the control function PLD would be required to cause the superscripting to be cancelled before the following text.

The format of the content generator referred to above is described in 6.6.6.3.

6.2.3.9 FootnoteBody

FootnoteBody is a constituent constraint which represents the content of a footnote.

FootnoteBody defines a composite logical constituent whose subordinates consist of a FootnoteNumber constituent, which is mandatory and represents the footnote identifier, followed by a sequence of one or more FootnoteText constituents which represents the footnote content. The identifier referred to above is identical to the corresponding footnote identifier which is embedded in the content of the document body and represented by the constituent constraint FootnoteReference.

The constituents subordinate to FootnoteBody are intended to be laid out separately from the other parts of the document content. When a generic layout structure is specified for the document, these constituents are constrained to be laid out in a FootnoteArea frame (see 6.3.5.9).

6.2.3.10 FootnoteNumber

FootnoteNumber is a constituent constraint that represents the footnote identifier within the footnote body.

This identifier is identical to the content associated with the constituent constraint FootnoteReference but is intended to be laid out so that it immediately precedes the content of the footnote body.

FootnoteNumber defines a basic logical constituent that contains a content generator which when evaluated produces the identifier referenced above. The format of this content generator is the same as the content generator that may be specified for the constituent constraint FootnoteReference.

It is required to specify the attribute "layout category" with the value 'Footnote' for this constituent. This along with the attribute "permitted categories" with the value 'Footnote' on the footnote frame will ensure that this constituent is laid out in a FootnoteArea frame when generic layout structure is specified within the document.

6.2.3.11 FootnoteText

FootnoteText is a constituent constraint that is used to represent the footnote content. It is the lowest logical subdivision of a FootnoteBody.

FootnoteText defines a basic logical constituent that references one or more content portions each containing processable, formatted or formatted processable character content.

It is required to specify the attribute "layout category" with the value 'Footnote' for this constituent. This along with the attribute "permitted categories" with the value 'Footnote' on the footnote frame will ensure that this constituent is laid out in a FootnoteArea frame when generic layout structure is specified within the document.

6.2.4 Common content part of the logical structure

6.2.4.1 CommonContent

CommonContent is a constituent constraint that represents common content that is to be laid out in the header and footer areas of the pages of a document. Common content consists of any combination of character, raster graphics and geometric graphics content.

Any number of constituent constraints of the type CommonContent may be contained in a document. CommonContent is a composite logical object class whose immediate subordinates consist of an arbitrary ordered sequence of one or more of the following constituent constraints:

- CommonText;
- PageNumber;
- CommonRaster;
- CommonGeometric.

When the generic layout structure is present, constituents of the type CommonContent and their associated subordinate constituents are constrained to be laid out in frames representing header or footer areas using the logical source mechanism (see 6.3.6).

6.2.4.2 CommonText

CommonText is a constituent constraint that represents the common character content that is to be laid out in the header or footer area of a document. For example, header or footer content that appears on each page in a sequence of pages may be represented by this constituent constraint.

CommonText is a constituent constraint for a basic logical object class that references one or more content portions each containing character content in a processable, formatted or formatted processable form.

6.2.4.3 PageNumber

PageNumber is a constituent constraint that represents the common character content that is to be laid out in the header or footer area of a document. This constituent constraint is specifically used when it is required to present a header or footer content which contains an automatically generated page number.

PageNumber is a basic logical object class that contains a content generator. This content generator contains a reference to a page number which is automatically evaluated when the document is laid out. This provides the means of representing the page numbers that are displayed on the consecutive pages of a document.

Each page number consists of a single number which may be represented in the form of Arabic or Roman numerals or in its alphabetic equivalent. Page numbering schemes may start at 0 or any value greater than 0, at the document root or pageset levels.

The format of the content generators is defined in 6.6.6.1.

6.2.4.4 CommonRaster

CommonRaster is a constituent constraint that represents the common raster graphics content that is to be laid out in the header or footer area of a document. For example, this constraint may be used to represent a logo which is to be laid out on each page of a document.

CommonRaster is a basic logical object class which references a single content portion containing formatted processable raster graphics content.

6.2.4.5 CommonGeometric

CommonGeometric is a constituent constraint that represents the common geometric graphics content that is to be laid out in the header or footer area of a document. For example, this constraint may be used to represent a graphical icon which is to be laid out on each page of a document.

CommonGeometric is a constituent constraint for a basic logical object class which refers to a single content portion containing formatted processable geometric graphics content.

6.3 Layout characteristics

This subclause defines the constituent constraints for layout components which are provided by this Profile to represent the characteristics of documents.

Different constituent constraints may be used to represent and distinguish parts of a document that have different layout characteristics. This subclause describes the general characteristics and typical uses of the constituent constraints that are provided.

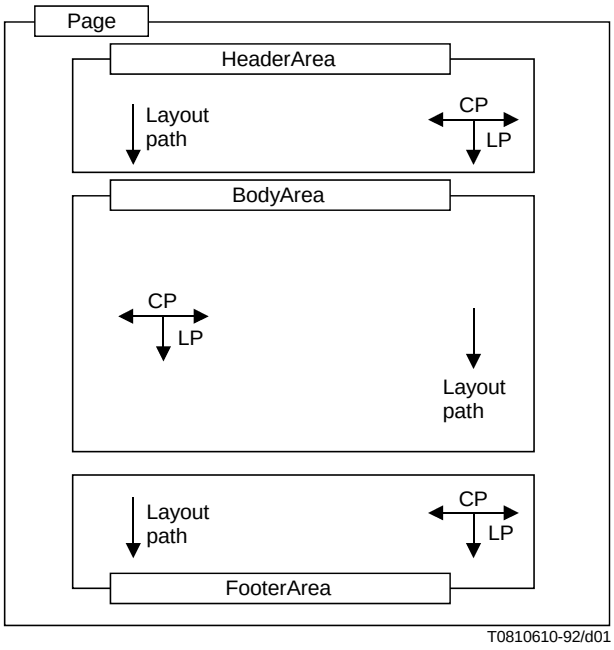
The descriptions of the layout characteristics represented by each of the constituent constraints is provided for guidance only. It is the responsibility of the user to determine how a document is to be represented using the constituents provided. Adherence to these guidelines can enhance the mutual understanding of a document by an originator and a recipient.

6.3.1 Overview of the layout characteristics

The document structure allows the document content to be laid out and presented in one or more page sets. Each page set may be used for different parts of the document, for example, the title page, foreword, table of contents, document body and appendices.

Each page set consists of a series of pages. In general, each page may be sub-divided into three areas: the body area, which is used to layout the document body; and the header and footer areas, which may be used to layout the common content.

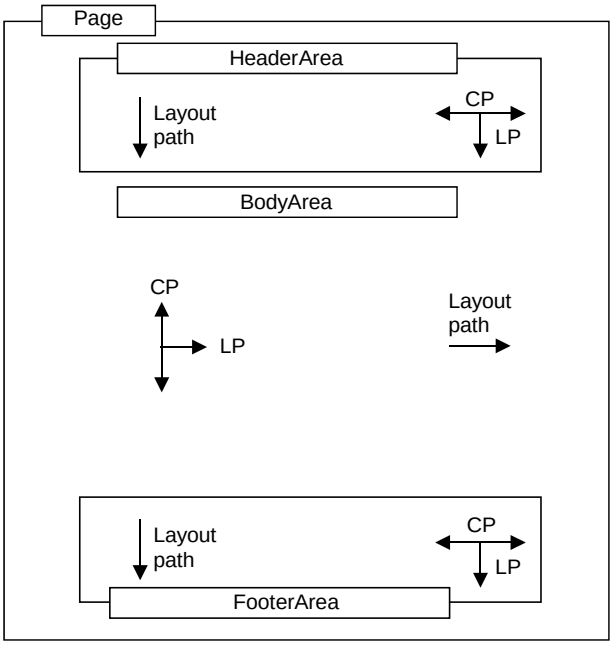
Four page layout types are provided by this Profile. Each page layout type specifies how the body, header and footer areas are positioned within each page and how the content may be presented within each of those areas. These four types are referred to as page layouts A, B, C and D and are illustrated in Figures 1, 2, 3 and 4 respectively.



CP Character Path
LP Line Progression

FIGURE 1/T.505

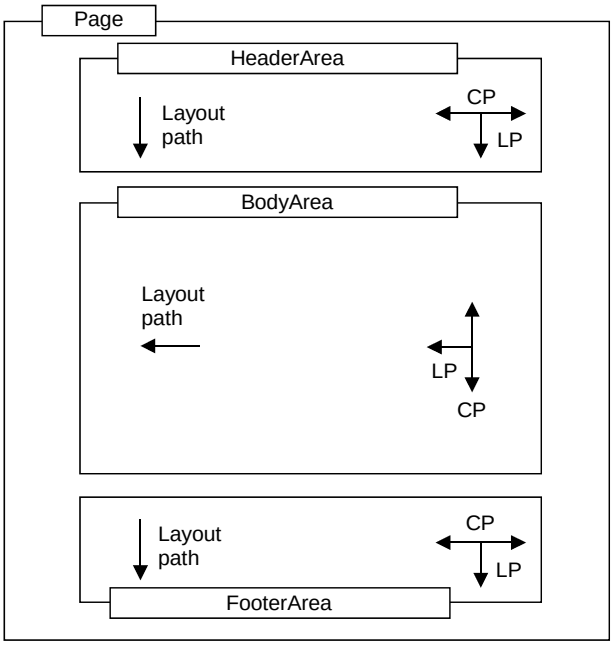
Page layout type A



CP Character path
LP Line progression

FIGURE 2/T.505

Page layout type B



CP Character path
LP Line progression

FIGURE 3/T.505

Page layout type C

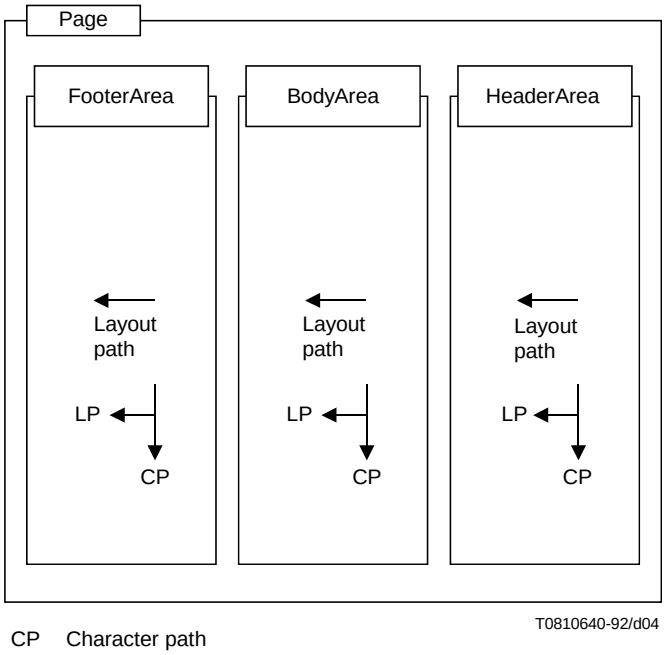


FIGURE 4/T.505

Page layout type D

It is intended that all applications which use this Profile shall support page layout A, whereas support for the other three page layouts may be specified as optional.

Page layout A is used when the character content is to be laid out horizontally (from left to right or from right to left) and from top to bottom within the body area. This layout is typically used for documents written in Latin based, Hebrew and Arabic languages.

Page layout B is used when the character content is to be laid out vertically bottom to top or top to bottom and from left to right within the body area. This layout is typically used for documents written in Latin based, Hebrew and Arabic languages in which it is required to layout the content in landscape orientation within the body area of the page.

Page layouts C and D are used when the character content is to be laid out from top to bottom and from right to left within the body area. These layouts are typically used in documents written in languages which use ideograms, such as Japanese and Chinese characters.

The body area may be further sub-divided into areas composed of single and multiple columns and an area may be reserved for footnotes. In addition, the header and footer areas may be sub-divided to allow the representation of different content types.

6.3.2 DocumentLayoutRoot

DocumentLayoutRoot is a constituent constraint that represents the top level in the document layout structure. Its immediate subordinates consist of a sequence of one or more constituents of the type PageSet. The numbering schemes for pages may be initialised on this constituent constraint.

6.3.3 PageSet

PageSet is a constituent constraint that represents a grouping of pages within a document. A PageSet is typically used to represent a part of a document that has different layout requirements from other parts of a document. Also, a PageSet may correspond to a part of a document that has a certain logical significance, for example, a PageSet might represent the front matter in a document or an individual chapter.

Only one level of PageSet is allowed in a document. However, a document may contain any number of class definitions of the type PageSet which may be used, for example, to provide a choice of alternative layouts for different parts of a document or to specify the exact layout requirements for each successive part of a document.

The immediate subordinates of a PageSet consist of a combination of constituent constraints of the types Page, RectoPage and VersoPage, as described in 6.3.4.1.

6.3.4 Page characteristics

6.3.4.1 Page constituents

Three constituent constraints are provided to represent the pages within a document, namely *Page*, *RectoPage* and *VersoPage*.

The only difference in the characteristics of these types of constituent constraints concerns the values that may be specified for the parameter "side of sheet" in the attribute "medium type". In the case of Page, this parameter may be specified with the value 'recto', 'verso' or 'unspecified'. In the case of RectoPage, this parameter may be specified with the value 'recto' or 'unspecified'; in the case of VersoPage, this parameter may be specified with the value 'verso' or 'unspecified'. The values 'recto' and 'verso' of the parameter "side of sheet" of the attribute "medium type" are non-basic.

The pages that make up a page set consist of an optional initial page which is represented by the constituent constraint Page and which is optionally followed by either:

- A sequence of pages represented by the constituent constraint Page. All pages in this sequence shall have the same layout characteristics but these characteristics may differ from those of the initial page.
- A sequence of pages which are intended to be laid out alternatively on the recto and verso (or on the verso and recto) sides of the presentation medium and are represented by the constituent constraints RectoPage and VersoPage respectively. All pages in this sequence shall have the same layout characteristics but these characteristics may differ from those of the initial page.

Pages having the same layout characteristics are pages that have the same page layout (see 6.3.4.5) and for which the body area, header area (if present) and footer area (if present) have the same dimensions and positions within the page (see 6.3.4.3). Pages having the same layout characteristics do not necessarily have the same position on the presentation medium (see 6.3.4.4).

A page set shall contain at least one page.

An initial page is typically used at the beginning of a document or of a section within a document. It may be used, for example, for a title page whose layout requirements differ from the following pages.

The following restrictions also apply to the pages within a page set:

- all the pages shall have the same dimensions and orientation (see 6.3.4.2);
- all pages shall be laid out on the same size of presentation medium (see 6.3.4.3).

6.3.4.2 Page dimensions

The dimensions of the pages may be specified as any value (in BMUs) that is equivalent to or less than ISO A3 or ANSI B paper sizes in portrait or landscape orientation. The dimensions may be specified in portrait or landscape orientation. Japanese page sizes B4 and B5 are also supported but the dimensions of these pages lie within the range of dimensions given above.

Dimensions equivalent to or less than the common assured reproduction area of ISO A4 and ANSI A in portrait or landscape orientation are basic values. Larger page sizes are non-basic and their use shall be indicated in the document profile.

Any default page dimensions may be specified in the document profile subject to the maximum dimensions defined above.

NOTE – The size termed *North American Letter (NAL)* in CCITT Rec. T.410 series | ISO/IEC 8613, clause 7 is called *ANSI A* in this Recommendation to be consistent with the other references to ANSI standard paper sizes.

6.3.4.3 Nominal page sizes

The nominal page sizes that may be specified are listed in Table 1. These may be specified in portrait or landscape orientation. All values of nominal page size are non-basic and hence all values used in a document shall be indicated in the document profile.

Any value of nominal page size defined in Table 1, subject to the restrictions specified above, may be specified as the default value in the document profile.

Table 1 also includes the recommended Assured Reproduction Area (ARA). Information loss may occur when a document is reproduced if the dimensions of constituent constraints of the type page exceed the ARA for the specified nominal page size.

TABLE 1/T.505

Nominal page sizes

Page type	Size in inches or millimetres	Size in BMUs	ARA in BMUs
ISO A5	148 mm × 210 mm	7 015 × 9 920	Not defined
ISO A4	210 mm × 297 mm	9 920 × 14 030	9 240 × 13 200
ISO A3	297 mm × 420 mm	14 030 × 19 840	13 200 × 18 480
ANSI legal	8.5 in. × 14 in.	10 200 × 16 800	9 240 × 15 480
ANSI A	8.5 in. × 11 in.	10 200 × 13 200	9 240 × 12 400
ANSI B	11 in. × 17 in.	13 200 × 20 400	12 744 × 19 656
Japan-legal	257 mm × 364 mm	12 141 × 17 196	11 200 × 15 300
Japan-letter	182 mm × 257 mm	8 598 × 12 141	7 600 × 10 200

6.3.4.4 Page offset

The page offset is the distance of the position of the left and top edges of the page relative to the left and top edges respectively of the presentation medium on which each page is reproduced. Any value of page offset may be specified provided that no part of the page area lies outside the area of the nominal page. Also, page offsets specified for the initial, recto and verso pages within a given page set may differ. The default page offset may be specified in the document profile.

6.3.4.5 Page layout characteristics

6.3.4.5.1 General characteristics

Each page in a document may be subdivided into three rectangular areas, as follows:

- a body area which is reserved for content that belongs to the body part of the document (see 6.3.5);
- a header area which is reserved for common header content (see 6.3.6);
- a footer area which is reserved for common footer content (see 6.3.6).

The body area is mandatory and shall occur on every page in a document. The header and footer areas are both optional.

Also, these three areas shall be entirely contained within the page area and shall not overlap.

Four types of page layout are defined:

6.3.4.5.2 Page layout A

For page layout A the header and footer areas are placed above and below the body area respectively. The layout paths in the header, body and footer areas are specified as 270°. This type of layout is illustrated in Figure 1.

6.3.4.5.3 Page layout B

For page layout B the header and footer areas are placed above and below the body area respectively. The layout path in the body area is specified as 0°; in the header and footer areas the layout paths are specified as 270°. This type of layout is illustrated in Figure 2.

6.3.4.5.4 Page layout C

For page layout C the header and footer areas are placed above and below the body area respectively. The layout path in the body area is specified as 180°; in the header and footer areas, the layout paths are specified as 270°. This type of layout is illustrated in Figure 3.

6.3.4.5.5 Page layout D

For page layout D the header and footer areas are placed to the right and left of the body area respectively. The layout paths in the header, body and footer areas are all specified as 180°. This type of layout is illustrated in Figure 4.

6.3.5 Body area characteristics

6.3.5.1 General characteristics

The body area is the area within a page where the main matter of the document, that is the *body* part of the document, is laid out.

The body area may consist of a single frame into which the content is directly laid out. In this case, the body area is represented by a BasicBody frame.

Alternatively, the body area may be subdivided into different rectangular areas to provide for different layout requirements. In this case, the body frame is represented by a VariableCompositeBody frame.

6.3.5.2 BasicBody

BasicBody is a constituent constraint which defines a lowest level frame into which content is directly laid out.

The position and dimensions of this frame are fixed. The layout path specified depends upon the page layout type being used (see 6.3.4.5).

6.3.5.3 VariableCompositeBody

VariableCompositeBody is a constituent constraint that defines a composite frame which contains one or more subordinate variably positioned frames. These subordinates are not pre-determined and are automatically adjusted during the layout process to accommodate the content that is allocated to them. For example, to provide for combinations of single or multiple column layout and the layout of footnotes. A VariableCompositeBody frame has a fixed position and fixed dimensions. The layout path specified for this frame depends upon the page layout type being used (see 6.3.4.5).

The immediate subordinates of VariableCompositeBody frames consist of an arbitrary ordered sequence of one or more frames of the following constituent constraints:

- BasicFloat;
- SnakingColumns;
- SynchronizedColumns.

It may also contain a single frame of the type FootnoteArea.

The subordinate frames are all variably positioned and may have variable dimensions.

Thus the relative positions of these frames in the body area may vary and depend upon the positions of other frames (if any) that are placed in the same VariableCompositeBody frame.

The layout path for VariableCompositeBody frames may be specified as 270°, 0° or 180°. This determines the page layout type used in the case where VariableCompositeBody represents the entire body area.

Frames of the type BasicFloat, SnakingColumns and SynchronizedColumns are laid out along the layout path specified (in normal positioning fill order). FootnoteArea frames are laid out in the same direction as the body area layout path, but reverse fill order is used.

These frames are constrained to have the same layout path as the VariableCompositeBody frame to which they are subordinate. An exception to this rule is made for frames of the type SnakingColumns (see subclause below).

Figures 5, 6 and 7 provide illustrations of the layout of frames within a VariableCompositeBody frame for the various page layout types.

A choice of subordinate frames of the types listed above may be specified for a VariableCompositeBody frame. Different frame types may be selected using various layout directives (see 6.4) and hence the layout characteristics of the body areas within a page set may change from page to page within a page set.

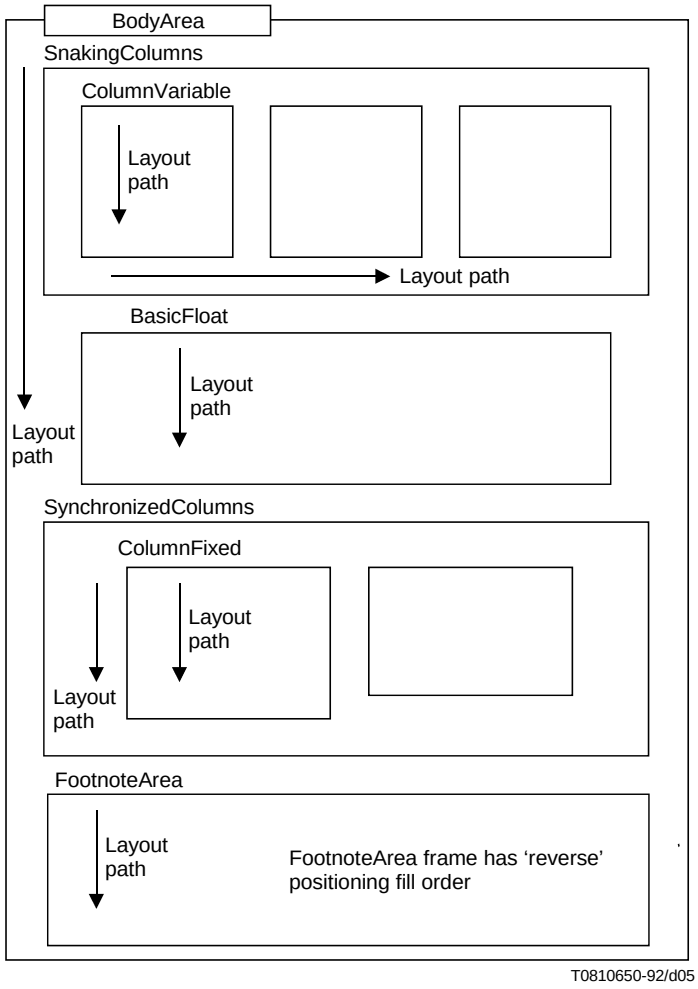


FIGURE 5/T.505

Example of body area layout for page layout A

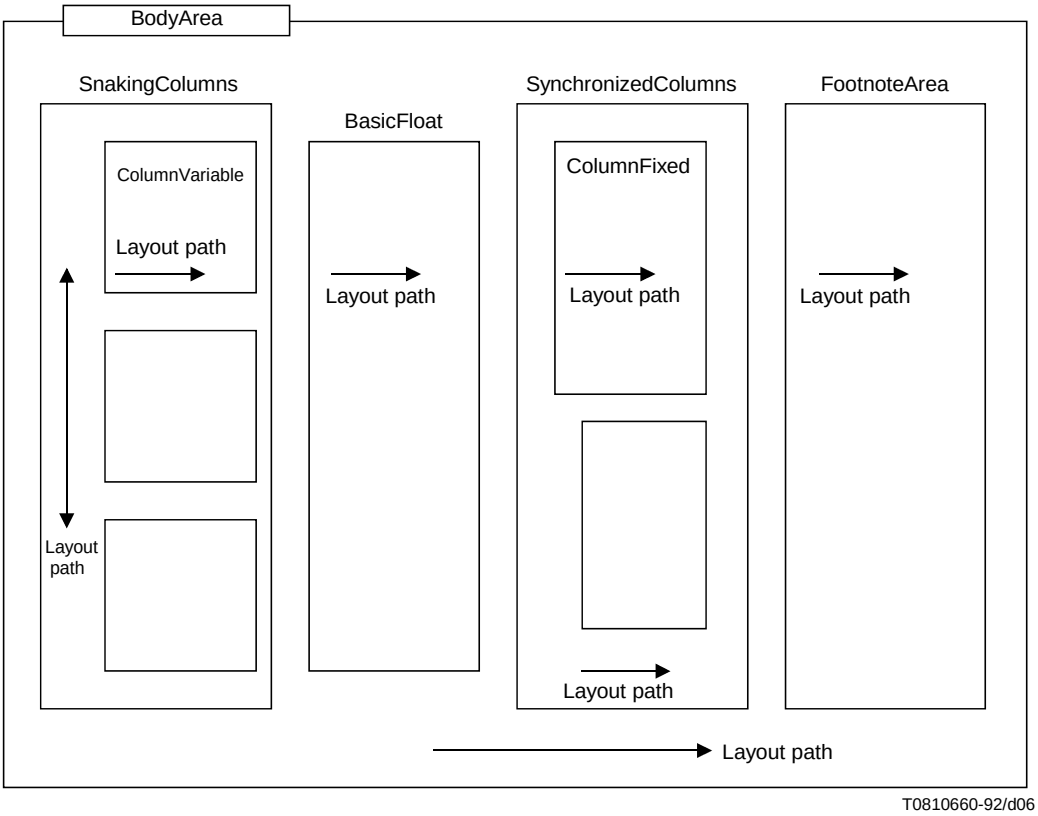


FIGURE 6/T.505

Example of body area layout for page layout B

6.3.5.4 BasicFloat

BasicFloat is a constituent constraint that defines a lowest level frame that is used to represent a single column area within a body area. A single column area is typically used to layout content in the form of a single column. This is a variably positioned frame.

The dimension of this frame in the direction orthogonal to the layout path of the body area is fixed or defaults to the maximum value allowed within the body area.

The dimension of this frame in the direction parallel to the layout path of the body area is specified by sub-parameter "Rule-B". This dimension is therefore automatically adjusted during the layout process to be the minimum required to contain all the content allocated to the frame.

The layout path specified for this frame is the same as that specified for the body area. Content shall only be laid out in this frame in the direction of the layout path specified.

6.3.5.5 SnakingColumns

SnakingColumns is a constituent constraint that defines a composite frame that represents a snaking columns area within a body area. A snaking columns area is typically used for the layout of one or more columns of content in which the content is allowed to flow freely from one column to the next.

This frame is variably positioned. Its immediate subordinates consist of one or more frames of the type ColumnVariable. Examples of the layout of SnakingColumns frames are given in Figure 8.

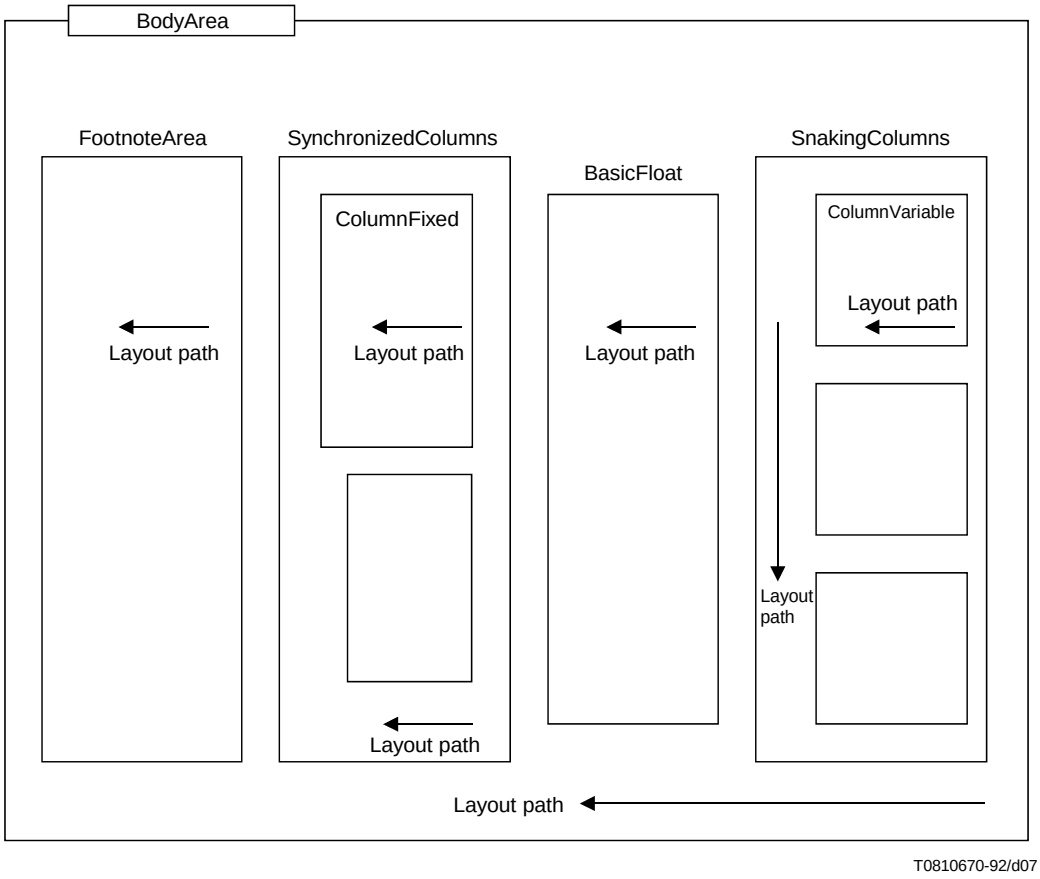


FIGURE 7/T.505

Example of body area layout for page layouts C and D

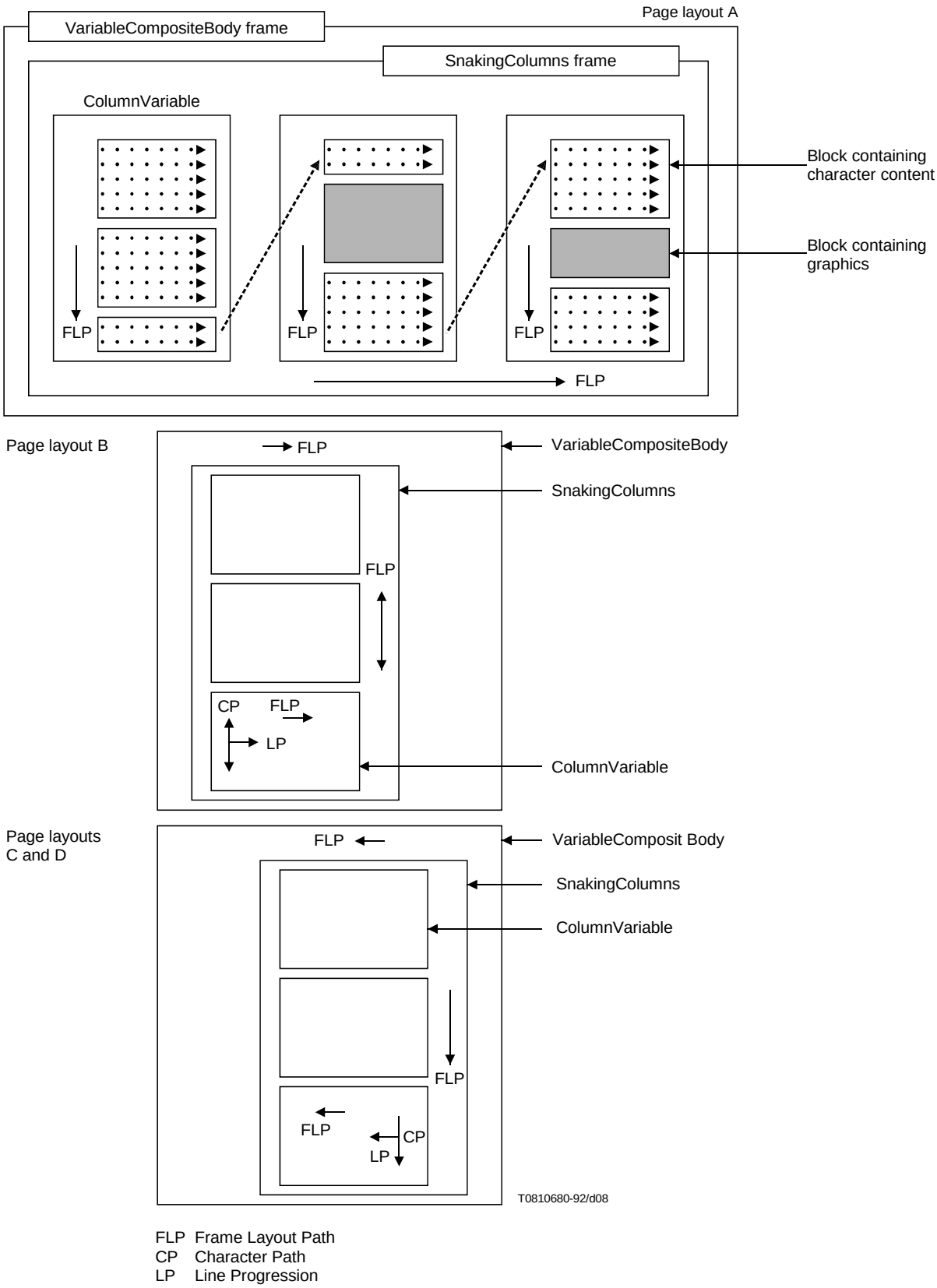
The dimension of a SnakingColumns frame in the direction orthogonal to the layout path of the body area is fixed or defaults to the maximum value allowed within the body area.

The dimension of this frame in the direction parallel to the layout path of the body area is specified by sub-parameter "Rule-B". This dimension is therefore automatically adjusted to accommodate the subordinate frames which are laid out in it.

The layout path for a SnakingColumns frame may be specified as 0° or 180° in the case of page layout A, 90° or 270° in the case of page layout B, and 270° in the cases of page layouts C and D.

The attribute "balance" may be specified for a SnakingColumns frame to indicate that two or more of the subordinate ColumnVariable frames are to be approximately equal in length in the vertical dimension in the case of page layout A and are to be approximately equal in length in the horizontal dimension in the cases of page layouts B, C and D. Note that the term *approximately equal* in the context of the attribute "balance" means that the leading edges of the layout objects being balanced are aligned, as closely as possible, to a line orthogonal to the layout path for the objects.

The attribute "balance" may be ignored when the subordinate ColumnVariable frames have unequal widths.



FLP Frame Layout Path
CP Character Path
LP Line Progression

FIGURE 8/T.505

Examples of the layout of snaking columns

6.3.5.6 SynchronizedColumns

SynchronizedColumns is a constituent constraint that defines a composite frame that represents a synchronized columns area within a body area. A synchronized columns area is typically used to represent one or more columns of content such that the content laid out in each column belongs to different layout streams. Thus content laid out in one column is not allowed to flow into the next column.

This type of column layout is typically used when it is required to layout separate amounts of content in parallel with one another such that they are aligned. Examples are the synchronized layout of content belonging to different languages and the layout of a figure in parallel with some text. An example is shown in Figure 9.

With regard to positioning and dimensioning, SynchronizedColumns frames have the same characteristics as SnakingColumns frames.

The immediate subordinates of a SynchronizedColumns frame consist of any number of frames of the type ColumnFixed.

The layout path for a SynchronizedColumns frame is 270° for page layout A, 0° for page layout B and 180° for page layouts C and D.

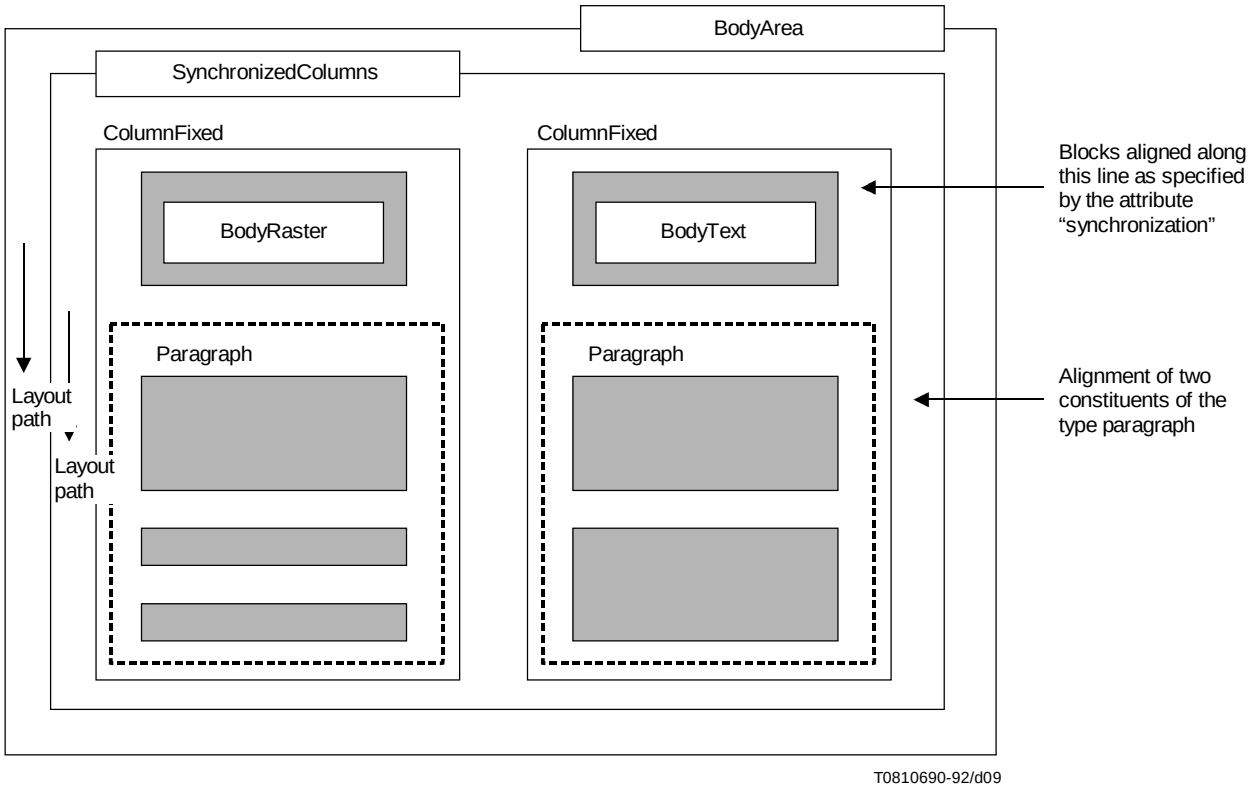


FIGURE 9/T.505

Example of synchronized column layout for page layout A

6.3.5.7 ColumnVariable

ColumnVariable is a constituent constraint that defines a lowest level frame that is used to represent a column of content within a SnakingColumns frame. This is a frame which is variably positioned.

The dimension of this frame in the direction parallel to the layout path of the superior SnakingColumns frame (that is, the column width) is fixed. The dimensions of different instances of ColumnVariable frames within a given SnakingColumns frame may differ to allow columns of different widths to be specified.

The dimension in the direction orthogonal to the layout path of the superior frame (that is, the column length) may be specified by the sub-parameters "Rule-B" or "maximum-size".

The layout path for ColumnVariable frames is 270° in the case of page layout A, 0° in page layout B and 180° in page layouts C and D.

All ColumnVariable frames subordinate to the same SnakingColumns frame shall have the same category name; different names may be used for ColumnVariable frames laid out in different SnakingColumns frames.

6.3.5.8 ColumnFixed

ColumnFixed is a constituent constraint that defines a lowest level frame that is used to represent a column of content within a SynchronizedColumns frame. This is a frame which has a fixed position.

The dimension of this frame in the direction orthogonal to the layout path of the superior SynchronizedColumns frame (that is, the column width) may be specified by either of the sub-parameters "fixed dimension" or "maximum size" in all page layout types. This dimension may differ for different instances of ColumnFixed frames within a given SynchronizedColumns frame to allow columns of different widths to be specified. However, the widths shall be specified such that the columns do not overlap.

The dimension of this frame in the direction parallel to the layout path of the superior frame (that is, the column length) may be specified by sub-parameters "Rule-B" or "maximum-size" in the cases of page layouts A and B. In the cases of page layouts C and D, this dimension shall only be specified as 'maximum-size'.

The ColumnFixed frames subordinate to a given SynchronizedColumns frame shall have different category names.

The layout path for ColumnFixed frames shall be equal to the layout path of the superior SynchronizedColumns frame.

The content laid out in different ColumnFixed frames within the same SynchronizedColumns frame may be specified synchronized by using the attribute "synchronization".

The sub-parameter "maximum size" shall only be specified for the right most ColumnFixed frames (relative to the page co-ordinate system) laid out in a SynchronizedColumns frame, to prevent overlapping of the frames.

6.3.5.9 FootnoteArea

FootnoteArea is a constituent constraint that defines a lowest level frame that is used to represent a footnote area within a body area. A footnote area is typically used for the layout of footnotes.

Frames of this type are variably positioned with a positioning fill order specified as 'reverse'. Hence this frame is positioned adjacent to the leading edge of the VariableCompositeBody frame.

The dimension of FootnoteArea frames in the direction orthogonal to the layout path of its superior frame is specified by one of the sub-parameters "fixed dimension" or "maximum size". In the direction of the layout path, the dimension is specified by sub-parameter "Rule-B" which means that this dimension is automatically adjusted to contain all the content that is allocated to it.

The layout path for FootnoteArea frames is the same as that specified for the body area.

The content that may be laid out in this frame is limited to the content that is associated with basic logical objects which are subordinates of the composite logical object FootnoteBody. To achieve this, the attribute "permitted categories" of this frame shall specify the value 'Footnote', the same name required on the basic logical objects for footnotes (see 6.2.3.10 and 6.2.3.11).

6.3.6 Header and footer area characteristics

6.3.6.1 General characteristics

The header and footer areas consist of either basic areas or composite areas.

A basic header or footer area is an area into which the content is directly laid out. This type of area is represented by a constituent constraint of the type BasicHeader or BasicFooter respectively.

A composite header or footer area is an area which is subdivided into separate sourced content and arranged content areas to provide greater versatility with regard to the layout of the content. This type of area is represented by a constituent constraint of the type CompositeHeader or CompositeFooter respectively.

In the case of basic header or footer areas, the content allocated to these areas is derived from the common part of the logical structure of a document. In the case of composite header or footer areas, the content may again be derived from the common part of the logical structure of a document but the content may also be derived from common content specified in the generic layout structure.

6.3.6.2 BasicHeader and BasicFooter

BasicHeader and *BasicFooter* are constituent constraints that define lowest level frames that represent areas within a page that are reserved for common content.

These types of frame have fixed positions and dimensions. The positioning of these frames within a page and the layout paths that may be specified for them depends upon the page layout type used (see 6.3.4.5).

The content that is laid out in these frames is derived, using the logical source mechanism, from the content associated with the composite logical object classes of the type CommonContent.

6.3.6.3 CompositeHeader and CompositeFooter

CompositeHeader and *CompositeFooter* are constituent constraints that define composite frames that represent areas within a page that are reserved for common content.

These types of frame have fixed positions and dimensions. The positioning of these frames within a page and the layout path that may be specified for them depends upon the page layout type used (see 6.3.4.5).

The subordinates of these frames may consist of either:

a) any number and combination of variably positioned frames of the types SourcedContentVariable and ArrangedContentVariable; or

b) any number and combination of fixed positioned frames of the types SourcedContentFixed and ArrangedContentFixed.

In case b), the subordinate frames may overlap without restriction.

6.3.6.4 SourcedContentVariable

A *SourcedContentVariable* frame is a constituent constraint that defines a lowest level frame that represents a region within a header or footer area that contains common content derived from the generic logical structure. This frame is variably positioned and its layout path is the same as that of the containing header or footer area.

The dimension of this frame in the direction orthogonal to the layout path of the superior frame is specified by one of the sub-parameters "fixed dimension" or "maximum-size". The dimension of the frame in the direction parallel to the layout path of the superior frame is specified by one of the sub-parameters "fixed dimension" or "Rule-B".

This frame is required to specify the attribute "logical source" which indicates the particular instance of the constituent constraint CommonContent which contains the content to be laid out in this frame.

Typically, this frame is used for the positioning of content which is generated during the layout process, such as a character sequence containing a page number.

6.3.6.5 ArrangedContentVariable

An *ArrangedContentVariable* frame is a constituent constraint that defines a lowest level frame that represents a region within a header or footer area that contains pre-defined common content contained in the generic layout structure.

This frame is variably positioned and its dimensions are fixed.

This frame refers to one or more blocks of type GenericBlock (see 6.3.7) which contain the content to be laid out in this frame. Thus, this frame is typically used when it is required to layout pre-determined common content.

6.3.6.6 SourcedContentFixed

A *SourcedContentFixed* frame is a constituent constraint that defines a lowest level frame that represents a region within a header or footer area that contains common content derived from the generic logical structure. This frame has a fixed position and its layout path is equal to that of the containing header or footer area.

The horizontal dimension of this frame is specified by the sub-parameter "fixed dimension". The vertical dimension of this frame is specified by one of the sub-parameters "fixed dimension" or "Rule-B".

This frame is required to specify the attribute "logical source" which indicates the particular instance of the constituent constraint CommonContent which contains the content to be laid out in this frame.

Thus, as in the case of SourcedContentVariable frames, this frame is used for the positioning of content which is generated during the layout process, such as a character sequence containing a page number.

6.3.6.7 ArrangedContentFixed

An *ArrangedContentFixed* frame is a constituent constraint that defines a lowest level frame that represents a region within a header or footer area that contains pre-defined common content derived from the generic layout structure.

The position and dimensions of this frame are fixed. This frame refers to one or several blocks of type GenericBlock (see 6.3.7) which contain the content to be laid out in this frame. Thus this frame is typically used when it is required to layout common content at pre-determined positions in the header or footer areas.

6.3.7 GenericBlock and SpecificBlock

Two types of constituent constraints of the object type 'block' are defined, namely *GenericBlock* and *SpecificBlock*. Object classes of the type GenericBlock may occur in the generic layout structure refersed to by the attribute "generator for subordinates" of object classes of the types ArrangedContentVariable and ArrangedContentFixed. When the layout process is performed to produce a document in formatted processable form, equivalent blocks may occur in the specific layout structure. Objects of this type are restricted to occur within the header and footer areas of the page.

Objects of the type SpecificBlock shall only occur in the specific layout structure. They are created during the document layout process and result from the layout of basic logical objects into lowest level frames that constitute the body, header and footer areas.

6.4 Document layout characteristics

Mechanisms for controlling the allocation of logical constituents to various areas in the layout structure are defined in 6.4.1. Mechanisms for controlling the layout of the content within the allocated areas are defined in 6.4.2.

These mechanisms relate to documents for which a generic layout structure is specified. When a generic layout structure is not present, then these mechanisms are restricted as described in 6.4.3.

6.4.1 Flow controls

Various mechanisms are provided to control the allocation of constituent constraints representing the body parts of the logical structure of a document to pages sets, pages and body areas. These are described in 6.4.1.1, 6.4.1.2 and 6.4.1.3. The mechanisms for controlling the layout of the common parts of a document are described in 6.4.1.4.

6.4.1.1 Allocation of content to page sets

Two methods of allocating the constituent constraints associated with the *body* part of the document to page sets are provided:

- layout in a nominated page set;
- starting a new page set.

The first method provides the ability to specify that a part of a document is to be laid out entirely within a specified page set. This may be specified for constituent constraints of the types Passage, NumberedSegment and Paragraph using the attribute "layout object class" which specifies the object class identifier of the required class of page set.

The second method provides the ability to specify that the logical objects derived from a particular constituent constraint in a document and all subsequent parts of a document are to be laid out starting at the beginning of a new page set. This may be specified for the following constituent constraints:

- Passage;
- NumberedSegment;
- Paragraph;
- Number;
- BodyText;
- BodyRaster;
- BodyGeometric.

This is achieved using the attribute "new layout object" which specifies the object class identifier of the required class of page set.

6.4.1.2 Page breaks

This provides the ability to specify that the logical objects derived from a particular constituent constraint in a document and all subsequent parts of a document are to be laid out starting at the beginning of a new page. The page specified shall

belong to the page set in which the logical objects from the immediately preceding constituent constraint is laid out. The specification of a page break shall not be used to layout part of a document in a new page set. If a new page set is required, then this shall be explicitly specified as described in 6.4.1.1.

This may be specified for the following constituent constraints:

- Passage;
- NumberedSegment;
- Paragraph;
- Number;
- BodyText;
- BodyRaster;
- BodyGeometric.

This is achieved using the attribute "new layout object". This attribute may specify the value 'page' indicating that the constituent constraint is to be laid out starting on the next available page which may be of any class. Alternatively, the attribute may specify that the constituent constraint is to be laid out starting on a page of a particular class; this is achieved by specifying the object class identifier of the required page class.

6.4.1.3 Allocation of content to body areas

If the page to which the content is allocated contains a basic body area, then the content is laid out in sequential order in that body area in the form of a single column.

If the page contains a composite body area, then the content is allocated to single, snaking and synchronized columns areas and footnote areas as described below.

6.4.1.3.1 Layout of content into column areas

When laying out content into a composite body area having more than one subordinate frame class (excluding FootnoteArea frame classes), it is necessary to indicate which of the column areas is to be used.

Logical objects of the types Number, FootnoteReference, BodyText, BodyRaster and BodyGeometric may be specified to be laid out in instances of one or more single columns area, snaking columns area or synchronized columns area.

This is done by giving each such basic logical component a value of the attribute "layout category" which corresponds to the value of the attribute "permitted categories" which applies to the lowest level of frame in which the content is to be laid out.

Note that any basic logical objects in the specific logical structure to which this attribute does not apply will be laid out only in a lowest level frame which has the implicit value of the attribute "permitted categories".

The use of layout categories ensures that if there is insufficient area on one page to lay out all of the content allocated to a particular type of area, then the laying out of the content will automatically continue in the same type of area in a succeeding page. Thus content is allowed to flow freely from one page to the next when the type of layout used at the end of one page is the same as that at the beginning of the succeeding page.

It is necessary to ensure the correct use of the mechanism for the layout of independent layout streams. In the absence of additional layout directives, content may be placed in available space within an earlier frame with appropriate values for the attribute "permitted categories". If this is not intended, it may be prevented by the use of the attribute "new layout object".

The attribute "new layout object" may be applied to logical components of the types NumberedSegment, Paragraph, Number, FootnoteReference, BodyText, BodyRaster and BodyGeometric, whenever a change in column layout is required.

The attribute "new layout object" may specify the identifier of the frame class that represents the single, snaking or synchronized column area required. In the case of single or synchronised column areas, the attribute "new layout object" may indicate the category name corresponding to the frame class of the single column area or of any of the columns within the synchronised column area that is required.

When layout occurs in a snaking columns area, column breaks may be indicated by using the attribute "new layout object". This attribute may specify the identifier or the category name of the frame corresponding to the column in which the layout is to continue. However only the use of a category name will ensure that a single column break is always obtained, irrespective of the frame class actually used.

When the layout is to occur in a synchronized columns area, category names are used to control the particular columns that are to be used to layout the logical entities. Each column within a synchronized columns area shall have a different permitted category and each basic logical entity to be laid out in this particular area shall have a category name corresponding to a name allocated to one of the columns. The logical entities allocated to different columns may be aligned using the attribute "synchronization".

6.4.1.3.2 Layout of footnotes

The basic logical objects derived from constituent constraints that represent the content belonging to a footnote (i.e FootnoteNumber and FootnoteText) are constrained to be laid out in a footnote area which is represented by a FootnoteArea frame (see 6.3.5.9).

This constraint is specified by means of category names. That is, the logical constituents of the types FootnoteNumber and FootnoteText and layout constituents of the type FootnoteArea are all required to have the category name with the value 'Footnote'.

More than one footnote may be placed in a footnote area within a given body area. In this case the content belonging to the footnotes are laid out sequentially in the footnote area in accordance with their reading order.

If the content belonging to a footnote cannot all be accommodated in the footnote area on one page, then the content may freely flow into the next footnote area on the subsequent page. Alternatively, it is possible to specify that a footnote is to be laid out entirely within a particular footnote area. This is achieved using the attribute "indivisibility".

6.4.1.4 Allocation of content to header-footer areas

A header or footer area may be basic or composite (see 6.3.6.1). In the case of a basic area, the frame representing that area specifies the attribute "logical source" which indicates the particular instance of the constituent constraint of the type CommonContent that is to be laid out in that area. The basic logical constituents subordinate to CommonContent are then laid out in accordance with their sequential order.

In the case of a composite header or footer area (see 6.3.6.3), the area is divided into one or more separate areas, each of which is represented by a lowest level frame. The content allocated to the separate areas may be derived from one of two sources. That is, the content may be pre-defined and represented by one or more blocks which are directly associated with the lowest level frame. Alternatively, the lowest level frame may specify the attribute "logical source" which, as above, indicates the particular logical object of the type CommonContent that is to be laid out in that frame.

6.4.2 Layout of the document content

Various constraints may be specified to control the layout of the content into the body, header and footer areas. These constraints are described below.

6.4.2.1 Margins

The margins are the minimum distances, or offsets, between a part of the document content and the edge of the particular area in which that content is laid out. The margins define the maximum extents of the available area into which the content shall be positioned.

Margins may be specified for any constituent constraint representing a basic logical object; different margin values may be specified for different constituent constraints without restriction.

Four margins may be independently specified for each constituent constraint, namely:

- trailing edge margin;
- leading edge margin;
- right hand edge margin;
- left hand edge margin.

These margins are defined in relationship to the layout path specified for the frame in which the content is to be laid out in (see Figure 10).

Any combination of the above margins may be specified for a particular constituent constraint. These margins are specified by the attribute "offset". Any value may be specified in units of BMUs. If a particular margin is not specified, then it is assumed to be 0 BMUs.

6.4.2.2 Separation

Leading separation is the minimum distance between one basic logical object and the next one, if any, when they are laid out; trailing separation is the minimum distance between one basic logical object and the previous one, if any, when they

are laid out. Both may be specified for basic logical components of any constituent constraint types. These distances are specified in BMUs by the attribute "separation". If no value is specified, then the minimum distance is assumed to be 0 BMUs.

6.4.2.3 Indivisibility

Indivisibility provides the means to specify whether or not a basic or composite logical object derived from a constituent constraint is allowed to be split over more than one page or over more than one area within a page. It may be specified for constituent constraints of the types Passage, NumberedSegment, Paragraph, Footnote, Number, FootnoteReference and BodyText. The attribute "indivisibility" is used to specify this feature.

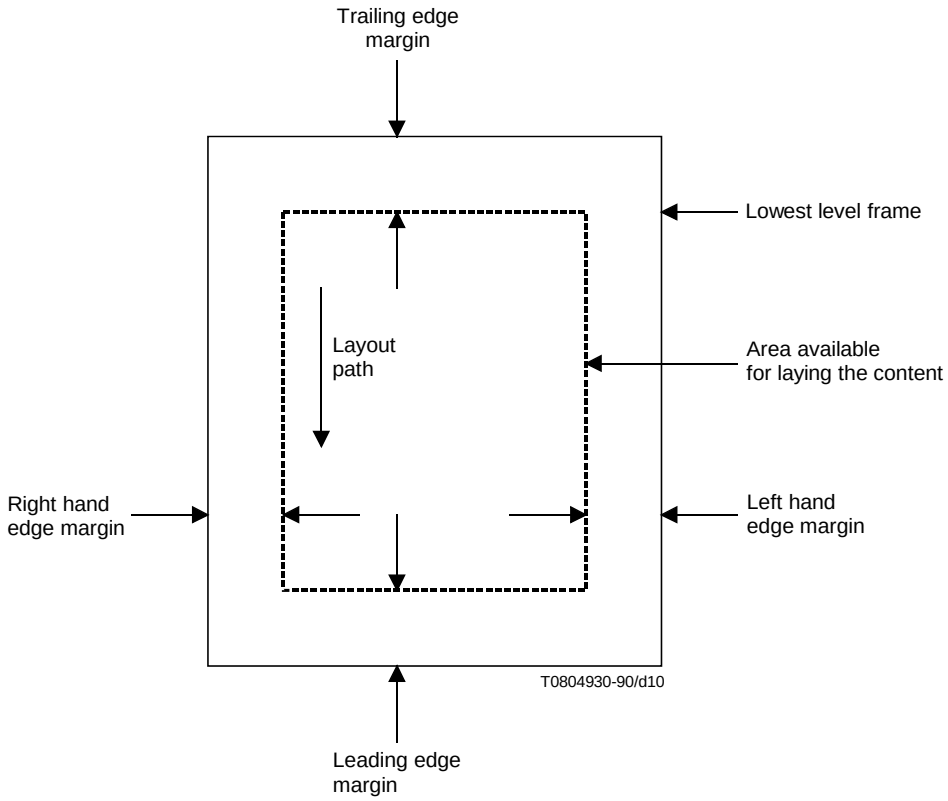


FIGURE 10/T.505

Specification of margins

6.4.2.4 Same layout object

Same layout object provides the means to specify that the start of the content associated with a logical object and the end of the content associated with the previous logical object are to be laid out within a single layout object. This may be specified for logical objects of the types NumberedSegment, Paragraph, Number, Footnote, FootnoteReference, BodyText, BodyRaster and BodyGeometric. The attribute "same layout object" is used to specify this feature.

6.4.2.5 Concatenation

Concatenation provides the means to specify that the content associated with a basic logical object derived from a constituent constraint and the content associated with the basic logical object derived from a previous constituent constraint are to be regarded as an unbroken stream of content. This may be specified for constituent constraints of the types BodyText, Number, FootnoteReference, FootnoteText, CommonText and PageNumber. The attribute "concatenation" is used to specify this feature.

6.4.2.6 Block alignment

Block alignment allows the content associated with a basic logical object to be specified as 'left aligned', 'right aligned' or 'centred' within the area in which that content is laid out. Left aligned means that the content is laid out adjacent to the left hand edge margin. Right aligned means that the content is laid out adjacent to the right hand edge margin and centred means that the content is laid out midway between the left and right margins.

This feature may only be specified using the attribute "block alignment" for constituent constraints of the types Number, FootnoteReference, FootnoteText, PageNumber, FootnoteNumber, BodyText and CommonText, and when they contain formatted character content, BodyRaster, BodyGeometric, CommonRaster and CommonGeometric.

6.4.3 Layout controls applicable in the absence of a generic layout structure

In processable form documents the generic layout structure is optional. If the generic layout structure is omitted, then it is the responsibility of the receiver to define an appropriate layout structure. No limitations are placed on the layout structure that is used.

When a generic layout structure is not specified within a processable form document, then restrictions are placed on the layout control functions described in 6.4.1 and 6.4.2 that may be specified within the document. These restrictions are indicated below.

- It is not possible to specify that certain logical parts of a document are to be allocated to a given page set or that a part of a document is to be laid out starting in a new page set, as defined in 6.4.1.1.
- It is possible to specify page breaks as defined in 6.4.1.2 but it is only possible to indicate that the layout shall begin on a new page. It is not possible to specify a particular page class.
- The logical parts of the document that are intended to be laid out in the body area and in the header/footer areas of each page may be distinguished from each other by means of application comments specified for them (see 6.6.4). An exception is that it is not possible to distinguish whether a particular portion of the common content is to be placed in a header or footer area (or both).
- It is not possible to indicate the type of layout area to be used to layout each logical constituent in the body part of a document. That is, it is not possible to indicate whether single column or multiple column areas are to be used (see 6.4.1.3.1). This shall be decided by the receiver.
- Footnotes within the body part of a document may be distinguished by use of the attribute "application comments". Footnotes are intended to be read and laid out separately from the other logical constituents of the body part (see 6.4.1.3.2). However, it is the responsibility of the receiver to decide how footnotes are laid out.
- Margins, separation, indivisibility, same layout object, concatenation, and block alignment, as defined in 6.4.2 may all be specified. Only one restriction applies. Indivisibility (see 6.4.2.3) may be assumed to specify that a logical object derived from the constituent constraint is not to split over more than one page but indivisibility shall not be specified for other types of layout areas such as single or multiple column areas.

6.5 Content layout and imaging characteristics

A document may contain character, raster graphics and geometric graphics content.

The content architectures that may be specified using the attribute "content architecture class" are formatted character, processable character, formatted processable character, formatted processable raster graphics and formatted processable geometric graphics. Any of these may be specified as the default in the document profile.

6.5.1 Character content

6.5.1.1 Introduction

This subclause defines the features that are applicable to the character content contained in a document and the presentation attributes and control functions that may be used to specify these features. These features may apply to basic logical and layout components unless otherwise indicated.

The default values for the following features may be specified in the document profile:

- graphic character sets;
- graphic character subrepertoire;
- code extension announcers;
- line spacing;

– character spacing;
– character path;
– line progression;
– character orientation;
– graphic rendition, including the parameters values: default rendition, increased intensity (bold), italicized, underlined, crossed-out, primary font, 1st alternative font, 2nd alternative font, 3rd alternative font, 4th alternative font, 5th alternative font, 6th alternative font, 7th alternative font, 8th alternative font, 9th alternative font, doubly underlined, normal intensity, not italicized, not underlined, not crossed out;
– line layout table;
– indentation;
– alignment;
– first line offset;
– itemization;
– widow size;
– orphan size;
– character fonts;
– kerning offset;
– proportional line spacing;
– initial offset.

The specification in a document of a non-basic feature by a presentation attribute or control function shall be indicated in the document profile.

6.5.1.2 Character content architecture class

Processable and formatted processable form documents may contain processable, formatted or formatted processable character content. Formatted form documents may contain formatted and formatted processable character content.

When using character content, any number of content portions may be associated with a basic component.

The content information in a content portion may be absent. This is to allow the representation and interchange of documents in which parts of the content may be supplied, for example, during subsequent editing.

6.5.1.3 Character repertoires

The basic character repertoire supported by this Profile is composed of the 94 characters of ISO-IR 6 (the IRV of ISO/IEC 646) plus the character space.

Any other graphic character set which is registered in accordance with ISO 2375 may be designated and invoked at any point in the document provided its use is indicated in the document profile as a non-basic value using the character presentation feature attribute "graphic character sets". No locking shift functions are specified in this presentation feature.

The code extension techniques allowed for the designation and invocation of character sets to the left hand side and right hand side of the 8-bit code table (GL and GR respectively) are defined in 6.5.1.4.

Using these code extension techniques, the graphic character sets designated and/or invoked at the beginning of a content portion containing character content are specified by the presentation attribute "graphic character sets". The graphic character sets may also be changed at any point within a content portion.

The default graphic character sets which apply to the content portions within a document may be specified in the document profile using the presentation attribute "graphic character sets".

If the character set defined in ISO 6937 is designated and invoked, then the use of any of its sub-repertoires registered according to ISO/IEC 7350 may be specified using the presentation attribute "graphic character sub-repertoire". All sub-repertoires are non-basic and their use shall be indicated in the document profile. The sub-repertoire shall not be changed within a content portion.

NOTE – The basic character repertoire supported by this Profile is not the standard default value specified in CCITT Rec. T.416 | ISO/IEC 8613-6; hence it may be necessary to specify, in the document profile of a particular document, that this is the default value being used for that document.

6.5.1.4 Code extension techniques

The code extension techniques specified in ISO 2022 may be used subject to the following restrictions:

a) G0 set: only ISO-IR 6 (the IRV of ISO/IEC 646), ISO-IR 2 (the primary set of ISO 6937-2), or any other version of ISO/IEC 646 may be designated for this set; these graphic character sets may only be invoked in GL.

b) G1, G2, G3 sets: no restrictions are placed on the character sets that may be designated for these sets; these graphic character sets may only be invoked in GR.

c) The locking and single shift functions allowed are as follows:

- LS0 to invoke the G0 set into GL;
- LS1R to invoke the G1 set into GR;
- LS2R to invoke the G2 set into GR;
- LS3R to invoke the G3 set into GR;
- SS2 to invoke one character from the G2 set into GL;
- SS3 to invoke one character from the G3 set into GL.

(Here GL and GR refer to the left and right hand parts respectively of the 8-bit code table)

d) When specifying the presentation attribute "graphic character sets", it is necessary to invoke character sets for both GL and GR. Thus an allowed character set shall be designated into G0 (see item a) above) and invoked into GL. It is also necessary to invoke a character set into GR which has been designated into the G1, G2 or G3 set.

e) The empty set shall be designated into G1 and invoked into GR if no other specific character set is invoked into GR.

The code extension techniques allowed are illustrated in Figures 11 and 12.

The announcement and encoding of these functions are to be as specified in ISO 2022.

The code extension techniques that are used or may be used in a basic component shall be specified by the presentation attribute "code extension announcers". The default code extension announcers used throughout a document may be specified in the document profile, using the presentation attribute "code extension announcers".

NOTE – In accordance with CCITT Rec. T.416 | ISO/IEC 8613-6, there is no restriction concerning the number of graphic character sets which may be designated and/or invoked in the presentation attribute "graphic character sets" providing the restrictions defined in this subclause are applied. Hence designation to a particular G set overrides a previous designation to that set and invocation to GL or GR overrides the previous invocation to the GL or GR respectively. Thus the sequential order of designation and/or invocation sequences in the attribute "graphic character sets" is significant.

6.5.1.5 Line spacing

Any value of line spacing may be specified. The values of 150, 200, 300 and 400 BMUs are basic; the use of any other value in a document is non-basic and shall be indicated in the document profile.

The line spacing may be specified at the beginning of the content associated with a basic component using the presentation attribute "line spacing". The value may be changed anywhere within the content portion using the control functions SVS and SLS.

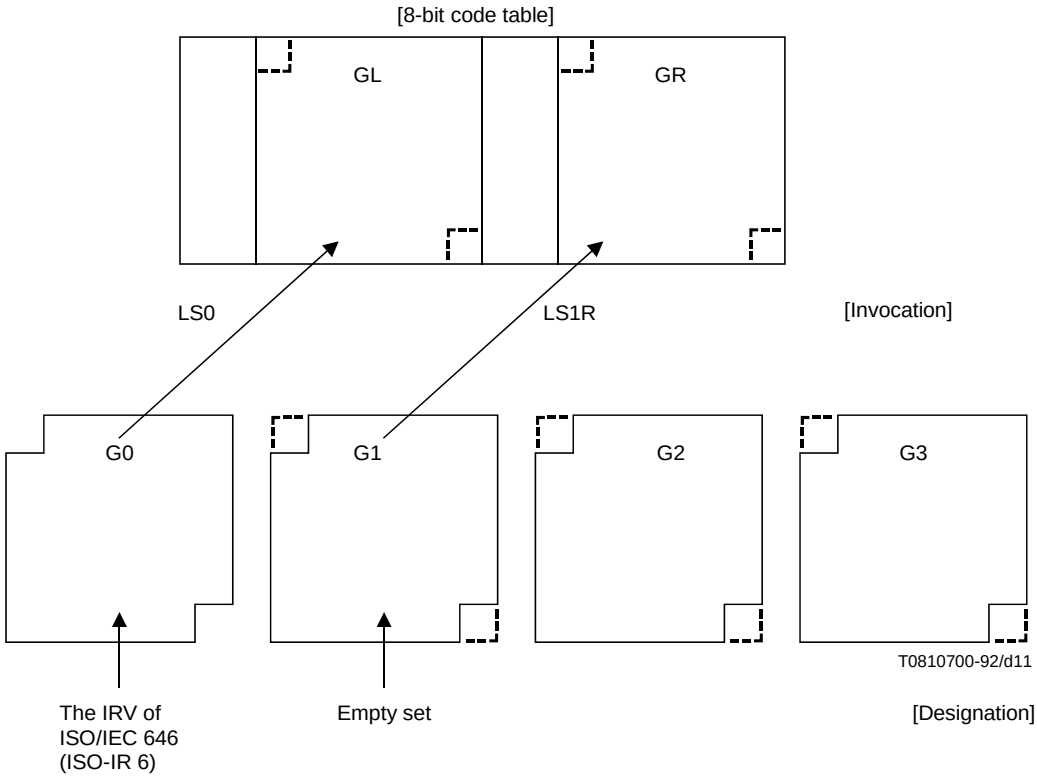


FIGURE 11/T.505

Code extension features (basic case)

6.5.1.6 Character spacing

Any value of character spacing may be specified. The values greater than or equal to 100 are basic; the use of any other value in a document is non-basic and shall be indicated in the document profile.

The character spacing may be specified at the beginning of the content associated with a basic component using the attribute “character spacing”. The value may be changed anywhere within a content portion using the control functions SHS or SCS.

NOTES

1 A character spacing value of 160 BMUs is provided for use with Korean Hangul characters.

2 SHS parameters of 0, 1, 2, 3 and 4 are provided. The use of parameters 5 and 6 may be provided in a future edition of this Recommendation for use with Chinese characters.

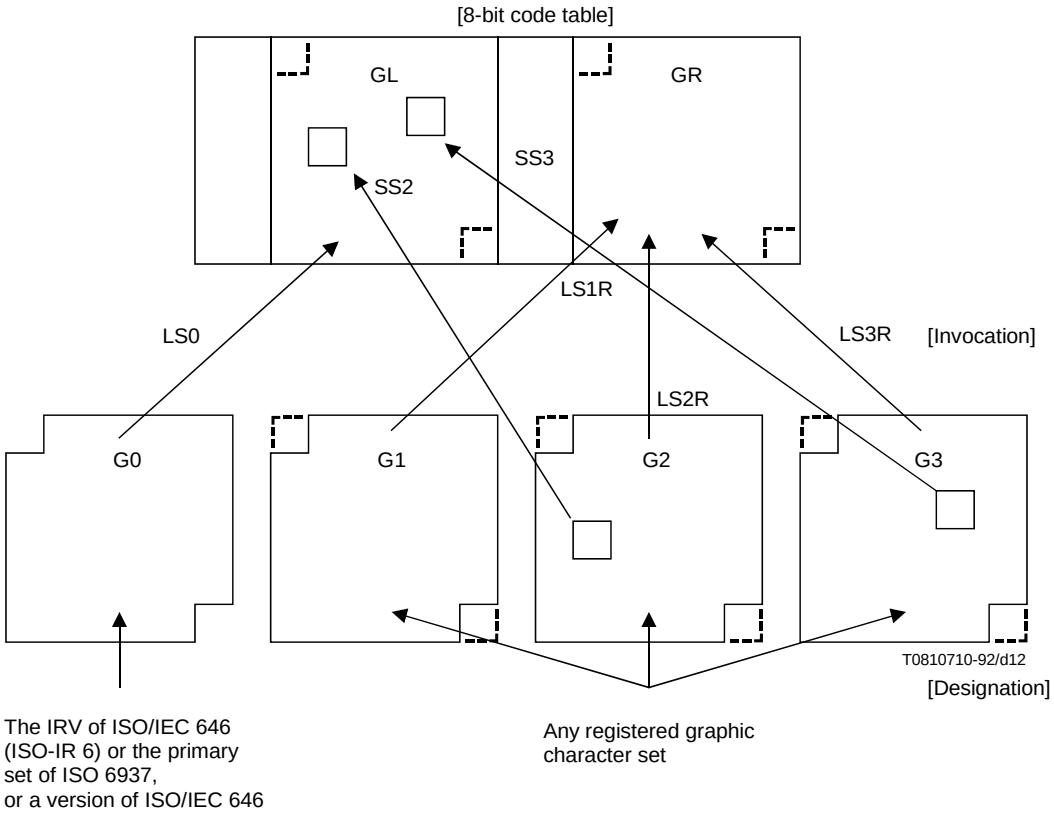


FIGURE 12/T.505

Code extension features (all possible cases)

6.5.1.7 Character path and line progression

Both horizontal and vertical writing directions may be used within a document. In the case of horizontal writing, the characters progress either from left to right or from right to left across the page and the line progression is from the top of the page to the bottom. In the case of vertical writing, the characters progress from the top of the page to the bottom and the line progression is from the right to the left. The use of these writing directions is restricted by the page layout type used.

For page layout A, only horizontal writing may be used in the body, header and footer areas. Thus, in this case the character path and line progression is specified either as 0° and 270° respectively or 180° and 90° respectively.

For page layout B, again only horizontal writing may be used in the body, header and footer areas. However, in this case the content in the body area is presented for viewing with the page in landscape orientation and the content in the header and footer areas is presented for viewing when the page is in the portrait orientation.

Thus for page layout B, in the body area the character path and line progression is specified either as 90° and 270° respectively or 270° and 90° respectively. In the header and footer areas, the character path and line progression is specified as in page layout A.

For page layout C, only vertical writing may be used in the body area and only horizontal writing may be used in the header and footer areas. Thus in the body area the character path and line progression are specified as 270° and 270° respectively. In the header and footer areas, the character path and line progression is specified as in page layout A.

For page layout D, only vertical writing may be used in the body, header and footer areas. Thus in all these areas, the character path and line progression are specified as 270° and 270° respectively.

A character path value of 0° and a line progression value of 270° are basic values. All other values are non-basic and their use in a document shall be indicated in the document profile.

The values of character path and line progression may be specified at the beginning of the content associated with a basic component using the presentation attributes "character path" and "line progression" respectively. These values cannot be changed within a content portion.

6.5.1.8 Character positioning controls

The active position of a character (as defined in CCITT Rec. T.416 | ISO/IEC 8613-6) can be moved forward or backward along the direction of the line progression using the control functions Line Position Backward (VPB) and Line Position Relative (VPR). These control functions may be specified in all forms of character content, and any parameter value may be specified.

6.5.1.9 Character orientation

The character orientation may be specified as 0° or 90° depending on whether vertical or horizontal writing is used (see 6.5.1.7).

When horizontal writing is used, characters may only be orientated at 0°. When vertical writing is used, characters may be orientated at 90°.

A value of 0° is basic; a value of 90° is non-basic and its use in a document shall be indicated in the document profile.

The value of the character orientation is specified at the beginning of the content associated with a basic component by the presentation attribute "character orientation". This value cannot be changed within the content.

6.5.1.10 Emphasis

The following modes of emphasizing graphic characters may be distinguished:

- default rendition;
- normal intensity;
- increased intensity (bold);
- italicized;
- not italicized;
- underlined;
- doubly underlined;
- not underlined;
- crossed-out;
- not crossed-out.

All the above mentioned modes of emphasis are basic. If no default mode is explicitly specified in the document profile, then the default mode is 'default rendition'.

The mode of emphasis may be specified at the beginning of the content associated with a basic component using the presentation attribute "graphic rendition". The mode may be changed anywhere within the content using the control function SGR.

The mode of emphasis remains in effect within the content associated with a basic component until changed into a mutually exclusive mode or by the specification of 'default rendition'. Mutually exclusive modes are normal/increased intensity, italicized/not italicized, underlined/doubly underlined/not underlined and crossed out/not crossed-out. One mode from each mutually exclusive set may be in operation at any point in the document content.

'Default rendition' cancels the effect of all modes of emphasis that are currently in operation and specifies that the text shall be displayed in accordance with the default rendition parameters set for the presentation device. Thus, for example, if it is required to ensure that the content is not underlined, then it is necessary to explicitly specify that underlined is not to be used.

6.5.1.11 Tabulation

Tabulation stop positions may be specified at any position along the character path. Each stop is specified by means of the following:

a) the tabulation position relative to the margin position in the direction opposite to the character path;

b) an optional alignment qualifier that specifies the type of alignment to be used at the designated tabulation position. The type may be specified as one of the following:

   - start aligned;

– end aligned;

– centred;

– aligned around.

These alignment qualifiers are defined in CCITT Rec. T.416 | ISO/IEC 8613-6. If the alignment qualifier is not explicitly specified, then it is assumed that start aligned is to be used.

Only one set of tabulation stops can be specified to be applicable to the content associated with a basic component. No limit is placed on the number of tabulation stops that may be specified within a given set.

The set of tabulation stop positions associated with the content of a basic component are specified using the presentation attribute "line layout table". Tabulation stop positions are invoked within the content using the control function STAB.

The tabulation reference numbers used in the control function STAB and associated presentation attribute "line layout table" shall be chosen so that, in any given line layout table the reference numbers are unique, sequential in the direction of the character path and do not include leading zeroes.

6.5.1.12 Indentation

Indentation is the distance between the first character on a line of content and the position of the margin that is in the direction opposite to the direction of the character path. Thus the value of the indentation specified determines the line home position (as defined in CCITT Rec. T.416 | ISO/IEC 8613-6).

Indentation acts as temporary alteration in the position of the offset in the direction opposite to the direction of the character path. When text is formatted, it is intended to be laid out between the indentation position and the margin position in the direction of the character path.

Any value of indentation may be specified for basic logical components using the presentation attribute "indentation". The indentation value shall not be changed within a content portion.

6.5.1.13 Alignment

This feature is concerned with how the first and last characters on each line of character content are to be laid out during the formatting process.

The following values of alignment may be specified as basic:

– start aligned;

– end aligned;

– centred;

– justified.

The semantics of these values are as defined in CCITT Rec. T.416 | ISO/IEC 8613-6.

The presentation attribute "alignment" is used to specify the alignment that is applicable to the content associated with a basic component. The alignment value cannot be changed within a content portion.

6.5.1.14 First line format

This feature specifies how the first line of the content associated with a basic component is to be laid out and provides for the itemisation of paragraphs.

It allows the first character in the content to be positioned at some point along the character path relative to the indentation position (as specified in 6.5.1.12). This point may be in the direction of the character path or in the direction opposite to the direction of the character path relative to the indentation position.

In addition, this feature provides for the specification of an item identifier on the first line. The item identifier is a string of characters that precedes and is separated from the remaining characters that form the first line. The control function Carriage Return (CR) is used as the separator.

The features provided correspond to examples 10.1 to 10.5 shown in Figure 10 of CCITT Rec. T.416 | ISO/IEC 8613-6.

First line format is specified by the presentation attributes "first line offset", "itemisation", and "indentation". Only those values of the attributes which combine to form the examples shown in Figure 10 of CCITT Rec. T.416 | ISO/IEC 8613-6 may be used.

6.5.1.15 Widow and orphan sizes

The widow size specifies the minimum number of lines of content that shall be allocated to a following frame or page when the content associated with a basic logical component is laid out such that it flows over two frames or pages. To accommodate this, it may be necessary to move a number of lines of content from one frame or page to the next frame or page.

The orphan size specifies the minimum number of lines of content that shall be placed in the current frame or page when the content associated with a basic logical component is split over two frames or pages. If this minimum cannot be accommodated, then the whole content shall be placed in the next frame or page.

Any value of widow or orphan size may be specified using the presentation attributes “widow size” and “orphan size” respectively.

Widow and orphan size may only be specified for character content placed in body area of pages.

6.5.1.16 Fonts

Any number of fonts may be used within a document. The fonts used in a particular document are specified in the document profile using the attribute “fonts list”.

Further information concerning the specification of font references in the document profile is given in B.2.

The fonts that may be used within the content associated with each basic component are specified by the presentation attribute “character fonts”. Up to 10 fonts taken from the list specified by the attribute “fonts list” may be specified by the attribute “character fonts”.

The font to be used at the start of the content associated with a basic component is specified using the attribute “graphic rendition”. The fonts used within the content may be changed using the control function SGR.

The document profile may specify, using the attribute “character fonts”, a default set of up to 10 fonts that are applicable to the whole document.

6.5.1.17 Reverse character strings

Bi-directional writing is supported by this Profile. Hence, a string of characters in a content portion associated with a basic component may be specified to be imaged in the reverse direction of the immediately preceding character string. Such strings may be specified by the control function SRS as defined in CCITT Rec. T.416 | ISO/IEC 8613-6.

This control function is provided for cases in which the text belongs to different languages and the character content is written, for example, from left to right or from right to left within the same line of characters, dependent upon the language and/or character set being used.

NOTE – The use of this control function cannot be indicated in the document profile. Thus it is intended that implementations shall ignore this control function when reverse character string layout and presentation is not supported.

6.5.1.18 Kerning offset

A kerning offset value for the content associated with a basic component may be specified using the presentation attribute “kerning offset”. It is necessary to specify such a value when certain fonts are invoked to ensure that no part of character images are positioned outside the boundary of the available area.

6.5.1.19 Proportional line spacing

The use of proportional line spacing may be invoked for the content associated with a basic logical component using the attribute “proportional line spacing”. When this invocation occurs, the line spacing between each pair of consecutive lines is determined in an implementation-defined way from the attributes associated with the fonts used within the two lines and may vary from one line to the next. This process is application dependent.

6.5.1.20 Superscripts and subscripts

Superscripts and subscripts may be specified anywhere within the content associated with a basic component by using the control functions PLU and PLD. The use of these control functions shall be in accordance with CCITT Rec. T.416 | ISO/IEC 8613-6.

6.5.1.21 Line breaks

The control functions BPH and NBH may be inserted in processable and formatted precessable form character content to indicate where line breaks may occur or may not occur respectively, when the content is laid out.

6.5.1.22 Substitution of characters

The control function SUB is provided to represent characters produced by a local system that cannot be represented by a character within a character set supported by this Profile.

6.5.1.23 Initial point

The initial point which is applicable to basic layout components may be specified by the attribute "initial offset". Any value may be specified.

6.5.1.24 Use of control functions

The following is a list of all the control functions and parameter values (where applicable) that may be specified in character content:

SHS Select Character Spacing (allowed parameter values: 0, 1, 2, 3, 4)

SCS Set Character Spacing (allowed parameter values: any)

SVS Select Vertical Spacing (allowed parameter values: any)

SLS Set Line Spacing (allowed parameter values: any)

SGR Select Graphic Rendition (allowed parameter values: 0, 1, 2, 3, 4, 9-19, 21-24, 29)

STAB Selective Tabulation (allowed parameter values: any)

SRS Start Reverse String (allowed parameters: any)

VPB Line Position Backward (allowed parameter values: any)

VPR Line Position Relative (allowed parameter values: any)

PLD Partial Line Down

PLU Partial Line Up

BPH Break Permitted Here

NBH No Break Here

JFY No Justify

SUB Substitute

SP Space

CR Carriage Return

LF Line Feed

SOS Start Of String

ST String Terminator

Code extension control functions (see 6.5.1.4)

The use of all these control functions, with the exception of SP, CR, LF, SOS and ST are described in various subclauses in 6.5.1.

6.5.1.25 Formatting the content

The attribute "formatting indicator" shall not be specified within documents that are conformant with this Profile.

6.5.2 Raster graphics content

6.5.2.1 Introduction

This subclause defines the features that are applicable to the raster graphics content contained in a document. These features may apply to basic logical and layout components unless otherwise indicated.

The default values for the following features may be specified in the document profile:

- type of coding;
- compression;
- pel spacing;
- spacing ratio;
- image dimensions.

The specification in a document of a non-basic feature by a presentation or coding attribute or control function shall be indicated in the document profile.

6.5.2.2 Raster graphics content architecture

Only the formatted processable raster graphics content architecture class may be used in documents that conform to this document application profile. This type of content may be used in processable, formatted and formatted processable form documents.

When using raster graphics content, only one content portion may be associated with an object or object class.

The content information in a content portion may be absent. This is to allow the representation and interchange of documents in which parts of the content may be supplied, for example, during subsequent editing.

Also, the scalable or fixed dimension content layout process may be used when laying out and imaging the content depending upon the specification of the presentation attributes "pel spacing" and "imaging dimensions" as described in 6.5.2.6 and 6.5.2.8. Both forms of content layout processes may be used in a single document.

6.5.2.3 Raster graphics encoding methods

The content may be encoded in accordance with the encoding schemes defined in CCITT Recommendations T.4 and T.6. In the case of Recommendation T.4, either the one-dimensional or two dimensional encoding scheme may be used. Also the 'bit-map encoding' scheme defined in CCITT Rec. T.417 | ISO/IEC 8613-7 may be used. All these forms of encoding may be used in a single document and all are basic. 'Uncompressed' mode of encoding may also be used but as a non-basic feature.

When using the T.4 or T.6 encoding method, the relationship between the order of pels and the order of bits in the octets in the coded data stream shall be such that the first pel in the order of bits is allocated to the least significant bit of an octet. In the case of bit-map encoding, the order of encoding shall be that the first pel is allocated to the most significant bit of an octet.

In a content portion, if content information is specified then it is required that the coding attribute "number of pels per line" is specified; the coding attribute "number of lines" may also be specified. No restriction is placed on the values that may be specified for these coding attributes. Thus this Profile places no restriction of the size of the pel arrays that may be used.

The type of encoding method used is specified by the attribute "type of coding". The use of this attribute is non-mandatory. Thus, if this attribute is not specified for a particular content portion and if the content architecture class specified corresponds to the formatted processible raster graphics content architecture class, then the default encoding method is assumed to be that defined in CCITT Recommendation T.6.

6.5.2.4 Pel path and line progression

The pel path and line progression provided are 0° and 270° respectively. This Profile does not allow the specification of other values.

6.5.2.5 Clipping

A sub-region within a pel array represented by a content portion associated with a basic component may be defined using the presentation attribute "clipping". No restriction is placed on the use of this attribute.

6.5.2.6 Pel spacing

The pel spacing is the distance in BMUs between any two pels on a line when a pel array is imaged. Any value may be explicitly specified provided that the spacing between pels is not less than 1 BMU. The pel spacing need not be an integer value. Also, the value 'null' may be specified, indicating that the scalable layout process is to be used.

The specification of the value 'null' or spacings of 16, 12, 8, 6, 5, 4, 3, 2, and 1 BMU between adjacent pels are basic. The specification of any other spacing is non-basic and shall be indicated in the document profile.

The pel spacing applicable to content associated with basic logical components is specified by the presentation attribute "pel spacing".

NOTES

1 The basic pel spacing values listed above are equivalent to resolutions of 75, 100, 150, 200, 240, 300, 400, 600 and 1200 pels per 25.4mm respectively when the document is imaged in its specified size.

2 The attribute "pel spacing" specifies two integers, the ratio of which determines the pels spacing. No restriction is placed on the values of these integers.

6.5.2.7 Spacing ratio

The spacing ratio is the ratio between the pel spacing and the line spacing when a pel array is imaged. This ratio is used to determine the line spacing from the pel spacing specified.

No restrictions are placed on the value of this ratio providing that the resultant line spacing is not less than 1 BMU. Also, the line spacing need not be an integral number of BMUs. All values are basic.

The default value may be specified in the document profile. If no default value is explicitly specified then the default value is the ratio 1:1, that is, the line spacing is equal to the pel spacing.

The spacing ratio applicable to the content associated with a basic logical component is specified by the presentation attribute "spacing ratio".

6.5.2.8 Image dimensions

The image dimensions are the constraints to be applied to the size of the image produced when laying out a pel array represented by a content portion associated with a basic logical component.

These constraints are specified for basic logical components by the presentation attribute "image dimensions". The value of this attribute is only taken into account if the value of the attribute "pel spacing" is 'null'.

6.5.3 Geometric graphics content

A document may contain graphic images composed of geometric graphic content encoded as CGM metafiles in accordance with ISO/IEC 8632. Each CGM Figure shall consist of a single picture only. Each GCM Figure may specify its minimum dimensions.

Further information concerning the specification of geometric graphics content information is given in Annex B.

6.6 Miscellaneous features

6.6.1 Resource-documents

Object classes of the types BodyText, BodyRaster and BodyGeometric, CommonText, CommonRaster, CommonGeometric and GenericBlock may refer to corresponding constituents in a resource-document.

The constituents in the resource-document may refer to content portions and to layout and presentation styles that are contained within the resource-document.

The constituents listed above are the only ones that are allowed to be referenced from another document via the attribute "resource"; however, a generic-document used as a resource-document may contain any combination of generic constituents which is conformant to this document application profile.

6.6.2 External documents

In the case of processable and formatted processable, the generic logical structure, and the generic layout structure if present, may be contained in an external document. Note that it is not permitted to exchange one generic structure in the interchanged document whilst referencing the other through the external document.

6.6.3 Border

Border may be specified for all the frames defined in 6.3.5 and 6.3.6 using the attribute "border". Borders may also be specified in presentation styles. All the features of border specified in CCITT Rec. T.412 | ISO/IEC 8613-2, may be specified. The use of border is a non-basic feature and shall be indicated in the document profile. Border shall not be specified for the constituents GenericBlock and SpecificBlock.

6.6.4 Application comments

Specification of the attribute "application comments" is mandatory for all object classes contained in a document that conforms to this Profile. Specification of this attribute is mandatory for all objects that do not refer to an object class. Specification of this attribute is optional for all objects that refer to object classes.

This attribute is structured so that it contains two fields. The first field is mandatory when the attribute is specified and contains a numeric string which uniquely identifies the constituent constraint applicable to the constituent for which the attribute is specified. This facilitates the processing of documents. A list of these identifiers is given in Table 2.

NOTES

1 The values of the constituent constraint numeric identifiers are not unique between the logical and layout structures and therefore in order to identify the constituent constraint applicable to a constituent it is necessary to know the structure of which the constituent is a part.

2 For constituent constraints that correspond to each other between the hierarchically related Profiles to which this Profile belongs, the same constituent constraint numeric identifier is specified.

The second field is optional and may contain any information that is relevant to the application or user. The format of the second field is not defined in this Profile and the interpretation of this field depends upon a private agreement between the originator and recipient of the document.

The encoding of the attribute "application comments" is defined in 8.1.3. and 8.2.3.

6.6.5 Alternative representation

The content information in a content portion may be replaced by a string of characters specified in the attribute "alternative representation". This attribute may be specified in content portions that contain character, raster graphics or geometric graphics content.

The specification and use of this attribute is optional. The string of characters specified shall belong to the character repertoires indicated in the document profile attribute "alternative representation character sets" (see 6.7.4.3). If the latter attribute is not explicitly specified in the document profile, then the default defined in CCITT Rec. T.411 | ISO/IEC 8613-1 is assumed. The control functions SP, CR and LF may also be used within the character string but no other control function is allowed; hence graphic character sets cannot be changed within the alternative representation.

TABLE 2/T.505

List of number string identifiers

Logical constituent constraints	Constituent constraint numeric identifiers
DocumentLogicalRoot	0
Passage	1
NumberedSegment	2
Number	3
Paragraph	6
Footnote	8
FootnoteNumber	9
FootnoteReference	10
FootnoteBody	11
FootnoteText	12
BodyText	14
BodyRaster	17
BodyGeometric	18
CommonContent	19
CommonText	20
CommonRaster	21
CommonGeometric	22
PageNumber	40
Layout constituent constraints	**Constituent constraint numeric identifiers**
DocumentLayoutRoot	0
PageSet	1
Page	2
RectoPage	3
VersoPage	4
CompositeHeader	5
VariableCompositeBody	7
ColumnFixed	8
ColumnVariable	9
SnakingColumns	10
SynchronizedColumns	11
BasicFloat	12
FootnoteArea	15
ArrangedContentFixed	16
ArrangedContentVariable	17
SourcedContentFixed	18
SourcedContentVariable	19
BasicHeader	27
BasicBody	28
GenericBlock	29
SpecificBlock	30
CompositeFooter	32
BasicFooter	33

6.6.6 Automatic numbering mechanisms

6.6.6.1 Page Numbering

As described in 6.2.4.3, the constituent constraint PageNumber contains a content generator which may refer to a page number. This content generator is evaluated when the document is laid out and this mechanism provides a means of reproducing the appropriate number of each page of a document.

The content generator has the following format:

<string-literal><num-expr><string-literal>

The format of this content generator is defined in the macro PGNUMBER (see Note).

The <string-literal> fields are optional and are pre-defined character strings. The basic character repertoire used to specify these strings is the ISO-IR 6 (the IRV of ISO/IEC 646). Any other character repertoire, and sub-repertoire if appropriate, may be used provided that it is designated and invoked by the appropriate code designation and invocation

sequences, and indicated in the document profile as a non-basic value. SP and no other control functions may be used in these strings.

The field <num-expr> is a reference to a binding PGnum which specifies the number of the page concerned. This binding is initialised at the document layout root or page set level (see the macro INITIALISEPGNUM in 7.4.1) and automatically incremented on each successive page (see macro PAGENUMBER in 7.4.1). By placing initialisation on the layout root, rather than on the pageset class(es), the pagenumbering may be defined to be continued from one pageset to the next.

The content associated with logical object classes of the type PageNumber is laid out in a frame of one of the following types: BasicHeader, BasicFooter, SourcedContentVariable, SourcedContentFixed (see 6.3.6) using the logical source mechanism. Thus when the appropriate frame is being laid out, the field <num-expr> in the content generator contained in a logical object class of the type PageNumber is evaluated and this determines the value of the binding PGnum that is associated with the current page being laid out.

The number associated with the binding PGnum is applied to a string function during its evaluation in order to convert the number into a character string. This enables the number to be represented in the form of an Arabic numeric string, an upper or lower case Roman numeric string or an upper or lower case alphabetic string.

Each page class may refer to a different instance of logical object classes of the type PageNumber and this allows different page numbering formats to be used for different parts of the document.

An example of page numbering is ‘Page X’ which consists of two concatenated character strings. The first is the literal character string ‘Page’ and this is concatenated to a string function denoted by ‘X’. When ‘X’ is evaluated in a particular instance it may, for example, return the character string ‘iv’, the Roman numeral (lower case) for the number ‘4’.

6.6.6.2 Segment numbering

As described in 6.2.3.4, the constituent constraint Number contains a content generator which, when evaluated during the layout process, produces a character string that serves to identify the NumberedSegment to which the Number belongs.

The format of this character string is as follows:

<pre-str><num-str><suf-str>

This format is defined in the macro SEGMENTNUMBER (see Note).

The fields <pre-str> and <suf-str> are optional prefix and suffix character strings respectively which may be of any length. The basic character repertoire used to specify these strings is the primary character repertoire of ISO 8859-1. Any other character repertoire, and sub-repertoire if appropriate, may be used provided that it is designated and invoked by the appropriate designation and innovation sequences and indicated in the document profile as a non-basic value. No other control functions may be used in these strings.

The field <num-str> is the segment identifier which consists of a single number or a sequence of two or more numbers, each of which is separated by a separator. The separator is a character string and may, for example, consist of a full stop or space. An example of a segment identifier is “6.3.4.2.1”. Thus segment identifiers have the general form:

<number>[<separator><number>]...

where [..]... indicates optional repetition.

In a document, the prefix and suffix character strings are string literals or carried by the bindings ‘prefix -<n>’ and ‘suffix -<n> respectively. The separator character strings are carried by bindings of the form <separator-<n>’ and the segment identifier <num-str> is carried by the binding ‘numberstring-<n>’.

In all these bindings ‘<n>’ is a sequence of one or more digits which indicate the depth of numbering, such that n = 1 indicates the number (prefix, suffix, numberstring etc) for the numbered segments immediately subordinate to a Passage, n = 2 indicates the number (prefix, suffix, separator etc) for the numbered segments immediately subordinate to the first level of numbered segments, and so on. The level number shall be indicated using the smallest possible number of characters, that is, there shall be no leading zeroes.

These bindings may be initialised at the document logical root, passage or at any numbered segment level to start the numbering scheme sequence at a subordinate level of numbered segment. They may also be re-specified at any level within the numbering scheme. The initialisation of bindings is specified by the macro INITIALISEANY.

The placement of bindings initialisations for numbering schemes is significant. Initial values for numbers-n bindings shall be placed either at the Passage level, or on the NumberedSegment class which is superior to that in which the binding will be referenced. Similarly, prefix and suffixes and separators shall be initialised either at the Passage or at the

immediately superior NumberedSegment level to their use. In particular note the prefix and suffix are not inherited by lower levels in hierachy (since they belong to the content generator SEGMENTNUMBERS rather than the binding Numberstrings-n). Thus to have concatenation to say '(1).a', lower level shall have a prefix of '(' and separator of').'.

In order to evaluate the value of 'numberstring-<n>' for each numbered segment, a number is assigned to each numbered segment at a given level. If the numbered segments are all of the same class then this number may be determined by the ORDINAL numeric function. If they are of different classes, then the number is carried by a binding of the form 'number-<n>'.

A different binding of the type 'number-<n>' is used for each numbered segment level and is initialised at a higher level constituent than the one in which it is used. The number associated with each numbered segment level is automatically incremented for each successive numbered segment (see the macro USENUMBERS).

The binding 'numberstring-<n>' that is applicable to a given level of numbered segment is now constructed as follows:

<numberstring-x><separator-y><number-z>

Hence, the segment identifier consists of a concatenation of up to three fields. The field <numberstring-x> is a reference to the segment identifier applicable to the immediately superior level of numbered segment (if any). This identifier is in the form of a character string. The field <separator-y> is a reference to a separator defined at some higher level in the document structure.

The field <number-z> is the number applicable to the given numbered segment whose identifier is being constructed. As indicated above, this number may be determined from an ORDINAL expression or by reference to a binding of the form 'number-<n>' which is specified for the same numbered segment whose identifier is being constructed. In either case, a string function is applied to the number to convert it into a character string. This string function allows the number to be represented in one of the following forms: Arabic number string, upper or lower case Roman numeral string, or upper or lower case alphabetic characters. This construction is defined in the macro USENUMBERSTRINGS.

The constructed binding of the form 'numberstring-<n>' is then available for constructing the identifiers at lower levels of numbered segments. This binding is also referred to in a content generator carried by the Number, which causes the identifier (with optional prefix and suffix strings) to be generated and reproduced when the document is laid out.

NOTE – The macros referred to in this subclause are defined in 7.3.1.

6.6.6.3 Footnote numbering

A footnote number is a character string that identifies a given footnote. The format of this string is as follows:

<string-literal><num-str><string-literal>

This format is defined in the macro FNOTENUMBER.

The <string-literal> fields are optional and are pre-defined prefix or suffix character strings. The basic character repertoire used to specify these strings is the primary character repertoire of ISO 8859-1. Any other character repertoire, and sub-repertoire if appropriate, may be used provided that it is designated and invoked by the appropriate designation and innovation sequences, and indicated in the document profile as a non-basic value. No other control functions may be used in these strings.

The field <num-str> is an automatically generated numeral or a user supplied character string that generally serves to identify a particular footnote. Numerals may be represented in the form of Arabic numerals, upper or lower case Roman numerals or upper or lower case Alphabetic characters. Automatically generated footnote numbers are incremented sequentially from an initial value which may be set to any positive value at the beginning of the document and reset at any passage.

A single binding 'fnotenumber' is provide to represent footnote numbers. This may be initialised to any non-negative number at the logical root or on any Passage (see specification of the macro INITIALISEFNOTE).

The footnote number is incremented using a binding expression at each footnote object (see the macro INCFNOTENUMBER). This is then made into a character string using a string function. This value is assigned to the binding 'fnotestring' (see the macro FNOTENUMBERSTRING).

Alternatively, a character string literal may be assigned to the binding 'fnotestring'; this provides the user with the ability to supply particular footnote labels for individual footnotes (see the macro FNOTESTRINGLITERAL).

The constituents FootnoteReference and FootnoteNumber contain content generators whose format is defined by the macro FNOTENUMBER. As indicated above, this format consists of a field represented by <num-str> which has optional prefix and suffix string literals. The field <num-str> consists of a reference to a binding 'fnotestring' which specifies the number of the footnote in the form of a character string.

6.6.7 User readable comments

Information which is to be interpreted as comments relevant to constituents and associated content portions may be specified using the attribute "user readable comments". This information is intended for presentation to humans.

The information consists of a string of characters which shall belong to one of the character repertoires indicated in the document profile attribute "comments character sets" (see 6.7.4.2). If the latter attribute is not explicitly specified, then the default defined in CCITT Rec. T.411 | ISO/IEC 8613-1 is assumed. The control functions CR, LF, SP and code extension control functions may also be used within the character string but no other control functions are allowed.

6.6.8 User visible name

Information which may be used to identify constituents within a document may be specified using the attribute "user visible name". This information is intended for presentation to humans, for example, to assist in the editing of documents.

The information consists of a string of characters which shall belong to one of the character repertoires indicated in the document profile attribute "comments character sets" (see 6.7.4.2). If the latter attribute is not explicitly specified, then the default defined in CCITT Rec. T.411 | ISO/IEC 8613-1 is assumed. The control functions CR, LF, SP and code extension control functions may also be used within the character string but no other control functions are allowed.

6.7 Document management features

Information relating to the document as a whole is specified in the document profile which is represented by the constituent constraint *DocumentProfile*. This constituent shall be specified in every document.

The information in the document profile is classified into the following categories:

- document constituent information;
- document identification information;
- document default information;
- non-basic characteristics information;
- document management information.

The information in the document profile may be of interest to the user or may be used for machine processing of the document.

6.7.1 Document constituent information

This information specifies which constituents are used to represent the document, including constituents that are external to the interchanged document. This information is divided into three categories.

6.7.1.1 Presence of document constituents

This information indicates which constituents are included in the document. That is, this information indicates whether or not the document contains a generic logical structure, a specific logical structure, a generic layout structure, a specific layout structure, layout styles and presentation styles (see Note). It is mandatory to specify this information in the document profile.

NOTE – If the generic logical or layout structure is external to the document (see 6.7.1.3), then it is still necessary to indicate that these structures are present and form part of the document.

6.7.1.2 Resource-document information

This information consists of a reference to a resource-document (see 6.6.1). This is specified by the attribute "resource-document". If constituents in the document contain references to object classes in a resource-document, then it is mandatory to specify this information in the document profile.

6.7.1.3 External document information

This information consists of a reference to an external document which may consist of a generic logical structure, or both generic layout and generic logical structures (see 6.6.2). If such a reference is required, then it is specified by the attribute "external document class" in the document profile.

6.7.2 Document identification information

This information relates to the identification of the document. This information is divided into six categories.

6.7.2.1 Document application profile information

This information indicates the document application profile to which the document belongs. It is mandatory to specify this information using the attribute "document application profile".

6.7.2.2 Document architecture class information

This information indicates the document architecture class to which the document belongs (see 6.1). It is mandatory to specify this information using the attribute "document architecture class".

6.7.2.3 Content architecture classes information

This information indicates the content architecture classes used in the document (see 6.5.1.2, 6.5.2.2 and 6.5.3). It is mandatory to specify this information using the attribute "content architecture classes".

6.7.2.4 Interchange format class information

This information indicates the interchange format class used to represent the document (see clause 8). It is mandatory to specify this information using the attribute "interchange format class".

6.7.2.5 ODA version information

This information indicates the International Standard or CCITT/ITU-T Recommendation to which the document conforms. It also specifies a calendar date, which indicates that the document conforms to the version of the International Standard or CCITT/ITU-T Recommendation and any addenda that are current on that date. It is mandatory to specify this information using the attribute "ODA version".

6.7.2.6 Document reference

This information serves to identify the document. Typically this information is allocated to the document by the creator of the document. The identifier may consist of an ASN.1 object identifier or a string of characters. It is mandatory to specify this information using the attribute "document reference".

6.7.3 Document default information

This information specifies various default values for attributes used in the document. The default values that are allowed are specified in the various subclauses of clause 6. The specification of this information is only required when it is required to specify a default value which is other than the standard default value specified in CCITT Rec. T.411 | ISO/IEC 8613-1.

Default values for the following groups of attributes may be specified:

- document architecture attributes;
- character content attributes;
- raster graphics attributes;
- geometric graphics attributes.

6.7.4 Non-basic characteristics information

This information specifies the non-basic attribute values specified in the document. It is mandatory to specify a non-basic attribute value in the document profile when such a value is used in the document.

The following types of non-basic attribute values may be specified:

- comments character sets;
- alternative representation character sets;
- page dimensions;
- medium-types;
- layout paths;
- borders;
- character presentation features;
- raster graphics presentation features;
- raster graphics coding attributes.

Further information concerning document profile, comments and alternative representation character sets is given below.

6.7.4.1 Profile character sets

Some document profile attributes have values consisting of character strings, for example, the document management attributes. The character sets used in these character strings are specified by the document profile attribute "profile character sets".

This attribute "profile character sets" specifies a code extension announcer and designations of character sets, which are subject to the following restrictions:

- The code extension announcer shall be 04/03 when specified. This code extension announcer specifies the use of G0 and G1 sets in an 8-bit environment and also the invocation of G0 and G1 sets into GL and GR respectively. Thus, in each attribute to which this attribute applies, invocation shift functions are not necessary, because G0 and G1 sets are implicitly invoked by this code extension announcer.
- G0 set: only ISO-IR 6 (the IRV of ISO/IEC 646), ISO-IR 2 (the primary set of ISO 6937), or any other version of ISO/IEC 646 may be designated for this set; these graphic character sets are implicitly invoked in GL.
- G1: no restrictions are placed on the graphic character sets that may be designated for this set; these graphic character sets are implicitly invoked in GR.
- Profile character sets.
- The empty set shall be designated into G1 and invoked into GR if no other specific character set is invoked in GR.

If the attribute "profile character sets" is not specified, then the default defined in CCITT Rec. T.411 | ISO/IEC 8613-1 is assumed.

6.7.4.2 Comments character sets

The character sets assumed to have been designated and optionally invoked at the beginning of the character strings specified by the attributes "user readable comments" (see 6.6.7) and "user visible name" (see 6.6.8) are specified using the document profile attribute "comments character sets".

It also specifies the code extension techniques and the graphic character sets which may be used in the attributes "user readable comments" and "user visible name".

If this attribute is specified, the code extension techniques which may be used in the attributes "user readable comments" and "user visible name" shall be announced by appropriate code extension announcers. The use of G0 set and GL shall always be announced. Other code extension announcers are to be specified according to the requirements of a particular document.

Two kinds of code extension techniques are permitted for this attribute. One is to use GL and GR without shift functions, and the other is to use various character sets by shift functions. The former is rather restricted but no shift functions are necessary in the "user readable comments" and "user visible name". The same restriction as in 6.7.4.1 is applied in this case.

The latter permits various usages of character sets but invocations shall be specified by shift functions in the "user readable comment" and "user visible name". The same restriction as in 6.5.1.4 is applied in this case.

All the graphic character sets which may be used in the attributes "user readable comments" and "user visible name" shall be designated in the "comments character sets".

There are no restrictions concerning the number of graphic character sets which are designated and/or invoked in the "comments character sets"; hence designation to the same G set overrides the previous G set.

If the attribute "comments character sets" is not specified, then the default defined in CCITT Rec. T.411 | ISO/IEC 8613-1 is assumed.

6.7.4.3 Alternative representation character sets

This attribute specifies the graphic character sets designated and invoked at the beginning of the attribute "alternative representation" other than the standard default graphic character sets.

The restriction on profile character sets described in 6.7.4.1 is also applied. If this attribute is not explicitly specified in the document profile, the default defined in CCITT Rec. T.411 | ISO/IEC 8613-1 is assumed.

6.7.5 Fonts list

This information specifies all the fonts (if any) used in the document. It is specified using the attribute "fonts list" (see B.2).

6.7.6 Document management attributes

Document management attributes contain information about the content of the document and its purpose. Information relating to the following may be specified:

- document description (see the Note below);
- dates and times;
- originators;
- other user information;
- external references;
- local file references;
- content attributes;
- security information.

The attributes that may be used to specify this information are defined in CCITT Rec. T.414 | ISO/IEC 8613-4.

The string of characters used in the document management attributes shall belong to the character sets indicated in the document profile attribute "profile character sets" (see 6.7.4.1). If the latter attribute is not explicitly specified in the document profile, then the default character set is the minimum sub-repertoire of ISO 6937.

The control functions SP, CR and LF may also be used within the character strings but no other control functions are allowed. Hence the graphic character set cannot be changed in the document management attributes.

NOTE – The document description includes the specification of the document reference (see 6.7.2.6).

7 Specification of constituent constraints

This clause specifies the definition of the constituent constraints which may be represented by data streams conforming to this Profile.

7.1 Introduction

The structure diagrams illustrating the relationships between the constituents in the logical structures are shown in 7.1.1 (see Figures 13 to 15). The macros indicated on these diagrams are defined in 7.3.1. These macros define the permissible values for the attribute "generator for subordinates" that are applicable to the constituents and define the allowed structures that are provided by this Profile.

The structure diagrams illustrating the layout structures are shown in 7.1.2 (see Figures 16 to 18). The macros indicated in these diagrams are defined in 7.4.1.

7.1.1 Diagrams of relationships of logical constituents

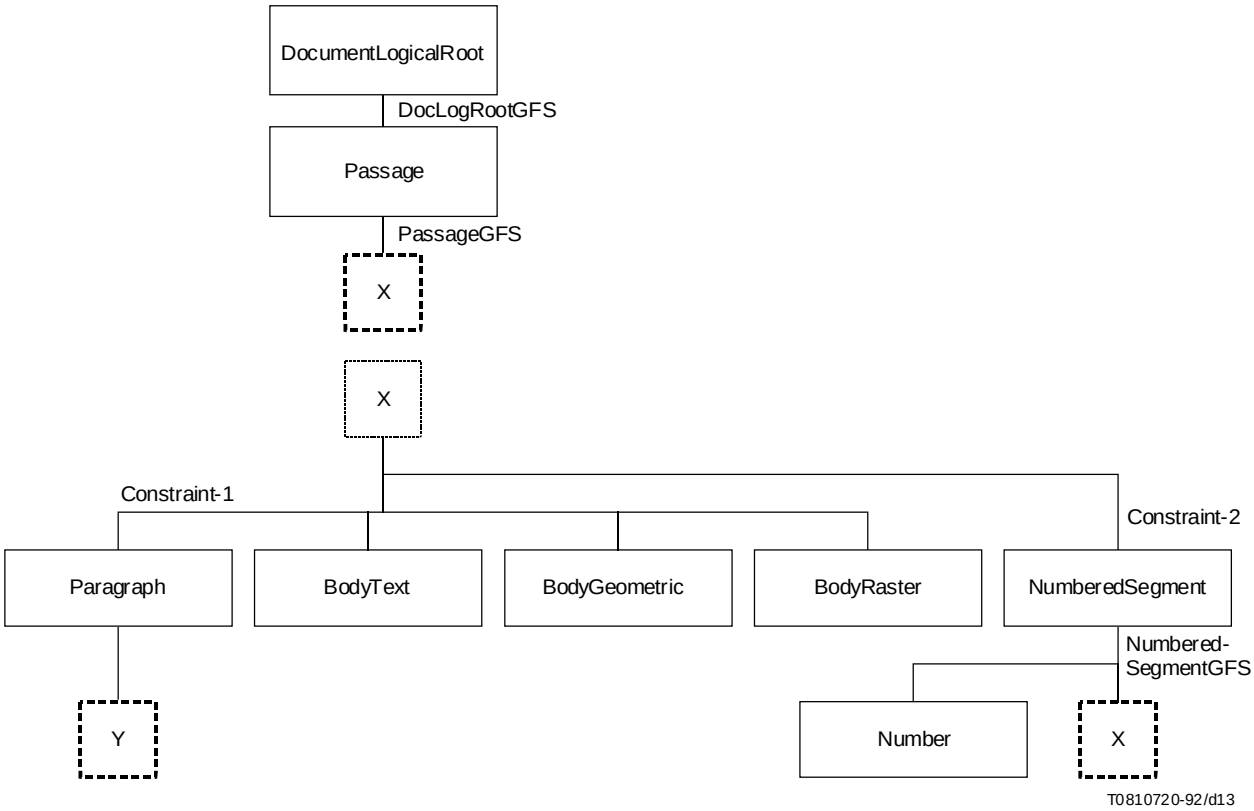


FIGURE 13/T.505

The body part of the generic logical structure – The passage and numbered segment levels

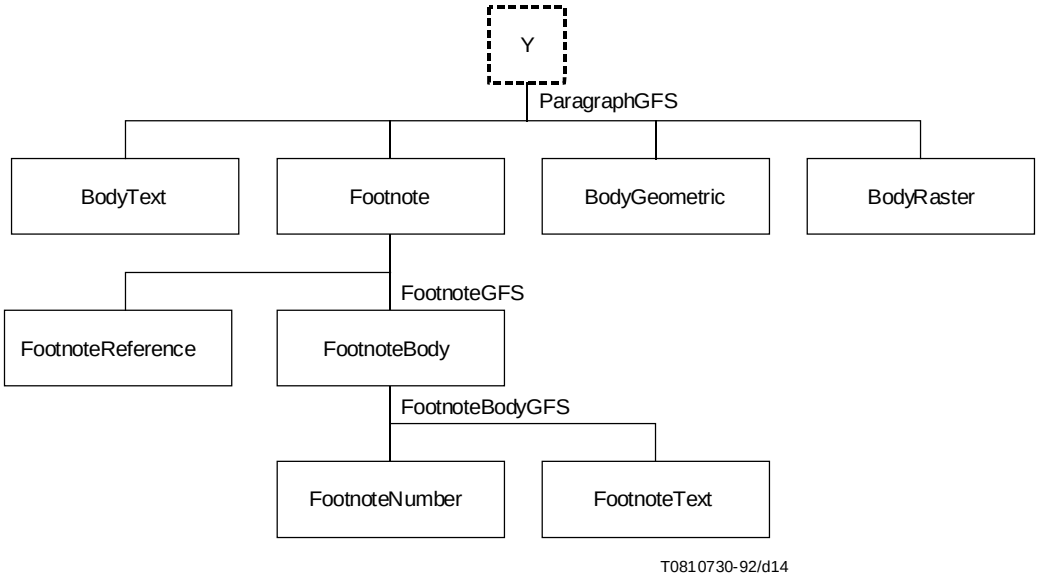


FIGURE 14/T.505

The body part of the generic logical structure – The paragraph level

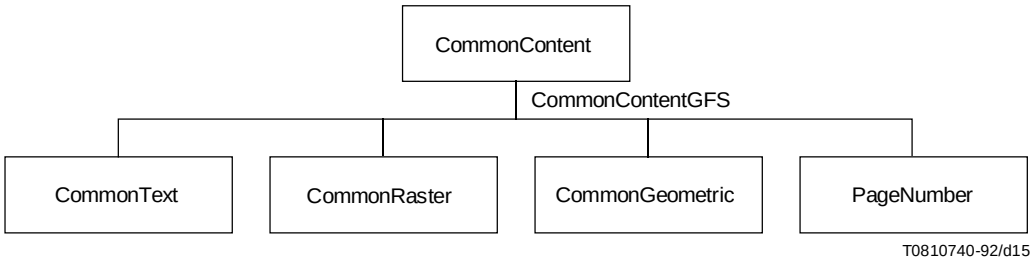


FIGURE 15/T.505

The common part of the generic logical structure

7.1.2 Diagrams of relationships of layout constituents

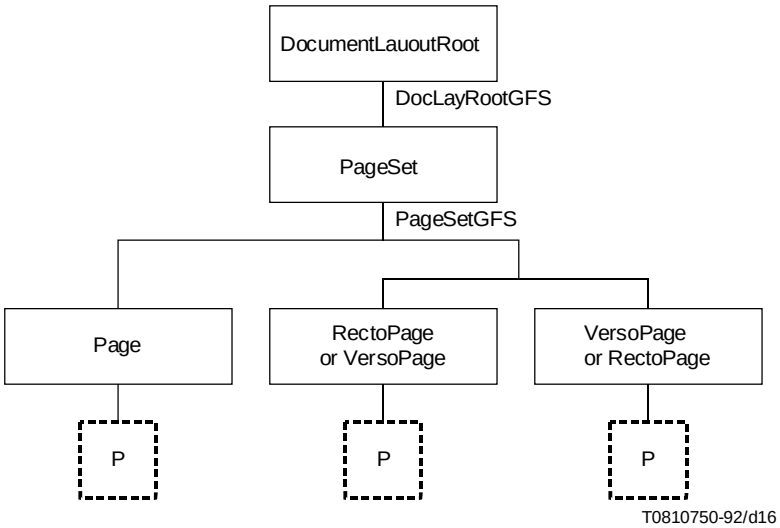


FIGURE 16/T.505

The layout structure – Document root and page sets

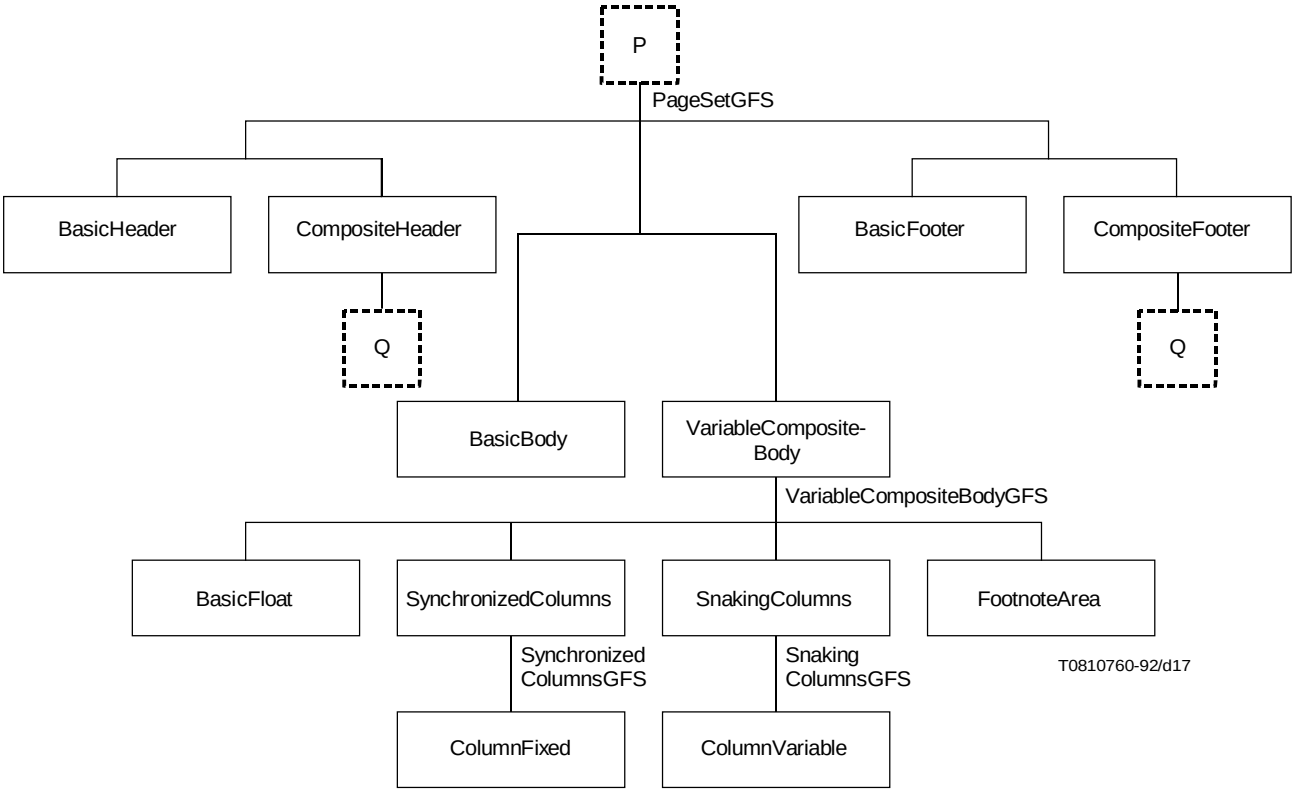


FIGURE 17/T.505

The layout structure – The page structure

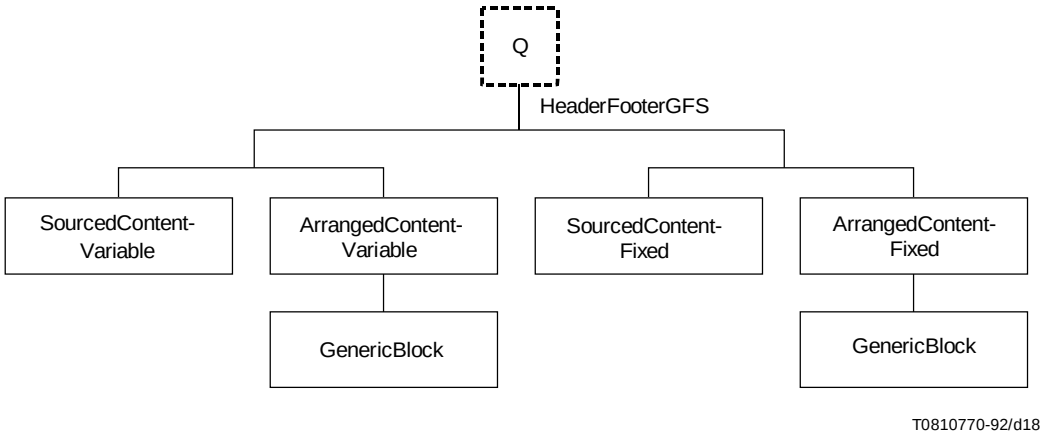


FIGURE 18/T.505

The layout structure – The header and footer frame structure

7.1.3 Notation

This subclause is written in accordance with the Document Application Profile Proforma and Notation (DAPPN) of CCITT Rec. T.411 | ISO/IEC 8613-1, Annex F. The following clarifications and minor extensions apply:

a) [Clarification]

The value range definition for the attributes “subordinates” and “imaging order” specify the set of object instances that may occur. The ordering and number (which may be zero) of object instances for the attribute “subordinates” shall be in accordance with the value of the attribute “generator for subordinates” in the respective object class.

b) [Clarification]

The value 'ANY_STRING' may include code extension control functions as well as graphic characters.

c) [Extension]

In order to write the specification of the usage of character sets and code extension control functions precisely, the following extensions are applied:

1) Table 3 defines the symbols that are introduced to denote shift functions.

TABLE 3/T.505

Symbols to denote shift functions

Symbol	Shift function	Coded representation
LS0	Locking shift zero	00/15
LS1R	Locking shift one right	ESC 07/14
LS2R	Locking shift two right	ESC 07/13
LS3R	Locking shift three right	ESC 07/12
SS2	Single shift two	08/14
SS3	Single shift three	08/15

2) <escape-sequence> is extended to include shift functions:

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

<invocation-control-function>::='LS0'|'LS1R'|'LS2R'|'LS3R'|'SS2'|'SS3';

3) Data type specification for #ESC in content information is extended as:

<escape-sequence>...

d) [Clarification]

When an attribute value is specified by a set of production rules, a non-terminal symbol which occurs first is its start symbol. Note that start symbols other than <object-id-expr>, <string-expr> and <construction-expr> are used.

e) [Extension]

Data type specifications other than those specified in the tables in DAPPN are applied for some attributes within the range that the base standards permit.

f) [Extension]

'|' is used in CASE SUPERIOR expressions in the following format in order to shorten the text:

CASE SUPERIOR ({const1|const2|...|constn}(aaaa)) OF {.....}

where “const1, const2, ... , constn” are names of constituent constraints, and “aaaa” is the name of an attribute.

This expression is equivalent to the following expression:

CASE SUPERIOR (const1(aaaa)) OF {.....}

CASE SUPERIOR (const2(aaaa)) OF {.....}

.....

CASE SUPERIOR (constn(aaaa)) OF {.....}

When CASE SUPERIOR is evaluated, constituents are searched from the immediate superior to the root. Only the first one which satifies one of the constituent constraints const1, const2, ... and constn is selected, and the attribute “aaaa” in it is tested.

7.2 Document profile constituent constraints

7.2.1 Macro definitions

```
DEFINE(FC, "ASN.1{2 8 2 6 0}" -- formatted character content --)
DEFINE(PC, "ASN.1{2 8 2 6 1}" -- processable character content --)
DEFINE(FPC, "ASN.1{2 8 2 6 2}" -- formatted processable character content --)
DEFINE(FPR, "ASN.1{2 8 2 7 2}" -- formatted processable raster graphics content --)
DEFINE(FPG, "ASN.1{2 8 2 8 0}" -- formatted processable geometric graphics content --)
DEFINE(FDA, "{’formatted’}")
DEFINE(PDA, "{’processable’}")
DEFINE(FPDA, "{’formatted-processable’}")
DEFINE(PDA-FPDA, "{’processable’|’formatted processable’}")
DEFINE(DAC, "DocumentProfile (Document-architecture-class)")

DEFINE(NominalPageSizes, "
        REQ #horizontal-dimension {7015},
            REQ #vertical-dimension {9920}        -- ISO A5 portrait --
       |REQ #horizontal-dimension {9920},
            REQ #vertical-dimension {7015}        -- ISO A5 landscape --
       |REQ #horizontal-dimension {9920},
            REQ #vertical-dimension {14030}       -- ISO A4 portrait --
       |REQ #horizontal-dimension {14030},
            REQ #vertical-dimension {9920}        -- ISO A4 landscape --
       |REQ #horizontal-dimension {14030},
            REQ #vertical-dimension {19840}       -- ISO A3 portrait --
       |REQ #horizontal-dimension {19840},
            REQ #vertical-dimension {14030}       -- ISO A3 landscape --
       |REQ #horizontal-dimension {12141},
            REQ #vertical-dimension {17196}       -- JIS B4 (Japanese legal) portrait --
       |REQ #horizontal-dimension {17196},
            REQ #vertical-dimension {12141}       -- JIS B4 (Japanese legal) landscape --
       |REQ #horizontal-dimension {8598},
            REQ #vertical-dimension {12141}       -- JIS B5 (Japanese letter) portrait --
       |REQ #horizontal-dimension {12141},
            REQ #vertical-dimension {8598}        -- JIS B5 (Japanese letter) landscape --
       |REQ #horizontal-dimension {10200},
            REQ #vertical-dimension {16800}       -- ANSI legal portrait --
       |REQ #horizontal-dimension {16800},
            REQ #vertical-dimension {10200}       -- ANSI legal landscape --
       |REQ #horizontal-dimension {10200},
            REQ #vertical-dimension {13200}       -- ANSI-A portrait --
       |REQ #horizontal-dimension {13200},
            REQ #vertical-dimension {10200}       -- ANSI-A landscape --
       |REQ #horizontal-dimension {13200},
            REQ #vertical-dimension {20400}       -- ANSI-B portrait --
       |REQ #horizontal-dimension {20400},
            REQ #vertical-dimension {13200}       -- ANSI-B landscape --
                ")

DEFINE(GRAPHICRENDITIONS,"
                {’cancel’|’increased-intensity’
                |’italicized’|’underlined’|’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’|’not-crossed-out’}...
                    ")
```

-- Macro defining permissible code extension announcers. Note that all the values are basic. --

DEFINE(CDEXTEN,"ESC 02/00 05/00, *-- LS0 --*
[ESC 02/00 05/03], *-- LSR1 --*
[ESC 02/00 05/05], *-- LSR2 --*
[ESC 02/00 05/07], *-- LSR3 --*
[ESC 02/00 05/10], *-- SS2 --*
[ESC 02/00 05/11] *-- SS3 --*
")

-- Macro defining code extension announcers for document application profile defaults --

DEFINE(DAP-DEFAULT-CDEXTEN, "$CDEXTEN")

-- Macros defining final character for designation --

DEFINE(FCORE, "04/02 *-- A final character designating ISO-IR 6 (the IRV of ISO/IEC 646, i.e ASCII) --***")**

DEFINE(F646, *"-- A final character designating any version of ISO/IEC 646 except, ISO-IR 6 --***")**

DEFINE(F94S, *"-- A final character designating any registered 94 single byte graphic character set, optionally preceded by one or more intermediate characters as defined in Annex C of ISO 2022 --***")**

DEFINE(F94M, *"-- A final character designating any registered 94 multi byte graphic character set, optionally preceded by one or more intermediate characters as defined in Annex C of ISO 2022 --***")**

DEFINE(F96S, *"-- A final character designating any registered 96 single byte graphic character set, optionally preceded by one or more intermediate characters as defined in Annex C of ISO 2022 --***")**

DEFINE(F96M, *"-- A final character designating any registered 96 multi byte graphic character set, optionally preceded by one or more intermediate characters as defined in Annex C of ISO 2022 --***")**

DEFINE(FEMPTY, "07/14 *-- The empty set --***")**

-- Macro defining a revision number of a character set --

DEFINE (REV,"-- *An octet between 04/00 and 07/14 which represents a revision number as defined in ISO 2022 --***")**

-- Macro defining designation sequences --

DEFINE(DEG-CORE-G0, "ESC 02/08 $FCORE")
-- Designate the 94 characters of ISO-IR 6 (the IRV of ISO/IEC 646) to G0 --

DEFINE(DEG-646-G0, "ESC 02/08 $F646")
-- Designate any version of ISO/IEC 646, except ISO-IR 6, to GO --

DEFINE(DEG-ANY-G1, "{ [ESC 02/06 $REV] {ESC 02/09 $F94S
|ESC 02/04 02/09 $F94M
|ESC 02/13 $F96S
|ESC 02/04 02/13 $F96M}}")
-- Designate any character set to G1 --

DEFINE(DEG-ANY-G2, "{ [ESC 02/06 $REV] {ESC 02/10 $F94S
|ESC 02/04 02/10 $F94M
|ESC 02/14 $F96S
|ESC 02/04 02/14 $F96M}}")
-- Designate any character set to G2 --

DEFINE(DEG-ANY-G3, "{ [ESC 02/06 $REV] {ESC 02/11 $F94S
|ESC 02/04 02/11 $F94M
|ESC 02/15 $F96S
|ESC 02/04 02/15 $F96M}}")
-- Designate any character set to G3 --

DEFINE(DEG-EMPTY-G1, "ESC 02/09 $FEMPTY")
-- Designate the empty set to G1 --

-- Macro defining permissible graphic character sets --

```
DEFINE(PERMIT-GRCHAR,  "          {$DEG-CORE-G0 LS0
                                  |$DEG-646-G0 LS0},
                                  {{$DEG-ANY-G1 LS1R
                                  |$DEG-ANY-G2 LS2R
                                  |$DEG-ANY-G3 LS3R}...
                                  |$DEG-EMPTY-G1 LS1R}  ")
```

-- Macro defining graphic character sets for document application profile defaults --

```
DEFINE(DAP-DEFAULT-GRCHAR, "$PERMIT-GRCHAR")
```

-- Macro defining basic character sets. Note that this macro is defined for clarification of the specification and is not -- used in any other part of this document application profile specification --

```
DEFINE(BASIC-GRCHAR, "            $DEG-CORE-G0 LS0,
                                  $DEG-EMPTY-G1 LS1R  ")
```

-- Macro defining non-basic graphic character sets --

```
DEFINE(NON-BASIC-GRCHAR,  "       {$DEG-646-G0
                                  |$DEG-ANY-G1
                                  |$DEG-ANY-G2
                                  |$DEG-ANY-G3}... ")
```

-- Macro defining character sets used in document profile attributes --

```
DEFINE(PROFCHAR, "
ESC 02/00 04/03                   -- announcement of use of G0 and G1, and invocation into GL and GR
                                  respectively. (No shift functions are necessary.) --
{$DEG-CORE-G0|$DEG-646-G0}        -- designate G0 --
{$DEG-ANY-G1|$DEG-EMPTY-G1}       -- designate G1 --
          ")
```

-- Macro defining comments character sets --

```
DEFINE(COMCHAR,"
-- in the case to use both GL and GR without shift functions --
ESC 02/00 04/03                   -- announcement of use of G0 and G1, and invocation into GL and GR
                                  respectively. (No shift functions are necessary.) --
{$DEG-CORE-G0|$DEG-646-G0}        -- designate G0 --
{$DEG-ANY-G1|$DEG-EMPTY-G1}       -- designate G1 --

-- in the case of use of various character sets (shift functions are necessary) --
{ESC 02/00 05/00,                 -- announcement to use G0 and LS0 --
[ESC 02/00 05/03],                -- announcement to use G1 and LS1R --
[ESC 02/00 05/05],                -- announcement to use G2 and LS2R --
[ESC 02/00 05/07],                -- announcement to use G3 and LS3R --
[ESC 02/00 05/10],                -- announcement to use G2 and SS2 --
[ESC 02/00 05/11]   }             -- announcement to use G3 and SS3 --

{$DEG-CORE-G0|$DEG-646-G0}        -- designate G0 --
{{$DEG-ANY-G1                     -- designate G1 --
|$DEG-ANY-G2                      -- designate G2 --
|$DEG-ANY-G3}                     -- designate G3 --
|$DEG-EMPTY-G1}
          ")
```

-- Macro defining character sets used for alternative representation --

```
DEFINE(ALTCHAR, "$PROFCHAR")
```

7.2.2 Constituents constraints

7.2.2.1 DocumentProfile {

CASE $DAC OF {

$FDA: **PERM Generic-layout-structure {'factor-set'},**
PERM Specific-layout-structure {'present'},
-- shall be present in the case of specific document --
-- and shall not be present in the case of generic document --
PERM Presentation-styles {'present'}

$PDA: **PERM Generic-layout-structure {'complete-generator-set'},**
PERM Generic-logical-structure {'complete-generator-set'
|'partial-generator-set'},
-- shall be present if there is no external document class reference --
PERM Specific-logical-structure {'present'},
-- shall be present in the case of complete document --
-- and shall not be present in the case of generic document --
PERM Presentation-styles {'present'},
PERM Layout-styles {'present'}

$FPDA: **PERM Generic-layout-structure {'complete-generator-set'},**
-- shall be present if there is no external document class reference --
PERM Specific-layout-structure {'present'},
-- shall be present in the case of complete document --
-- and shall not be present in the case of generic document --
PERM Generic-logical-structure {'complete-generator-set'
|'partial-generator-set'},
-- shall be present if there is no external document class reference --
PERM Specific-logical-structure {'present'},
-- shall be present in the case of complete document --
-- and shall not be present in the case of generic document --
PERM Presentation-styles {'present'},
PERM Layout-styles {'present'}
},

PERM External-document-class ANY_VALUE},

PERM Resource-document ANY_VALUE},

PERM Resources {MUL{REQ #resource-identifier {ANY_VALUE},
REQ #resource-object-class-identifier {ANY_VALUE}}},

-- document characteristics --

REQ Document-application-profile {-- *see clause 8 for a definition of the permitted values for this attribute* --**},**

PERM Document-application-profile-defaults {

CASE $DAC OF {
$FDA :{PERM #content-architecture-class {$FC|$FPC}}
$PDA :{PERM #content-architecture-class {$FC|$PC|$FPC}}
$FPDA :{PERM #content-architecture-class {$FC|$FPC}}
},

PERM #dimensions {REQ #horizontal-dimension
{REQ #fixed-dimension {<=14030}},
REQ #vertical-dimension
{REQ #fixed-dimension {<=19840}}
-- up to ISO A3 portrait --

|REQ #horizontal-dimension
{REQ #fixed-dimension {<=19840}},
REQ #vertical-dimension
{REQ #fixed-dimension {<=14030}}
-- up to ISO A3 landscape --

|REQ #horizontal-dimension
{REQ #fixed-dimension {<=13200}},
REQ #vertical-dimension

```
                                        {REQ #fixed-dimension {<=20400}}
                                            -- up to ANSI-B portrait --

                                        |REQ #horizontal-dimension
                                         {REQ #fixed-dimension {<=20400}},
                                        REQ #vertical-dimension
                                         {REQ #fixed-dimension {<=13200}}
                                            -- up to ANSI-B landscape --},

PERM  #medium-type                      {PERM #nominal-page-size{$NominalPageSizes},
                                         PERM #side-of-sheet {ANY_VALUE}},

PERM  #page-position                    {ANY_VALUE},

PERM  #layout-path                      {'0-degrees'|'180-degrees'|'270-degrees'},

PERM  #type-of-coding                   {ASN.1{2 8 3 6 0} -- character encoding --
                                        |ASN.1{2 8 3 7 0} -- T.6 encoding --
                                        |ASN.1{2 8 3 7 1} -- T.4 one dimensional encoding --
                                        |ASN.1{2 8 3 7 2} -- T.4 two dimensional encoding --
                                        |ASN.1{2 8 3 7 3} -- bitmap encoding --
                                        |ASN.1{2 8 3 8 0} -- geometric encoding --},

PERM #character-content-defaults  {
  PERM #alignment                       {ANY_VALUE},
  PERM #character-fonts                 {ANY_VALUE},
  PERM #character-path                  {ANY_VALUE},
  PERM #character-spacing               {ANY_VALUE},
  PERM #character-orientation           {'0-degrees' | '90-degrees'},
  PERM #code-extension-announcers       {$DAP-DEFAULT-CDEXTEN},
  PERM #first-line-offset               {ANY_VALUE},
  PERM #graphic-character-sets          {$DAP-DEFAULT-GRCHAR},
  PERM #graphic-character-sub-repertoire {ANY_VALUE},
  PERM #graphic-rendition               {$GRAPHICRENDITIONS},
  PERM #indentation                     {ANY_VALUE},
  PERM #initial-offset                  {ANY_VALUE},
  PERM #itemization                     {ANY_VALUE},
  PERM #kerning-offset                  {ANY_VALUE},
  PERM #line-layout-table               {ANY_VALUE},
  PERM #line-progression                {ANY_VALUE},
  PERM #line-spacing                    {ANY_VALUE},
  PERM #orphan-size                     {ANY_VALUE},
  PERM #proportional-line-spacing       {ANY_VALUE},
  PERM #widow-size                      {ANY_VALUE}},

PERM #raster-graphics-content-defaults  {
  PERM #image-dimensions                {ANY_VALUE},
  PERM #pel-spacing                     {ANY_VALUE},
  PERM #spacing-ratio                   {ANY_VALUE},
  PERM #compression                     {ANY_VALUE}}},

REQ     Document-architecture-class     {$FDA|$PDA|$FPDA},

REQ     Content-architecture-classes    {[$FC],[$PC],[$FPC],[$FPR],[$FPG]},

REQ     Interchange-format-class  {  -- see clause 8 for a definition of the permitted values for this attribute --},

REQ     Oda-version                     {REQ #standard-or-recommendation{"CCITT Rec. T.410
                                        Series(1988)|ISO 8613(1989); version 1.1"},
                                        REQ #publication-date{1992-01-01}},
  -- non basic document characteristics --

PERM    Profile-character-sets          {$PROFCHAR},

PERM    Comments-character-sets         {$COMCHAR},

PERM    Alternative-representation-character-sets        {$ALTCHAR},

PERM    Page-dimensions  {PMUL
                                        {REQ #horizontal-dimension
                                         {REQ #fixed-dimension {<=14030}},
                                        REQ #vertical-dimension
                                         {REQ #fixed-dimension {12401..19840}}
```

```
                                        |REQ #horizontal-dimension
                                          {REQ #fixed-dimension {9241..14030}},
                                        REQ #vertical-dimension
                                          {REQ #fixed-dimension {<=19840}}
                                              -- up to ISO A3 portrait --

                                        |REQ #horizontal-dimension
                                          {REQ #fixed-dimension {12401..19840}},
                                        REQ #vertical-dimension
                                          {REQ #fixed-dimension {<=14030}}
                                        |REQ #horizontal-dimension
                                          {REQ #fixed-dimension {<=19840}},
                                        REQ #vertical-dimension
                                          {REQ #fixed-dimension {9241..14030}}
                                              -- up to ISO A3 landscape --

                                        |REQ #horizontal-dimension
                                          {REQ #fixed-dimension {<=13200}},
                                        REQ #vertical-dimension
                                          {REQ #fixed-dimension {12401..20400}}
                                        |REQ #horizontal-dimension
                                          {REQ #fixed-dimension {9241..13200}},
                                        REQ #vertical-dimension
                                          {REQ #fixed-dimension {<=20400}}
                                              -- up to ANSI-B portrait --

                                        |REQ #horizontal-dimension
                                          {REQ #fixed-dimension {12401..20400}},
                                        REQ #vertical-dimension
                                          {REQ #fixed-dimension {<=13200}}
                                        |REQ #horizontal-dimension
                                          {REQ #fixed-dimension {<=20400}},
                                        REQ #vertical-dimension
                                          {REQ #fixed-dimension {9241..13200}}
                                              -- up to ANSI-B landscape -- }},
  -- any value of dimensions which is greater than the common assured reproduction area of ISO A4 and
  -- ANSI-A is non-basic --

PERM    Medium-types                    {PMUL
                                        {PERM #nominal-page-size{$NominalPageSizes},
                                         PERM #side-of-sheet{'recto'|'verso'}}},
  -- all values of "medium type" are non-basic --

PERM    Layout-paths {{'0-degrees'|'90-degrees'|'180-degrees'}...},

PERM    Borders                         {ANY_VALUE},

PERM    Coding-attributes {

PERM  #raster-graphics-coding-attributes {
    PERM #compression                   {'uncompressed'}}},

PERM  Presentation-features {
    PERM #character-presentation-features  {
       PERM #character-orientation      {'90-degrees'},
       PMUL {PERM #character-path       {'90-degrees'
                                        |'180-degrees'
                                        |'270-degrees'}},

    PMUL {PERM #character-spacing       {<100|100|160|200}},
    -- only values <100 are required to be specified.  Values 100, 160 200 need not be declared and are only
    -- permitted to be specified for upwards compatibility from PM-11 --

    PMUL {PERM #graphic-character-sets                  {$NON-BASIC-GRCHAR}},
    PMUL {PERM #graphic-character-sub-repertoire        {ANY_VALUE}},
    PMUL {PERM #graphic-rendition                       {'crossed-out' | 'not-crossed-out'}},
    -- values need not be declared and are only permitted to be specified for upwards compatibility from PM-11 --

    PMUL {PERM #line-spacing            {ANY_VALUE}
                                        EXCEPT{200,300,400}},
    -- value 150 need not be declared and is only permitted to be specified for upwards compatibility from PM-11 --
```

```
    PERM #line-progression                  {'90-degrees'}},

PERM #raster-graphics-presentation-features {
    PMUL {PERM #pel-spacing                 {ANY_VALUE}
                                            EXCEPT{16,12,8,6,5,4,3,2,1}}
          }
  -- Any value of #pel spaces is permitted as basic --
  -- Basic values of #length are multiples of #pel spaces as listed -- },

  -- additional document characteristics --

    PERM  Fonts-lis                         {PMUL{REQ #font-identifier {ANY_VALUE},
                                              REQ #font-reference {ANY_VALUE}}},


  -- document management attributes --

  -- document description --
PERM  Title                                 {ANY_STRING},
PERM  Subject                               {ANY_STRING},
PERM  Document-type                         {ANY_STRING},
PERM  Abstract                              {ANY_STRING},
PERM  Keywords                              {ANY_STRING...},
REQ   Document-reference                    {ANY_VALUE},


  -- dates and times --
PERM  Document-date-and-time                {ANY_STRING},
PERM  Creation-date-and-time                {ANY_STRING},
PERM  Local-filing-date-and-time            {ANY_VALUE},
PERM  Expiry-date-and-time                  {ANY_STRING},
PERM  Start-date-and-time                   {ANY_STRING},
PERM  Purge-date-and-time                   {ANY_STRING},
PERM  Release-date-and-time                 {ANY_STRING},
PERM  Revision-history                      {ANY_VALUE},


  -- originators --
PERM  Organizations                         {ANY_STRING...},
PERM  Owners                                {ANY_VALUE},
PERM  Preparers                             {ANY_VALUE},
PERM  Authors                               {ANY_VALUE},


  -- other user information --
PERM  Copyright                             {ANY_VALUE},
PERM  Status                                {ANY_STRING},
PERM  User-specific-codes                   {ANY_STRING...},
PERM  Distribution-list                     {ANY_VALUE},
PERM  Additional-information                {ANY_VALUE},
  -- external references --
PERM  References-to-other-documents         {ANY_VALUE},
PERM  Superseded-documents                  {ANY_VALUE},


  -- local file references --
PERM  Local-file-references                 {ANY_VALUE},


  -- content attributes --
PERM  Document-size                         {ANY_INTEGER},
PERM  Number-of-pages                       {ANY_INTEGER},
PERM  Languages                             {ANY_STRING...},


  -- security information --
PERM  Authorization                         {ANY_VALUE},
PERM  Security-classification               {ANY_STRING},
PERM  Access-rights                         {ANY_STRING...}}
```

7.3 Logical constituent constraints

7.3.1 Macro definitions

```
DEFINE(DocLogRootGFS,   "
<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>}...)
                                |CHO({<construction-term>}...);

<construction-factor>   ::=     OBJECT_CLASS_ID_OF(Passage)
                                |<construction-type>;
                        ")

DEFINE(CONSTRAINT-1,    "
<constraint-1>          ::=     <construction-term>
                                |<construction-type>;

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

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

<construction-factor>   ::=     OBJECT_CLASS_ID_OF(Paragraph)
                                |OBJECT_CLASS_ID_OF(BodyText)
                                |OBJECT_CLASS_ID_OF(BodyRaster)
                                |OBJECT_CLASS_ID_OF(BodyGeometric)
                                |<construction-type>;
                        ")

DEFINE(CONSTRAINT-2,    "
<constraint-2>          ::=     OBJECT_CLASS_ID_OF(NumberedSegment)
                                |OPT REP OBJECT_CLASS_ID_OF(NumberedSegment)
                                |REP OBJECT_CLASS_ID_OF(NumberedSegment)
                                |OPT OBJECT_CLASS_ID_OF(NumberedSegment);
                        ")

DEFINE(PassageGFS,      "
<construction-expr>     ::=     <constraint-1>
                                |<constraint-2>
                                |SEQ(<constraint-1><constraint-2>);

$CONSTRAINT-1
$CONSTRAINT-2           ")

DEFINE(NumberedSegmentGFS,  "
<construction-expr>     ::=     SEQ(<term-1>[<constraint-1>]
                                        [<constraint-2>]);

<term-1>                ::=     OBJECT_CLASS_ID_OF(Number);

$CONSTRAINT-1
$CONSTRAINT-2
                        ")

DEFINE(ParagraphGFS,    "
<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>}...)
                                |CHO({<construction-term>}...);
```

```
<construction-factor>        ::=    OBJECT_CLASS_ID_OF(BodyText)
                                    |OBJECT_CLASS_ID_OF(BodyRaster)
                                    |OBJECT_CLASS_ID_OF(BodyGeometric)
                                    |OBJECT_CLASS_ID_OF(Footnote)
                                    |<construction-type>;
                             ")

DEFINE(FootnoteGFS,          "
<construction-expr>          ::=    SEQ(OBJECT_CLASS_ID_OF(FootnoteReference)
                                    OBJECT_CLASS_ID_OF(FootnoteBody));
                             ")

DEFINE(FootnoteBodyGFS, "
<construction-expr>          ::=    SEQ(OBJECT_CLASS_ID_OF(FootnoteNumber)
                                        <term-1>);

<term-1>                     ::=    OBJECT_CLASS_ID_OF(FootnoteText)
                                    |REP OBJECT_CLASS_ID_OF(FootnoteText)
                                    |CHO({OBJECT_CLASS_ID_OF(FootnoteText)}...)
                                    |REP CHO({OBJECT_CLASS_ID_OF(FootnoteText)}...);
                             ")

DEFINE(CommonContentGFS, "
<construction-expr>          ::=    <construction-factor>
                                    |SEQ(<construction-factor>...);

<construction-factor>        ::=    OBJECT_CLASS_ID_OF(PageNumber)
                                    |OBJECT_CLASS_ID_OF(CommonText)
                                    |OBJECT_CLASS_ID_OF(CommonRaster)
                                    |OBJECT_CLASS_ID_OF(CommonGeometric);
                             ")

DEFINE(N,
  "<n>::={""0""|""1""|""2""|""3""|""4""|""5""|""6""|""7""|""8""|""9""}...;
  -- any string of characters from the set of characters: '0'...'9' --

-- Defines the prefix binding. This binding may be used to associate a string literal with an object or object class.
-- In addition, this binding is used to prefix text to another binding, such as a segment number, footnote number
-- or page number. The instances are differentiated by a suffix number. --
                             ")

DEFINE(PREFIX,     "
<prefix>                     ::=    ""prefix-""<n>;
$N
                             ")
-- Defines the suffix binding. This binding may be used to associate a string literal with an object or object class.
-- In addition, this binding is used to suffix text to another binding, such as a segment number, footnote number
-- or page number. The instances are differentiated by a suffix number. --

DEFINE(SUFFIX,     "
<suffix>                     ::=    ""suffix-""<n>;
$N
                             ")
-- Defines the separator binding.  This binding is used to provide a separator character
-- for a hierarchical form of a segment number, footnote number or page number.
-- The instances are differentiated by a suffix number. --

DEFINE(SEPARATOR,    "
<separator>                  ::=    ""separator-""<n>;
$N
                             ")
-- Defines the general number binding. This binding may be instanced for use as the numeric value
-- for use such as in segment number, footnote number or page number bindings.
-- The instances are differentiated by a suffix number. --

DEFINE(NUMBER,     "
<number>                     ::=    ""number-""<n>;
$N
                             ")
```

-- Defines the general number string binding. This binding may be instanced for use
-- as the string value such as for segment number, footnote number or page numbers.
-- The instances are differentiated by a suffix number. --

```
DEFINE(NUMBERSTRING, "
<numberstring>          ::=     ""numberstring-""<n>;
$N
                        ")
```

-- Used to initialise/specify any of the bindings. The bindings defined by this macro are permitted to:
– any logical object class
– any logical object
– any layout object class except frame classes and block classes, and
– any layout object except frames and blocks in the case of FPDA and FDA. --

```
DEFINE(INITIALISEANY, "
        REQ #binding-name{$PREFIX},
        REQ #binding-value{ANY_STRING}
        |REQ #binding-name{$SUFFIX},
        REQ #binding-value{ANY_STRING}
        |REQ #binding-name{$SEPARATOR},
        REQ #binding-value{ANY_STRING}
        |REQ #binding-name{$NUMBER},
        REQ #binding-value{>=0}
        |REQ #binding-name{$NUMBERSTRING},
        REQ #binding-value{ANY_STRING}
                        ")
```

-- Used to make a simple or compound string out of the number bindings. --

```
DEFINE(USENUMBERSTRINGS, "
        REQ #binding-name{$NUMBERSTRING},
        REQ #binding-value{
        <string-expr>::=<hierarchic-expr>|<simple-expr>;

<hierarchic-expr>       ::=     B_REF(SUP(CURR-OBJ))(<numberstring>)
                                |B_REF(SUP(CURR-OBJ))(<separator>)
                                |<simple-expr>;

<simple-expr>           ::=     MK-STR(B_REF(CURR-OBJ)(<number>))
                                |U-ALPHA(B_REF(CURR-OBJ)(<number>))
                                |L-ALPHA(B_REF(CURR-OBJ)(<number>))
                                |U-ROM(B_REF(CURR-OBJ)(<number>))
                                |L-ROM(B_REF(CURR-OBJ)(<number>))
                                |MK-STR(ORD(CURR-OBJ))
                                |U-ALPHA(ORD(CURR-OBJ))
                                |L-ALPHA(ORD(CURR-OBJ))
                                |U-ROM(ORD(CURR-OBJ))
                                |L-ROM(ORD(CURR-OBJ))
                                |ANY_STRING;

$NUMBERSTRING
$SEPARATOR
$NUMBER

}                       ")
```

-- Used to increment any of the number bindings. --

```
DEFINE(USENUMBERS, "
                                REQ #binding-name{$NUMBER},
                                REQ #binding-value
                                  {<num-expr>::=INC(B_REF(PREC(CURR-OBJ))(<number>));

$NUMBER}

                        ")
```

-- This string expression is allowed in a content generator for Number to automatically
-- generate text for segment numbers. --

```
DEFINE(SEGMENTNUMBER,   "
```

```
<string-expr>               ::=     [<pre-str>]<num-str>[<suf-str>];

<num-str>                   ::=     B_REF(SUP(CURR-OBJ))(<numberstring>);

<pre-str>                   ::=     B_REF(SUP(CURR-OBJ))(<prefix>)
                                    |ANY_STRING;

<suf-str>                   ::=     B_REF(SUP(CURR-OBJ))(<suffix>)
                                    |ANY_STRING;

$NUMBERSTRING
$PREFIX
$SUFFIX

                            ")
-- Used to initialise fnotenumber and fnotestring bindings. --

DEFINE(INITIALISEFNOTE,   "
                                    REQ #binding-name{""fnotenumber""},
                                    REQ #binding-value{>=0}
                            ")


-- Used to increment fnotenumber binding. --

DEFINE(INCFNOTENUMBER,   "
                                    REQ #binding-name{""fnotenumber""},
                                    REQ #binding-value{<num-expr>::=INC(B_REF(PREC
                                        (CURR-OBJ))(""fnotenumber""));}
                            ")


-- Used to create a fnotestring from a fnotenumber binding. --

DEFINE(FNOTENUMBERSTRING,   "
  REQ #binding-name{""fnotestring""},
  REQ #binding-value{<str-expr> ::=
                                    MK-STR(B_REF(CURR-OBJ)(""fnotenumber""))
                                    |U-ALPHA(B_REF(CURR-OBJ)(""fnotenumber""))
                                    |L-ALPHA(B_REF(CURR-OBJ)(""fnotenumber""))
                                    |U-ROM(B_REF(CURR-OBJ)(""fnotenumber""))
                                    |L-ROM(B_REF(CURR-OBJ)(""fnotenumber""));}
                            ")


-- Used to reset the footnote number string to a string literal. This provides a mechanism for setting
--  the footnote number string to something other than a numeric value. --

DEFINE(FNOTESTRINGLITERAL,  "
                                    REQ #binding-name{""fnotestring""},
                                    REQ #binding-value{ANY_STRING}
                            ")


-- This string expression is allowed in a content generator for FootnoteNumber and FootnoteReference
--  to automatically generate text for a footnote number. --

DEFINE(FNOTENUMBER,   "
<string-expr>               ::=     [ANY_STRING]<num-str>[ANY_STRING];
<num-str>                   ::=     B_REF(SUP(CURR-OBJ))(""fnotestring"");
                            ")

DEFINE(PGNUMBER        ,   "
<string-expr>               ::=     [ANY_STRING]{<num-str>}[ANY_STRING];

<num-str>                   ::=     MK-STR(<numeric-expr>)
                                    |U-ALPHA(<numeric-expr>)
                                    |L-ALPHA(<numeric-expr>)
                                    |U-ROM(<numeric-expr>)
                                    |L-ROM(<numeric-expr>);

<numeric-expr>              ::=     B_REF(SUP(CURR-INST(<class-or-type-1>,
                                      CURR-OBJ)))(""PGnum"")
```

```
                                |B_REF(CURR-INST(<class-or-type-2>,
                                  CURR-OBJ))(""PGnum"");

<class-or-type-1>          ::=  'frame';

<class-or-type-2>               ::=        'page'
                                           |OBJECT_CLASS_ID_OF(Page)
                                           |OBJECT_CLASS_ID_OF(RectoPage)
                                           |OBJECT_CLASS_ID_OF(VersoPage);
                           ")
```

7.3.2 Factor constraints

7.3.2.1 FACTOR ANY-LOGICAL {

```
GENERIC:
  REQ      Object-type                    {VIRTUAL},
  REQ      Object-class-identifier        {ANY_VALUE}

SPECIFIC:

  PERM     Object-type                    {VIRTUAL},
  REQ      Object-identifier              {ANY_VALUE},
  REQ      Object-class                   {VIRTUAL}

SPECIFIC_AND_GENERIC:
  PERM     User-readable-comments         {ANY_STRING},
  PERM     User-visible-name              {ANY_STRING}}
```

7.3.3 Constituent constraints

7.3.3.1 DocumentLogicalRoot: ANY-LOGICAL {

```
GENERIC:
  REQ      Object-type                    {'document-logical-root'},
  REQ      Generator-for-subordinates     {$DocLogRootGFS},
  REQ      Application-comments           {REQ #constraint-name {"0"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
  PERM     Object-type                    {'document-logical-root'},
  REQ      Object-class                   {OBJECT_CLASS_ID_OF (DocumentLogicalRoot)},
  REQ      Subordinates                   {SUB_ID_OF(Passage)+},
  PERM     Application-comments           {REQ #constraint-name {"0"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
  PERM     Bindings                       {PMUL{$INITIALISEANY},
                                           PERM{$INITIALISEFNOTE}}}
```

7.3.3.2 Passage: ANY-LOGICAL {

```
GENERIC:
  REQ      Object-type                    {'composite-logical-object'},
                                            REQ      Generator-for-subordinates{$PassageGFS},
  REQ      Application-comments           {REQ #constraint-name {"1"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
  PERM     Object-type                    {'composite-logical-object'},
  REQ      Object-class                   {OBJECT_CLASS_ID_OF(Passage)},
  REQ      Subordinates                   {SUB_ID_OF(NumberedSegment)+,
                                           SUB_ID_OF(BodyText)+,
                                           SUB_ID_OF(BodyRaster)+,
                                           SUB_ID_OF(BodyGeometric)+,
                                           SUB_ID_OF(Paragraph)+},
  PERM     Application-comments           {REQ #constraint-name {"1"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
  PERM     Layout-style                   {STYLE_ID_OF(L-Style1)},
  PERM     Bindings                       {PMUL{$INITIALISEANY},
                                           PERM{$INITIALISEFNOTE}}}
```

7.3.3.3 NumberedSegment: ANY-LOGICAL {

GENERIC:

REQ	**Object-type**	**{'composite-logical-object'},**
REQ	**Generator-for-subordinates**	**{$NumberedSegmentGFS},**
REQ	**Application-comments**	**{REQ #constraint-name {"2"},** **PERM #external-data {ANY_VALUE}},**
REQ	**Bindings**	**{PMUL{$INITIALISEANY},** **PERM{$USENUMBERS},** **REQ{$USENUMBERSTRINGS}}**

-- The binding USE NUMBERS shall also be present if USENUMBERSTRINGS does not use the ORD option. --

SPECIFIC:

PERM	**Object-type**	**{'composite-logical-object'},**
REQ	**Object-class**	**{OBJECT_CLASS_ID_OF(NumberedSegment)},**
REQ	**Subordinates**	**{SUB_ID_OF(Number),** **SUB_ID_OF(NumberedSegment)+,** **SUB_ID_OF(BodyText)+,** **SUB_ID_OF(BodyRaster)+,** **SUB_ID_OF(BodyGeometric)+,** **SUB_ID_OF(Paragraph)+},**
PERM	**Bindings**	**{PMUL{$INITIALISEANY}},**
PERM	**Application-comments**	**{REQ #constraint-name {"2"},** **PERM #external-data {ANY_VALUE}}**

SPECIFIC_AND_GENERIC:

PERM	**Layout-style**	**{STYLE_ID_OF(L-Style4)}}**

7.3.3.4 Number: ANY-LOGICAL {

GENERIC:

REQ	**Object-type**	**{'basic-logical-object'},**
REQ	**Content-generator**	**{$SEGMENTNUMBER},**
REQ	**Application-comments**	**{REQ #constraint-name {"3"},** **PERM #external-data {ANY_VALUE}}**

SPECIFIC:

PERM	**Object-type**	**{'basic-logical-object'},**
REQ	**Object-class**	**{OBJECT_CLASS_ID_OF(Number)},**
PERM	**Application-comments**	**{REQ #constraint-name {"3"},** **PERM #external-data {ANY_VALUE}}**

SPECIFIC_AND_GENERIC:

PERM	**Layout-style**	**{STYLE_ID_OF(L-Style2)},**
PERM	**Presentation-style**	**{STYLE_ID_OF(P-Style1)},**
PERM	**Content-architecture-class**	**{$FC\|$PC|$FPC}}**

7.3.3.5 Paragraph: ANY-LOGICAL {

GENERIC:

REQ	**Object-type**	**{'composite-logical-object'},**
REQ	**Generator-for-subordinates**	**{$ParagraphGFS},**
REQ	**Application-comments**	**{REQ #constraint-name {"6"},** **PERM #external-data {ANY_VALUE}}**

SPECIFIC:

PERM	**Object-type**	**{'composite-logical-object'},**
REQ	**Object-class**	**{OBJECT_CLASS_ID_OF (Paragraph)},**
REQ	**Subordinates**	**{SUB_ID_OF(BodyText)+,** **SUB_ID_OF(Footnote)+,** **SUB_ID_OF(BodyRaster)+,** **SUB_ID_OF(BodyGeometric)+},**
PERM	**Application-comments**	**{REQ #constraint-name {"6"},** **PERM #external-data {ANY_VALUE}}**

SPECIFIC_AND_GENERIC:

PERM	**Layout-style**	**{STYLE_ID_OF(L-Style4)}}**

7.3.3.6 BodyText: ANY-LOGICAL {

GENERIC:

REQ	**Object-type**	**{'basic-logical-object'},**

PERM Resource {ANY_VALUE},
REQ Application-comments {REQ #constraint-name {"14"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:
PERM Object-type {'basic-logical-object'},
REQ Object-class {OBJECT_CLASS_ID_OF(BodyText)},
PERM Application-comments {REQ #constraint-name {"14"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
PERM Layout-style {STYLE_ID_OF(L-Style2)},
PERM Presentation-style {STYLE_ID_OF(P-Style1)},
PERM Content-architecture-class {$FC|$PC|$FPC},
PERM Content-portions {CONTENT_ID_OF(Character-content-portion)+}}
-- if the attribute "content portions" is specified neither in the specific nor in the generic part then
-- the attribute "resource" shall be specified --

7.3.3.7 BodyGeometric: ANY-LOGICAL {

GENERIC:
REQ Object-type {'basic-logical-object'},
REQ Content-architecture-class {$FPG},
PERM Resource {ANY_VALUE},
REQ Application-comments {REQ #constraint-name {"18"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:
PERM Object-type {'basic-logical-object'},
REQ Object-class {OBJECT_CLASS_ID_OF(BodyGeometric)},
PERM Content-architecture-class {$FPG},
PERM Application-comments {REQ #constraint-name {"18"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
PERM Layout-style {STYLE_ID_OF(L-Style5)},
PERM Presentation-style {STYLE_ID_OF(P-Style4)},
PERM Content-portions {CONTENT_ID_OF(Geometric-graphics-content-portion)}}
-- if the attribute "content portions" is specified neither in the specific nor in the generic part then
-- the attribute "resource" shall be specified --

7.3.3.8 BodyRaster: ANY-LOGICAL {

GENERIC:
REQ Object-type {'basic-logical-object'},
REQ Content-architecture-class {$FPR},
PERM Resource {ANY_VALUE},
REQ Application-comments {REQ #constraint-name {"17"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:
PERM Object-type {'basic-logical-object'},
REQ Object-class {OBJECT_CLASS_ID_OF(BodyRaster)},
PERM Content-architecture-class {$FPR},
PERM Application-comments {REQ #constraint-name {"17"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
PERM Layout-style {STYLE_ID_OF(L-Style5)},
PERM Presentation-style {STYLE_ID_OF(P-Style3)},
PERM Content-portions {CONTENT_ID_OF(Raster-graphics-content-portion)}}
-- if the attribute "content portions" is specified neither in the specific nor in the generic part then
-- the attribute "resource" shall be specified --

7.3.3.9 Footnote: ANY-LOGICAL {

GENERIC:
REQ Object-type {'composite-logical-object'},
REQ Generator-for-subordinates {$FootnoteGFS},
PERM Bindings {PERM{REQ{$INCFNOTENUMBER,
$FNOTENUMBERSTRING}
|$FNOTESTRINGLITERAL}},

REQ Application-comments {REQ #constraint-name {"8"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:
PERM Object-type {'composite-logical-object'},
REQ Object-class {OBJECT_CLASS_ID_OF(Footnote)},
REQ Subordinates {SUB_ID_OF(FootnoteReference),
SUB_ID_OF(FootnoteBody)},
PERM Bindings {$FNOTESTRINGLITERAL},
PERM Bindings {$FNOTESTRINGLITERAL},
-- if bindings are not specified on the GENERIC then they shall be specified on the SPECIFIC --

PERM Application-comments {REQ #constraint-name {"8"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
PERM Layout-style {STYLE_ID_OF(L-Style7)}}

7.3.3.10 FootnoteReference: ANY-LOGICAL {

GENERIC:
REQ Object-type {'basic-logical-object'},
REQ Content-generator {$FNOTENUMBER},
REQ Application-comments {REQ #constraint-name {"10"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:
PERM Object-type {'basic-logical-object'},
REQ Object-class {OBJECT_CLASS_ID_OF (FootnoteReference)},
PERM Application-comments {REQ #constraint-name {"10"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
PERM Layout-style {STYLE_ID_OF(L-Style10)},
PERM Presentation-style {STYLE_ID_OF(P-Style1)},
PERM Content-architecture-class {$FC|$PC|$FPC}}

7.3.3.11 FootnoteBody: ANY-LOGICAL {

GENERIC:
REQ Object-type {'composite-logical-object'},
REQ Generator-for-subordinates {$FootnoteBodyGFS},
REQ Application-comments {REQ #constraint-name {"11"},
PERM #external-data {ANY_VALUE}},
PERM Layout-style {STYLE_ID_OF (L-Style11)}

SPECIFIC:
PERM Object-type {'composite-logical-object'},
REQ Object-class {OBJECT_CLASS_ID_OF(FootnoteBody)},
REQ Subordinates {SUB_ID_OF(FootnoteNumber),
SUB_ID_OF(FootnoteText)+},
PERM Application-comments {REQ #constraint-name {"11"},
PERM #external-data {ANY_VALUE}},
PERM Layout-style {STYLE_ID_OF (L-Style11)}
}

7.3.3.12 FootnoteNumber: ANY-LOGICAL {

GENERIC:
REQ Object-type {'basic-logical-object'},
REQ Content-generator {$FNOTENUMBER},
REQ Application-comments {REQ #constraint-name {"9"},
PERM #external-data {ANY_VALUE}},
REQ Layout-style {STYLE_ID_OF(L-Style9)}

SPECIFIC:
PERM Object-type {'basic-logical-object'},
REQ Object-class {OBJECT_CLASS_ID_OF(FootnoteNumber)},
PERM Application-comments {REQ #constraint-name {"9"},
PERM #external-data {ANY_VALUE}},
PERM Layout-style {STYLE_ID_OF(L-Style9)}

SPECIFIC_AND_GENERIC:

```
  PERM      Presentation-style              {STYLE_ID_OF(P-Style1)},
  PERM      Content-architecture-class      {$FC|$PC|$FPC}}
```

7.3.3.13 FootnoteText: ANY-LOGICAL {

```
GENERIC:
  REQ       Object-type                     {'basic-logical-object'},
  REQ       Application-comments            {REQ #constraint-name {"12"},
                                             PERM #external-data {ANY_VALUE}},
  REQ       Layout-style                    {STYLE_ID_OF(L-Style6)}

SPECIFIC:
  PERM      Object-type                     {'basic-logical-object'},
  REQ       Object-class                    {OBJECT_CLASS_ID_OF(FootnoteText)},
  REQ       Content-portions                {CONTENT_ID_OF(
                                             Character-content-portion)+},
  PERM      Application-comments            {REQ #constraint-name {"12"},
                                             PERM #external-data {ANY_VALUE}},
  PERM      Layout-style                    {STYLE_ID_OF(L-Style6)}

SPECIFIC_AND_GENERIC:
  PERM      Presentation-style              {STYLE_ID_OF(P-Style1)},
  PERM      Content-architecture-class      {$FC|$PC|$FPC}}
```

7.3.3.14 CommonContent {

```
GENERIC:
  REQ       Object-type                     {'composite-logical-object'},
  REQ       Object-class-identifier         {ANY_VALUE},
  REQ       Generator-for-subordinates      {$CommonContentGFS},
  REQ       Application-comments            {REQ #constraint-name {"19"},
                                             PERM #external-data {ANY_VALUE}},
  PERM      User-readable-comments          {ANY_STRING},
  PERM      User-visible-name               {ANY_STRING}}
```

7.3.3.15 CommonText {

```
GENERIC:
  REQ       Object-type                     {'basic-logical-object'},
  REQ       Object-class-identifier         {ANY_VALUE},
  PERM      Content-portions                {CONTENT_ID_OF(Character-content-portion)+},
  PERM      Resource                        {ANY_VALUE},
  PERM      Layout-style                    {STYLE_ID_OF(L-Style3)},
  PERM      Presentation-style              {STYLE_ID_OF(P-Style2)},
  PERM      Content-architecture-class      {$FC|$PC|$FPC},
  REQ       Application-comments            {REQ #constraint-name {"20"},
                                             PERM #external-data {ANY_VALUE}},
  PERM      User-readable-comments          {ANY_STRING},
  PERM      User-visible-name               {ANY_STRING}}
```

-- either the attribute "content portions" or "resource" shall be specified in the above constituent constraint --

7.3.3.16 PageNumber {

```
GENERIC:
  REQ       Object-type                     {'basic-logical-object'},
  REQ       Object-class-identifier         {ANY_VALUE},
  REQ       Content-generator               {$PGNUMBER},
  PERM      Layout-style                    {STYLE_ID_OF(L-Style3)},
  PERM      Presentation-style              {STYLE_ID_OF(P-Style2)},
  PERM      Content-architecture-class      {$FC|$PC|$FPC},
  REQ       Application-comments            {REQ #constraint-name {"40"},
                                             PERM #external-data {ANY_VALUE}},
  PERM      User-readable-comments          {ANY_STRING},
  PERM      User-visible-name               {ANY_STRING}}
```

7.3.3.17 CommonGeometric {

```
GENERIC:
  REQ       Object-type                     {'basic-logical-object'},
  REQ       Object-class-identifier         {ANY_VALUE},
  PERM      Content-portions                {CONTENT_ID_OF(Geometric-graphics-content-portion)},
```

```
  PERM      Resource                      {ANY_VALUE},
  PERM      Layout-style                  {STYLE_ID_OF(L-Style8)},
  PERM      Presentation-style            {STYLE_ID_OF(P-Style4)},
  REQ       Content-architecture-class    {$FPG},
  REQ       Application-comments          {REQ #constraint-name {"22"},
                                           PERM #external-data {ANY_VALUE}},
  PERM      User-readable-comments        {ANY_STRING},
  PERM      User-visible-name             {ANY_STRING}}
 -- either the attribute "content portions" or "resource" shall be specified in the above constituent constraint --
```

7.3.3.18 CommonRaster {

```
GENERIC:
  REQ       Object-type                   {'basic-logical-object'},
  REQ       Object-class-identifier       {ANY_VALUE},
  PERM      Content-portions              {CONTENT_ID_OF(Raster-graphics-content-portion)},
  PERM      Resource                      {ANY_VALUE},
  PERM      Layout-style                  {STYLE_ID_OF(L-Style8)},
  PERM      Presentation-style            {STYLE_ID_OF(P-Style3)},
  REQ       Content-architecture-class    {$FPR},
  REQ       Application-comments          {REQ #constraint-name {"21"},
                                           PERM #external-data {ANY_VALUE}},
  PERM      User-readable-comments        {ANY_STRING},
  PERM      User-visible-name             {ANY_STRING}}
 -- either the attribute "content portions" or "resource" shall be specified in the above constituent constraint --
```

7.4 Layout constituent constraints

7.4.1 Macro definitions

```
DEFINE(DocLayRootGFS,  "
<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>}...)
                                    |CHO({<construction-term>}...);

<construction-factor>        ::=    OBJECT_CLASS_ID_OF(PageSet)
                                    |<construction-type>;
                        ")

DEFINE(PageSetGFS,  "
<construction-expr>          ::=    <pageset-1>
                                    |<pageset-2>
                                    |<pageset-3>
                                    |SEQ(<pageset-1><pageset-2>)
                                    |SEQ(<pageset-1><pageset-3>);

<pageset-1>                  ::=    OBJECT_CLASS_ID_OF(Page)
                                    |OPT OBJECT_CLASS_ID_OF(Page);

<pageset-2>                  ::=    REP OBJECT_CLASS_ID_OF(Page)
                                    |OPT REP OBJECT_CLASS_ID_OF(Page);

<pageset-3>                  ::=    OPT REP SEQ(OBJECT_CLASS_ID_OF(RectoPage)
                                    OPT OBJECT_CLASS_ID_OF(VersoPage))
                                    |OPT REP SEQ(OBJECT_CLASS_ID_OF(VersoPage)
                                    OPT OBJECT_CLASS_ID_OF(RectoPage))
                                    |REP SEQ(OBJECT_CLASS_ID_OF(RectoPage)
                                    OPT OBJECT_CLASS_ID_OF(VersoPage))
                                    |REP SEQ(OBJECT_CLASS_ID_OF(VersoPage)
                                    OPT OBJECT_CLASS_ID_OF(RectoPage));
                        ")

DEFINE(PageGFS,  "
<construction-expr>          ::=    SEQ([<headerarea>]<bodyarea>[<footerarea>])
                                    |<bodyarea>;
```

```
<headerarea>                    ::=     OBJECT_CLASS_ID_OF(BasicHeader)
                                        |OBJECT_CLASS_ID_OF(CompositeHeader);

<bodyarea>                      ::=     OBJECT_CLASS_ID_OF(BasicBody)
                                        |OBJECT_CLASS_ID_OF(VariableCompositeBody);

<footerarea>                    ::=     OBJECT_CLASS_ID_OF(BasicFooter)
                                        |OBJECT_CLASS_ID_OF(CompositeFooter);
                        ")

DEFINE(VariableCompositeBodyGFS,   "
<construction-expr>             ::=     <construction-term>
                                        |<construction-type>
                                        |SEQ(<construction-term>, <construction-footnote>)
                                        |SEQ(<construction-type>, <construction-footnote>);

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

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

<construction-factor>           ::=     OBJECT_CLASS_ID_OF(BasicFloat)
                                        |OBJECT_CLASS_ID_OF(SnakingColumns)
                                        |OBJECT_CLASS_ID_OF(SynchronizedColumns)
                                        |<construction-type>;

<construction-footnote>         ::=     OBJECT_CLASS_ID_OF(FootnoteArea)
                                        |OPT OBJECT_CLASS_ID_OF(FootnoteArea);
                        ")

DEFINE(SnakingColumnsGFS,   "
<construction-expr>             ::=     SEQ({OBJECT_CLASS_ID_OF(ColumnVariable)}...)
                                        |REP OBJECT_CLASS_ID_OF(ColumnVariable);
                        ")

DEFINE(SynchronizedColumnsGFS,   "
<construction-expr>             ::=     SEQ({OBJECT_CLASS_ID_OF(ColumnFixed)}...);
                        ")

DEFINE(HeaderFooterGFS,   "
<construction-expr>             ::=     <fixed-common-content-frames>
                                        |<variable-common-content-frames>;

<fixed-common-content-frames>
                                ::=     SEQ({OBJECT_CLASS_ID_OF(SourcedContentFixed)
                                        |OBJECT_CLASS_ID_OF(ArrangedContentFixed)}...);

<variable-common-content-frames>
                                ::=     SEQ({OBJECT_CLASS_ID_OF(SourcedContentVariable)
                                        |OBJECT_CLASS_ID_OF(ArrangedContentVariable)}...);
                        ")

DEFINE(INITIALISEPGNUM,  "
                                        REQ #binding-name{""PGnum""},
                                        REQ #binding-value{>=–1}
                        ")

DEFINE(PAGENUMBER,  "
                                        REQ #binding-name{""PGnum""},
                                        REQ #binding-value{<num-expr>::=INC(B_REF(PREC(
                                        CURR-OBJ))(""PGnum""));}
                        ")

DEFINE(PDA-FPDA, "{'processable'|'formatted-processable'}")
```

7.4.2 Factor constraints

7.4.2.1 FACTOR ANY-LAYOUT {

```
GENERIC:
  REQ       Object-type                 {VIRTUAL},
  REQ       Object-class-identifier     {ANY_VALUE},
```

```
   REQ      Application-comments                {VIRTUAL}

SPECIFIC:
   PERM     Object-type                         {VIRTUAL},
   REQ      Object-identifier                   {ANY_VALUE},
   CASE     $DAC OF {
   $FDA:    PERM        Object-class            {VIRTUAL}
   $FPDA:   REQ         Object-class            {VIRTUAL}
            },
   REQ      Subordinates                        {VIRTUAL},
   PERM     Application-comments                {VIRTUAL}

SPECIFIC_AND_GENERIC:
   PERM     User-readable-comments              {ANY_STRING},
   PERM     User-visible-name                   {ANY_STRING}}
```

7.4.2.2 FACTOR ANY-PAGE: ANY-LAYOUT {

```
GENERIC:
   REQ      Object-type                         {'page'},
   CASE     $DAC OF {
            $PDA-FPDA:
   PERM     Bindings                            {$PAGENUMBER},
   REQ      Generator-for-subordinates          {$PageGFS}
            }

SPECIFIC:
   PERM     Object-type                         {'page'},
   REQ      Subordinates                        {SUB_ID_OF(BasicHeader),
                                                 SUB_ID_OF(CompositeHeader),
                                                 SUB_ID_OF(BasicBody),
                                                 SUB_ID_OF(VariableCompositeBody),
                                                 SUB_ID_OF(BasicFooter),
                                                 SUB_ID_OF(CompositeFooter)}

SPECIFIC_AND_GENERIC:
   PERM     Dimensions                          {REQ #horizontal-dimension
                                                      {REQ #fixed-dimension {<=14030}},
                                                   REQ #vertical-dimension
                                                      {REQ #fixed-dimension {<=19840}}
                                                      -- up to ISO A3 portrait --
                                                |REQ #horizontal-dimension
                                                      {REQ #fixed-dimension {<=19840}},
                                                   REQ #vertical-dimension
                                                      {REQ #fixed-dimension {<=14030}}
                                                      -- up to ISO A3 landscape --
                                                |REQ #horizontal-dimension
                                                      {REQ #fixed-dimension {<=13200}},
                                                   REQ #vertical-dimension
                                                      {REQ #fixed-dimension {<=20400}}
                                                      -- up to ANSI-B portrait --
                                                |REQ #horizontal-dimension
                                                      {REQ #fixed-dimension {<=20400}},
                                                   REQ #vertical-dimension
                                                      REQ #fixed-dimension {<=13200}}
                                                      -- up to ANSI-B landscape --},
   PERM  Page-position                          {ANY_VALUE}}
```

7.4.2.3 FACTOR ANY-FRAME-FIXED: ANY-LAYOUT {

```
GENERIC:
   REQ      Object-type                         {'frame'}

SPECIFIC:
   PERM     Object-type                         {'frame'}

SPECIFIC_AND_GENERIC:
   PERM     Position                            {REQ #fixed-position
                                                      {REQ #horizontal-position {ANY_VALUE},
                                                       REQ #vertical-position {ANY_VALUE}}},
   PERM     Dimensions                          {REQ #horizontal-dimension
```

```
                                                    {REQ #fixed-dimension {ANY_VALUE}},
                                                 REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}}},
  PERM      Border                                {ANY_VALUE}}
```

7.4.2.4 FACTOR ANY-FRAME-VARIABLE: ANY-LAYOUT {

```
GENERIC:
  REQ       Object-type                           {'frame'}

SPECIFIC:
  PERM      Object-type                           {'frame'},
  CASE      $DAC OF {
            $FPDA:      REQ   Position            {REQ #fixed-position
                                                    {REQ #horizontal-position {ANY_VALUE},
                                                     REQ #vertical-position {ANY_VALUE}}},
                        REQ   Dimensions          {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}},
                                                 REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}}}
            }

SPECIFIC_AND_GENERIC:
  CASE      $DAC OF {
            $FDA:       PERM  Position            {REQ #fixed-position
                                                    {REQ #horizontal-position {ANY_VALUE},
                                                     REQ #vertical-position {ANY_VALUE}}},
                        PERM  Dimensions          {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}},
                                                 REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}}}
            },
  PERM      Border                                {ANY_VALUE}
            }
```

7.4.2.5 FACTOR BLOCK {

```
SPECIFIC:
   REQ      Object-type                           {'block'},
   REQ      Object-identifier                     {ANY_VALUE},
   PERM  Content-architecture-class               {$FC|$FPC|$FPR|$FPG},
   PERM  Presentation-attributes {
      PERM #character-attributes {
            PERM #alignment                       {ANY_VALUE},
            PERM #character-fonts                 {ANY_VALUE},
            PERM #character-spacing               {ANY_VALUE},
            PERM #character-orientation           {'0-degrees'
                                                  |'90-degrees'},
      PERM #character-path                        {'0-degrees'
                                                  |'90-degrees'
                                                  |'180-degrees'
                                                  |'270-degrees'},
      PERM #code-extension-announcers             {$CDEXTEN},
      PERM #first-line-offset                     {ANY_VALUE},
      PERM #graphic-character-sets                {$PERMIT-GRCHAR},
      PERM #graphic-character-sub-repertoire      {ANY_VALUE},
      PERM #graphic-rendition                     {$GRAPHICRENDITIONS},
      PERM #itemization                           {ANY_VALUE},
      PERM #kerning-offset                        {ANY_VALUE},
      PERM #line-layout-table                     {ANY_VALUE},
      PERM #line-progression                      {'90-degrees'|'270-degrees},
      PERM #line-spacing                          {ANY_VALUE},
      PERM #initial-offset                        {ANY_VALUE}}},
   PERM  User-readable-comments                   {ANY_STRING},
   PERM  User-visible-name                        {ANY_STRING},
   PERM  Position                                 {REQ #fixed-position
                                                    {REQ #horizontal-position {ANY_VALUE},
                                                     REQ #vertical-position {ANY_VALUE}}},
   PERM  Dimensions                               {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}},
```

```
                                        REQ #vertical-dimension
                                            {REQ #fixed-dimension {ANY_VALUE}}}}
```

7.4.3 Constituent constraints

7.4.3.1 DocumentLayoutRoot: ANY-LAYOUT {

```
GENERIC:
  REQ     Object-type                   {'document-layout-root'},
  CASE    $DAC OF {
          $PDA-FPDA:
  PERM    Bindings                      {$INITIALISEPGNUM},
  REQ     Generator-for-subordinates    {$DocLayRootGFS}
          },
  REQ     Application-comments          {REQ #constraint-name {"0"},
                                         PERM #external-data {ANY_VALUE}}

SPECIFIC:
  PERM    Object-type                   {'document-layout-root'},
  CASE    $DAC OF {
          $FDA: PERM Object-class       {OBJECT_CLASS_ID_OF(DocumentLayoutRoot)}
          $FPDA: REQ Object-class       {OBJECT_CLASS_ID_OF(DocumentLayoutRoot)}
          },
  REQ     Subordinates                  {SUB_ID_OF(PageSet)+},
  PERM    Application-comments          {REQ #constraint-name {"0"},
                                         PERM #external-data {ANY_VALUE}}}
```

7.4.3.2 PageSet: ANY-LAYOUT {

```
GENERIC:
  REQ     Object-type                   {'page-set'},
  CASE    $DAC OF {
          $PDA-FPDA:
  PERM    Bindings                      {$INITIALISEPGNUM},
  REQ     Generator-for-subordinates    {$PageSetGFS}
          },
  REQ     Application-comments          {REQ #constraint-name {"1"},
                                         PERM #external-data {ANY_VALUE}}

SPECIFIC:
  PERM    Object-type                   {'page-set'},
  CASE    $DAC OF {
          $FDA: PERM Object-class       {OBJECT_CLASS_ID_OF(PageSet)}
          $FPDA: REQ Object-class       {OBJECT_CLASS_ID_OF(PageSet)}
          },
  REQ     Subordinates                  {SUB_ID_OF(Page)+,
                                         SUB_ID_OF(RectoPage)+,
                                         SUB_ID_OF(VersoPage)+},
  PERM    Application-comments          {REQ #constraint-name {"1"},
                                         PERM #external-data {ANY_VALUE}}}
```

7.4.3.3 Page: ANY-PAGE {

```
GENERIC:
  REQ     Application-comments          {REQ #constraint-name {"2"},
                                         PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE    $DAC OF {
          $FDA: PERM Object-class       {OBJECT_CLASS_ID_OF(Page)}
          $FPDA: REQ Object-class       {OBJECT_CLASS_ID_OF(Page)}
          },
  PERM    Application-comments          {REQ #constraint-name {"2"},
                                         PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
  PERM    Medium-type                   {PERM #nominal-page-size {$NominalPageSizes},
                                         PERM #side-of-sheet {ANY_VALUE}}}
```

7.4.3.4 RectoPage: ANY-PAGE {

```
GENERIC:
```

```
  REQ       Application-comments          {REQ #constraint-name {"3"},
                                           PERM #external-data {ANY_VALUE}},
  REQ       Medium-type                   {PERM #nominal-page-size {$NominalPageSizes},
                                           REQ  #side-of-sheet {'recto'|'unspecified'}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM  Object-class      {OBJECT_CLASS_ID_OF(RectoPage)}
            $FPDA: REQ  Object-class      {OBJECT_CLASS_ID_OF(RectoPage)}
            },
  PERM      Application-comments          {REQ #constraint-name {"3"},
                                           PERM #external-data {ANY_VALUE}},
  PERM      Medium-type                   {PERM #nominal-page-size {$NominalPageSizes},
                                           PERM #side-of-sheet {'recto'|'unspecified'}}}
```

7.4.3.5 VersoPage: ANY-PAGE {

```
GENERIC:
  REQ       Application-comments          {REQ #constraint-name {"4"},
                                           PERM #external-data {ANY_VALUE}},
  REQ       Medium-type                   {PERM  #nominal-page-size {$NominalPageSizes},
                                           REQ  #side-of-sheet {'verso'|'unspecified'}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM  Object-class      {OBJECT_CLASS_ID_OF(VersoPage)}
            $FPDA: REQ  Object-class      {OBJECT_CLASS_ID_OF(VersoPage)}
            },
  PERM      Application-comments          {REQ #constraint-name {"4"},
                                           PERM #external-data {ANY_VALUE}},
  PERM      Medium-type                   {PERM #nominal-page-size {$NominalPageSizes},
                                           PERM #side-of-sheet {'verso'|'unspecified'}}}
```

7.4.3.6 BasicBody: ANY-FRAME-FIXED {

```
GENERIC:
  PERM      Layout-path                   {'270-degrees' -- page layout A --
                                          |'0-degrees'  -- page layout B --
                                          |'180-degrees' -- page layouts C and D --},
  REQ       Application-comments          {REQ #constraint-name {"28"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM  Object-class      {OBJECT_CLASS_ID_OF(BasicBody)}
            $FPDA: REQ  Object-class      {OBJECT_CLASS_ID_OF(BasicBody)}
            },
  REQ       Subordinates                  {SUB_ID_OF(SpecificBlock)+},
  PERM      Application-comments          {REQ #constraint-name {"28"},
                                           PERM #external-data {ANY_VALUE}}}
```

7.4.3.7 VariableCompositeBody: ANY-FRAME-FIXED {

```
GENERIC:
  CASE      $DAC OF {
            $PDA-FPDA:
                  REQ   Generator-for-subordinates
                                          {$VariableCompositeBodyGFS},
                  PERM  Layout-path       {'270-degrees' -- page layout A --
                                          |'0-degrees'  -- page layout B --
                                          |'180-degrees' -- page layouts C and D --}
            },
  REQ       Application-comments          {REQ #constraint-name {"7"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM  Object-class      {OBJECT_CLASS_ID_OF(VariableCompositeBody)}
            $FPDA: REQ  Object-class      {OBJECT_CLASS_ID_OF(VariableCompositeBody)}
            },
  REQ       Subordinates                  {SUB_ID_OF(BasicFloat)+,
```

```
                                        SUB_ID_OF(SnakingColumns)+,
                                        SUB_ID_OF(SynchronizedColumns)+,
                                        SUB_ID_OF(FootnoteArea)},
  PERM      Application-comments        {REQ #constraint-name {"7"},
                                         PERM #external-data {ANY_VALUE}}}
```

7.4.3.8 BasicFloat: ANY-FRAME-VARIABLE {

```
GENERIC:
  CASE      $DAC OF {
            $PDA-FPDA:
                  REQ   Position              {REQ #variable-position {
                                                    PERM #offset {ANY_VALUE},
                                                    PERM #separation {ANY_VALUE},
                                                    PERM #alignment {ANY_VALUE},
                                                    PERM #fill-order {'normal-order'}}},
                  PERM  Permitted-categories  {ANY_STRING},
                  CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {
                  {'270-degrees'}: -- page layout A --
                        REQ  Dimensions       {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}
                                                    |REQ #maximum-size {'applies'}},
                                               REQ #vertical-dimension
                                                    {REQ #rule-b {ANY_VALUE}}},
                        PERM Layout-path      {'270-degrees'}
                  {'0-degrees'}: -- page layout B --
                        REQ  Dimensions       {REQ #horizontal-dimension
                                                    {REQ #rule-b {ANY_VALUE}},
                                               REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}
                                                    |REQ #maximum-size {'applies'}}},
                        REQ  Layout-path      {'0-degrees'}
                  {'180-degrees'}: -- page layouts C and D --
                        REQ  Dimensions       {REQ #horizontal-dimension
                                                    {REQ #rule-b {ANY_VALUE}},
                                               REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}
                                                    |REQ #maximum-size {'applies'}}},
                        REQ  Layout-path    {'180-degrees'}
                        }
                        },
  REQ       Application-comments        {REQ #constraint-name {"12"},
                                         PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
              $FDA:  PERM  Object-class     {OBJECT_CLASS_ID_OF(BasicFloat)}
              $FPDA: REQ   Object-class     {OBJECT_CLASS_ID_OF(BasicFloat)}
            },
  REQ       Subordinates                  {SUB_ID_OF(SpecificBlock)+},
  PERM      Application-comments          {REQ #constraint-name {"12"},
                                           PERM #external-data {ANY_VALUE}}}
```

7.4.3.9 SynchronizedColumns: ANY-FRAME-VARIABLE {

```
GENERIC:
  CASE      $DAC OF {
              $PDA-FPDA:
              REQ   Generator-for-subordinates
                                          {$SynchronizedColumnsGFS},
              REQ   Position              {REQ #variable-position {
                                                PERM #offset {ANY_VALUE},
                                                PERM #separation {ANY_VALUE},
                                                PERM #alignment {ANY_VALUE},
                                                PERM #fill-order {'normal-order'}}},
              CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {
              {'270-degrees'}: -- page layout A --
                    REQ  Dimensions       {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #maximum-size {'applies'}},
```

```
                                        REQ #vertical-dimension
                                              {REQ #rule-b {ANY_VALUE}}},
                    PERM Layout-path  {'270-degrees'}
                  {'0-degrees'}: -- page layout B --
                    REQ  Dimensions           {REQ #horizontal-dimension
                                              {REQ #rule-b {ANY_VALUE}},
                                               REQ #vertical-dimension
                                              {REQ #fixed-dimension {ANY_VALUE}
                                              |REQ #maximum-size {'applies'}}},
                    REQ  Layout-path  {'0-degrees'}
                  {'180-degrees'}: -- page layouts C and D --
                    REQ  Dimensions           {REQ #horizontal-dimension
                                              {REQ #rule-b {ANY_VALUE}},
                                               REQ #vertical-dimension
                                              {REQ #fixed-dimension {ANY_VALUE}
                                              |REQ #maximum-size {'applies'}}},
                    REQ  Layout-path  {'180-degrees'}
                }
                },
  REQ       Application-comments          {REQ #constraint-name {"11"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
              $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF(SynchronizedColumns)}
              $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF(SynchronizedColumns)}
                   },
  REQ       Subordinates                  {SUB_ID_OF(ColumnFixed)+},
  PERM      Application-comments          {REQ #constraint-name {"11"},
                                           PERM #external-data {ANY_VALUE}}}
```

7.4.3.10 SnakingColumns: ANY-FRAME-VARIABLE {

```
GENERIC:
  CASE      $DAC OF {
              $PDA-FPDA:
              REQ  Generator-for-subordinates {$SnakingColumnsGFS},
              REQ  Position               {REQ #variable-position {
                                              PERM #offset {ANY_VALUE},
                                              PERM #separation {ANY_VALUE},
                                              PERM #alignment {ANY_VALUE},
                                              PERM #fill-order {'normal-order'}}},
              PERM Balance                {ANY_VALUE},
              CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {
              {'270-degrees'}: -- page layout A --
                    REQ  Dimensions       {REQ #horizontal-dimension
                                              {REQ #fixed-dimension {ANY_VALUE}
                                              |REQ #maximum-size {'applies'}},
                                               REQ #vertical-dimension
                                              {REQ #rule-b {ANY_VALUE}}},
                    REQ  Layout-path  {'0-degrees'|'180-degrees'}
              {'0-degrees'}: -- page layout B --
                    REQ  Dimensions       {REQ #horizontal-dimension
                                              {REQ #rule-b {ANY_VALUE}},
                                           REQ #vertical-dimension
                                              {REQ #fixed-dimension {ANY_VALUE}
                                              |REQ #maximum-size {'applies'}}},
                    PERM Layout-path      {'90-degrees'|'270-degrees'}
              {'180-degrees'}: -- page layouts C and D --
                    REQ  Dimensions       {REQ #horizontal-dimension
                                              {REQ #rule-b {ANY_VALUE}},
                                           REQ #vertical-dimension
                                              {REQ #fixed-dimension {ANY_VALUE}
                                              |REQ #maximum-size {'applies'}}},
                    PERM Layout-path  {'270-degrees'}
              }
              },
  REQ       Application-comments          {REQ #constraint-name {"10"},
                                           PERM #external-data {ANY_VALUE}}
```

```
SPECIFIC:
  CASE      $DAC OF {
              $FDA: PERM  Object-class      {OBJECT_CLASS_ID_OF(SnakingColumns)}
              $FPDA: REQ  Object-class      {OBJECT_CLASS_ID_OF(SnakingColumns)}
                  },
  REQ       Subordinates                    {SUB_ID_OF(ColumnVariable)+},
  PERM      Application-comments            {REQ #constraint-name {"10"},
                                             PERM #external-data {ANY_VALUE}}}
```

7.4.3.11 ColumnVariable: ANY-FRAME-VARIABLE {

```
GENERIC:
  CASE      $DAC OF {
              $PDA-FPDA:
              PERM  Permitted-categories    {ANY_STRING},
              REQ   Position                {REQ #variable-position {
                                                 PERM #offset {ANY_VALUE},
                                                 PERM #separation {ANY_VALUE},
                                                 PERM #alignment {ANY_VALUE},
                                                 PERM #fill-order {'normal-order'}}},
              CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {
              {'270-degrees'}: -- page layout A --
                  REQ  Dimensions           {REQ #horizontal-dimension
                                                 {REQ #fixed-dimension {ANY_VALUE}},
                                            REQ #vertical-dimension
                                                 {REQ #rule-b {ANY_VALUE}
                                                 |REQ #maximum-size {'applies'}}},
                  PERM Layout-path          {'270 degrees''}
              {'0-degrees'}: -- page layout B --
                  REQ  Dimensions           {REQ #horizontal-dimension
                                                 {REQ #rule-b {ANY_VALUE}
                                                 |REQ #maximum-size {'applies'}},
                                            REQ #vertical-dimension
                                                 {REQ #fixed-dimension {ANY_VALUE}}},
                  REQ  Layout-path          {'0-degrees'}
              {'180-degrees'}: -- page layouts C and D --
                  REQ  Dimensions           {REQ #horizontal-dimension
                                                 {REQ #rule-b {ANY_VALUE}
                                                 |REQ #maximum-size {'applies'}},
                                            REQ #vertical-dimension
                                                 {REQ #fixed-dimension {ANY_VALUE}}},
                  REQ  Layout-path  {'180-degrees'}
            }},
  REQ       Application-comments            {REQ #constraint-name {"9"},
                                             PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM  Object-class        {OBJECT_CLASS_ID_OF(ColumnVariable)}
            $FPDA: REQ  Object-class        {OBJECT_CLASS_ID_OF(ColumnVariable)}
                  },
  REQ       Subordinates                    {SUB_ID_OF(SpecificBlock)+},
  PERM      Application-comments            {REQ #constraint-name {"9"},
                                             PERM #external-data {ANY_VALUE}}}
```

7.4.3.12 ColumnFixed: ANY-FRAME-VARIABLE {

```
GENERIC:
  CASE      $DAC OF {
            $PDA-FPDA:
              REQ   Permitted-categories    {ANY_STRING},
              REQ   Position                {REQ #fixed-position
                                                 {REQ #horizontal-position {ANY_VALUE},
                                                  REQ #vertical-position {ANY_VALUE}}},
              CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {
              {'270-degrees'}: -- page layout A --
                  REQ  Dimensions           {REQ #horizontal-dimension
                                                 {REQ #fixed-dimension {ANY_VALUE}
                                                 |REQ #maximum-size {'applies'}},
                                            REQ #vertical-dimension
```

```
                                        {REQ #rule-b {ANY_VALUE}
                                        |REQ #maximum-size {'applies'}}},
            PERM Layout-path            {'270-degrees'}
        {'0-degrees'}: -- page layout B --
            REQ  Dimensions             {REQ #horizontal-dimension
                                        {REQ #rule-b {ANY_VALUE}
                                        |REQ #maximum-size {'applies'}},
                                        REQ #vertical-dimension
                                        {REQ #fixed-dimension {ANY_VALUE}
                                        |REQ #maximum-size {'applies'}}},
            REQ  Layout-path            {'0-degrees'}
        {'180-degrees'}: -- page layouts C and D --
            REQ  Dimensions             {REQ #horizontal-dimension
                                        {REQ #maximum-size {'applies'}},
                                        REQ #vertical-dimension
                                        {REQ #fixed-dimension {ANY_VALUE}
                                        |REQ #maximum-size {'applies'}}},
            REQ  Layout-path            {'180-degrees'}
        }
        },
  REQ   Application-comments            {REQ #constraint-name {"8"},
                                         PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE  $DAC OF {
        $FDA:  PERM  Object-class       {OBJECT_CLASS_ID_OF(ColumnFixed)}
        $FPDA: REQ   Object-class       {OBJECT_CLASS_ID_OF(ColumnFixed)}
        },
  REQ   Subordinates                    {SUB_ID_OF(SpecificBlock)+},
  PERM  Application-comments            {REQ #constraint-name {"8"},
                                         PERM #external-data {ANY_VALUE}}}
```

7.4.3.13 FootnoteArea: ANY-FRAME-VARIABLE {

```
GENERIC:
  CASE  $DAC OF {
        $PDA-FPDA:
          REQ   Position                {REQ #variable-position {
                                        PERM #offset {ANY_VALUE},
                                        PERM #separation {ANY_VALUE},
                                        PERM #alignment {ANY_VALUE},
                                        REQ  #fill-order {'reverse-order'}}},
          REQ   Permitted-categories {"Footnote"},
          CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {
          {'270-degrees'}: -- page layout A --
            REQ  Dimensions             {REQ #horizontal-dimension
                                        {REQ #fixed-dimension {ANY_VALUE}
                                        |REQ #maximum-size {'applies'}},
                                        REQ #vertical-dimension
                                        {REQ #rule-b {ANY_VALUE}}},
            PERM Layout-path            {'270-degrees'}
          {'0-degrees'}: -- page layout B --
            REQ  Dimensions             {REQ #horizontal-dimension
                                        {REQ #rule-b {ANY_VALUE}},
                                        REQ #vertical-dimension
                                        {REQ #fixed-dimension {ANY_VALUE}
                                        |REQ #maximum-size {'applies'}}},
            REQ  Layout-path            {'0-degrees'}
          {'180-degrees'}: -- page layouts C and D --
            REQ  Dimensions             {REQ #horizontal-dimension
                                        {REQ #rule-b {ANY_VALUE}},
                                        REQ #vertical-dimension
                                        {REQ #fixed-dimension {ANY_VALUE}
                                        |REQ #maximum-size {'applies'}}},
            REQ  Layout-path            {'180-degrees'}
        }
        },
  REQ   Application-comments            {REQ #constraint-name {"15"},
                                         PERM #external-data {ANY_VALUE}}
```

```
SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM Object-class      {OBJECT_CLASS_ID_OF(FootnoteArea)}
            $FPDA: REQ Object-class      {OBJECT_CLASS_ID_OF(FootnoteArea)}
            },
  REQ       Subordinates                 {SUB_ID_OF(SpecificBlock)+},
  PERM      Application-comments         {REQ #constraint-name {"15"},
                                          PERM #external-data {ANY_VALUE}}}
```

7.4.3.14 BasicHeader: ANY-FRAME-FIXED {

```
GENERIC:
  CASE      $DAC OF{
            $PDA-FPDA:
  REQ       Logical-source               {OBJECT_CLASS_ID_OF(CommonContent)}},
  PERM      Layout-path                  {'270-degrees' -- page layouts A,B,C --
                                         |'180-degrees' -- page layout D --},
  REQ       Application-comments         {REQ #constraint-name {"27"},
                                          PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM Object-class      {OBJECT_CLASS_ID_OF(BasicHeader)}
            $FPDA: REQ Object-class      {OBJECT_CLASS_ID_OF(BasicHeader)}
            },
  REQ       Subordinates                 {SUB_ID_OF(SpecificBlock)+},
  PERM      Application-comments         {REQ #constraint-name {"27"},
                                          PERM #external-data {ANY_VALUE}}}
```

7.4.3.15 BasicFooter: ANY-FRAME-FIXED {

```
GENERIC:
  CASE      $DAC OF{
            $PDA-FPDA:
  REQ       Logical-source               {OBJECT_CLASS_ID_OF(CommonContent)}},
  PERM      Layout-path                  {'270-degrees' -- page layouts A,B,C --
                                         |'180-degrees' -- page layout D --},
  REQ       Application-comments         {REQ #constraint-name {"33"},
                                          PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM Object-class      {OBJECT_CLASS_ID_OF(BasicFooter)}
            $FPDA: REQ Object-class      {OBJECT_CLASS_ID_OF(BasicFooter)}
            },
  REQ       Subordinates                 {SUB_ID_OF(SpecificBlock)+},
  PERM      Application-comments         {REQ #constraint-name {"33"},
                                          PERM #external-data {ANY_VALUE}}}
```

7.4.3.16 CompositeHeader: ANY-FRAME-FIXED {

```
GENERIC:
  CASE      $DAC OF {
            $PDA-FPDA:
  REQ       Generator-for-subordinates   {$HeaderFooterGFS}
            },
  PERM      Layout-path                  {'270-degrees' -- page layouts A,B,C --
                                         |'180-degrees' -- page layout D --},
  REQ       Application-comments         {REQ #constraint-name {"5"},
                                          PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM Object-class      {OBJECT_CLASS_ID_OF(CompositeHeader)}
            $FPDA: REQ Object-class      {OBJECT_CLASS_ID_OF(CompositeHeader)}
            },
  REQ       Subordinates                 {SUB_ID_OF(SourcedContentFixed)+,
                                          SUB_ID_OF(ArrangedContentFixed)+,
                                          SUB_ID_OF(SourcedContentVariable)+,
                                          SUB_ID_OF(ArrangedContentVariable)+},
```

```
  PERM      Application-comments          {REQ #constraint-name {"5"},
                                           PERM #external-data {ANY_VALUE}}}
```

7.4.3.17 CompositeFooter: ANY-FRAME-FIXED {

```
GENERIC:
  CASE      $DAC OF {
            $PDA-FPDA:
  REQ       Generator-for-subordinates    {$HeaderFooterGFS}
            },
  PERM      Layout-path                   {'270-degrees' -- page layouts A,B,C --
                                          |'180-degrees' -- page layout D --},
  REQ       Application-comments          {REQ #constraint-name {"32"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM Object-class       {OBJECT_CLASS_ID_OF(CompositeFooter)}
            $FPDA: REQ Object-class       {OBJECT_CLASS_ID_OF(CompositeFooter)}
            },
  REQ       Subordinates                  {SUB_ID_OF(SourcedContentFixed)+,
                                           SUB_ID_OF(ArrangedContentFixed)+,
                                           SUB_ID_OF(SourcedContentVariable)+,
                                           SUB_ID_OF(ArrangedContentVariable)+},
  PERM      Application-comments          {REQ #constraint-name {"32"},
                                           PERM #external-data {ANY_VALUE}}}
```

7.4.3.18 SourcedContentVariable: ANY-FRAME-VARIABLE {

```
GENERIC:
  CASE      $DAC OF {
            $PDA-FPDA:
              REQ Logical-source        {OBJECT_CLASS_ID_OF(CommonContent)},
              REQ Position              {REQ #variable-position {
                                                PERM #offset {ANY_VALUE},
                                                PERM #separation {ANY_VALUE},
                                                PERM #alignment {ANY_VALUE},
                                                PERM #fill-order {'normal-order'}}},
              CASE SUPERIOR (CompositeHeader|CompositeFooter
                                (Layout-path)) OF {
              {'270-degrees'}:
                   REQ Dimensions       {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #maximum-size {'applies'}},
                                         REQ #vertical-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #rule-b {ANY_VALUE}}},
                   PERM Layout-path     {'270-degrees'}
              {'180-degrees'}:
                   REQ Dimensions       {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #rule-b {ANY_VALUE}},
                                         REQ #vertical-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #maximum-size {'applies'}}},
                   REQ Layout-path      {'180-degrees'}
            }
            },
  REQ       Application-comments          {REQ #constraint-name {"19"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM Object-class       {OBJECT_CLASS_ID_OF(SourcedContentVariable)}
            $FPDA: REQ Object-class       {OBJECT_CLASS_ID_OF(SourcedContentVariable)}
            },
  REQ       Subordinates                  {SUB_ID_OF(SpecificBlock)+},
  PERM      Application-comments          {REQ #constraint-name {"19"},
                                           PERM #external-data {ANY_VALUE}}}
```

7.4.3.19 ArrangedContentVariable: ANY-FRAME-VARIABLE {

```
GENERIC:
  CASE      $DAC OF {
            $PDA-FPDA:
              REQ  Generator-for-subordinates
                                          {<construction-expr>::=SEQ
                                          (OBJECT_CLASS_ID_OF(GenericBlock)...);},
              REQ  Position               {REQ #variable-position {
                                                PERM #offset {ANY_VALUE},
                                                PERM #separation {ANY_VALUE},
                                                PERM #alignment {ANY_VALUE},
                                                PERM #fill-order {'normal-order'}}},
              REQ  Dimensions             {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}},
                                          REQ #vertical-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}}
                   }},
  REQ       Application-comments          {REQ #constraint-name {"17"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
CASE        $DAC OF {
              $FDA: PERM  Object-class    {OBJECT_CLASS_ID_OF(ArrangedContentVariable)}
              $FPDA: REQ  Object-class    {OBJECT_CLASS_ID_OF(ArrangedContentVariable)}
            },
  REQ       Subordinates                  {SUB_ID_OF(GenericBlock)+},
  PERM      Application-comments          {REQ #constraint-name {"17"},
                                           PERM #external-data {ANY_VALUE}}}
```

7.4.3.20 SourcedContentFixed: ANY-FRAME-VARIABLE {

```
GENERIC:
  CASE      $DAC OF {
            $PDA-FPDA:
              REQ   Logical-source        {OBJECT_CLASS_ID_OF(CommonContent)},
              REQ   Position              {REQ #fixed-position
                                                {REQ #horizontal-position{ANY_VALUE},
                                                 REQ #vertical-position{ANY_VALUE}}},
              REQ  Dimensions             {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}},
                                           REQ #vertical-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #rule-b {ANY_VALUE}}},
              CASE SUPERIOR (CompositeHeader|CompositeFooter
                                          (Layout-path)) OF {
              {'270-degrees'}:
                   PERM Layout-path       {'270-degrees'}
              {'180-degrees'}:
                   REQ  Layout-path       {'180-degrees'}
            }
            },
  REQ       Application-comments          {REQ #constraint-name {"18"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM  Object-class      {OBJECT_CLASS_ID_OF(SourcedContentFixed)}
            $FPDA: REQ  Object-class      {OBJECT_CLASS_ID_OF(SourcedContentFixed)}
            },
  REQ       Subordinates                  {SUB_ID_OF(SpecificBlock)+},
  PERM      Application-comments          {REQ #constraint-name {"18"},
                                           PERM #external-data {ANY_VALUE}}}
```

7.4.3.21 ArrangedContentFixed: ANY-FRAME-FIXED {

```
GENERIC:
  CASE      $DAC OF{
            $PDA-FPDA:
  REQ       Generator-for-subordinates    {<construction-expr>::=SEQ
                                          (OBJECT_CLASS_ID_OF(GenericBlock)...);}},
```

```
  REQ       Application-comments          {REQ #constraint-name {"16"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM Object-class       {OBJECT_CLASS_ID_OF(ArrangedContentFixed)}
            $FPDA: REQ Object-class       {OBJECT_CLASS_ID_OF(ArrangedContentFixed)}
            },
  REQ       Subordinates                  {SUB_ID_OF(GenericBlock)+},
  PERM      Application-comments          {REQ #constraint-name {"16"},
                                           PERM #external-data {ANY_VALUE}}}
```

7.4.3.22 GenericBlock: BLOCK {

```
GENERIC:
  REQ       Object-class-identifier       {ANY_VALUE},
  REQ       Object-type                   {'block'},
  REQ       Content-architecture-class    {$FC|$FPC|$FPR|$FPG},
  PERM      Resource                      {ANY_VALUE},
  PERM      Content-portions              {CONTENT_ID_OF
                                           (Character-content-portion)+
                                           |CONTENT_ID_OF
                                           (Raster-graphics-content-portion)
                                           |CONTENT_ID_OF
                                           (Geometric-graphics-content-portion)},
  PERM      User-readable-comments        {ANY_STRING},
  PERM      User-visible-name             {ANY_STRING},
  PERM      Position                      {REQ #fixed-position
                                             {REQ #horizontal-position {ANY_VALUE},
                                              REQ #vertical-position {ANY_VALUE}}},
  PERM      Dimensions                    {REQ #horizontal-dimension
                                             {REQ #fixed-dimension {ANY_VALUE}},
                                           REQ #vertical-dimension
                                             {REQ #fixed-dimension {ANY_VALUE}}},
  REQ       Application-comments          {REQ #constraint-name {"29"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE      $DAC OF {
            $FDA: PERM Object-class       {OBJECT_CLASS_ID_OF(GenericBlock)}
            $FPDA: REQ Object-class       {OBJECT_CLASS_ID_OF(GenericBlock)}
            },
  CASE      GenericBlock(object-class)    OF {VOID:
            REQ Content-portions          {CONTENT_ID_OF(Character-content-portion)+
                                           |CONTENT_ID_OF(Raster-graphics-content-portion)
                                           |CONTENT_ID_OF(Geometric-graphics-content-portion)}
            },
  PERM      Application-comments          {REQ #constraint-name {"29"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
  PERM      Presentation-style            {STYLE_ID_OF(P-Style1)
                                           |STYLE_ID_OF(P-Style4)
                                           |STYLE_ID_OF(P-Style3)}}
```

7.4.3.23 SpecificBlock: BLOCK {

```
SPECIFIC:
  PERM      Presentation-style            {STYLE_ID_OF(P-Style1)
                                           |STYLE_ID_OF(P-Style4)
                                           |STYLE_ID_OF(P-Style3)
                                           |STYLE_ID_OF(P-Style2)},
  REQ       Content-portions              {CONTENT_ID_OF(Character-content-portion)+
                                           |CONTENT_ID_OF(Raster-graphics-content-portion)
                                           |CONTENT_ID_OF(Geometric-graphics-content-portion)},
  PERM      Application-comments          {REQ #constraint-name {"30"},
                                           PERM #external-data {ANY_VALUE}}}
```

7.5 Layout style constituent constraints

7.5.1 Macro definitions

```
DEFINE(LayoutObjectClasses,  "
                    OBJECT_CLASS_ID_OF(PageSet)
                    |OBJECT_CLASS_ID_OF(Page)
                    |OBJECT_CLASS_ID_OF(RectoPage)
                    |OBJECT_CLASS_ID_OF(VersoPage)
                    |OBJECT_CLASS_ID_OF(BasicBody)
                    |OBJECT_CLASS_ID_OF(VariableCompositeBody)
                    |OBJECT_CLASS_ID_OF(BasicFloat)
                    |OBJECT_CLASS_ID_OF(SnakingColumns)
                    |OBJECT_CLASS_ID_OF(SynchronizedColumns)
                    |OBJECT_CLASS_ID_OF(ColumnFixed)
                    |OBJECT_CLASS_ID_OF(ColumnVariable)
                        ")

DEFINE(SameLayoutObject, "
            REQ{REQ #logical-object{<object-id-expr>::=PREC-OBJ(CURR-OBJ);}
            |REQ#logical-object{'null'}},
            PERM #layout-object{'page'}
                ")
```

7.5.2 Factor constraints

7.5.2.1 FACTOR ANY-LAYOUT-STYLE {

```
  REQ       Layout-style-identifier          {ANY_VALUE},
  PERM      User-readable-comments           {ANY_STRING},
  PERM      User-visible-name                {ANY_STRING}}
```

7.5.3 Constituent constraints

7.5.3.1 L-Style1: ANY-LAYOUT-STYLE {

-- this style is used for the constituent constraint Passage only --

```
  CASE DocumentProfile(Generic-layout-structure) OF {
      {'complete-generator-set'}:
            PERM  Layout-object-class      {OBJECT_CLASS_ID_OF(PageSet)},
            PERM  New-layout-object        {OBJECT_CLASS_ID_OF(PageSet)},
            PERM  Indivisibility           {$LayoutObjectClasses
            |ANY_STRING|'page'|'null'}
      VOID:
            PERM  Indivisibility           {ANY_STRING|'page'|'null'}
                                    }}
```

-- ANY_STRING is interpreted as representing the name of a layout category --

7.5.3.2 L-Style2: ANY-LAYOUT-STYLE {

-- this style is used for the constituent constraints BodyText, and Number --

```
  CASE DocumentProfile(Generic-layout-structure) OF {
      {'complete-generator-set'}:
            PERM  Indivisibility           {$LayoutObjectClasses
                                           |ANY_STRING|'page'|'null'},
            PERM  New-layout-object        {$LayoutObjectClasses
                                           |ANY_STRING|'page'|'null'}
      VOID:
            PERM  Indivisibility           {ANY_STRING|'page'|'null'},
            PERM  New-layout-object        {ANY_STRING|'page'|'null'}
                                    },
  PERM      Layout-category                {ANY_STRING},
  PERM      Same-layout-object             {$SameLayoutObject},
  PERM      Concatenation                  {ANY_VALUE},
  PERM      Offset                         {ANY_VALUE},
  PERM      Separation                     {PERM #leading-edge{ANY_INTEGER},
                                            PERM #trailing-edge{ANY_INTEGER}},
  PERM      Block-alignment                {ANY_VALUE},
  PERM      Synchronization                {ANY_VALUE}}
```

7.5.3.3 L-Style3: ANY-LAYOUT-STYLE {

-- this style is used for the constituent constraints CommonText and PageNumber --

```
PERM    Concatenation                   {ANY_VALUE},
PERM    Offset                          {ANY_VALUE},
PERM    Block-alignment                 {ANY_VALUE},
PERM    Separation                      {PERM #leading-edge{ANY_INTEGER},
                                         PERM #trailing-edge{ANY_INTEGER}}}
```

7.5.3.4 L-Style4: ANY-LAYOUT-STYLE {

-- this style is used for the constituent constraints NumberedSegment and Paragraph --

```
CASE DocumentProfile(Generic-layout-structure) OF {
    {'complete-generator-set'}:
        PERM  Indivisibility            {$LayoutObjectClasses
                                        |ANY_STRING|'page'|'null'},
        PERM  Layout-object-class       {OBJECT_CLASS_ID_OF(PageSet)},
        PERM  New-layout-object         {$LayoutObjectClasses
                                        |ANY_STRING|'page'|'null'}
    VOID:
        PERM  Indivisibility            {ANY_STRING|'page'|'null'},
        PERM  New-layout-object         {ANY_STRING|'page'|'null'}
                                    },
PERM    Same-layout-object              {$SameLayoutObject},
PERM    Synchronization                 {ANY_VALUE}}
  -- ANY_STRING is interpreted as representing the name of a layout category --
```

7.5.3.5 L-Style5: ANY-LAYOUT-STYLE {

-- this style is used for the constituent constraints BodyRaster and BodyGeometric --

```
CASE DocumentProfile(Generic-layout-structure) OF {
    {'complete-generator-set'}:
        PERM  New-layout-object         {$LayoutObjectClasses
                                        |ANY_STRING|'page'|'null'}
    VOID:
        PERM  New-layout-object         {ANY_STRING|'page'|'null'}
        },
  -- ANY_STRING is interpreted as representing the name of a layout category --
PERM    Layout-category                 {ANY_STRING},
PERM    Offset                          {ANY_VALUE},
PERM    Same-layout-object              {$SameLayoutObject},
PERM    Separation                      {PERM #leading-edge {ANY_INTEGER},
                                         PERM #trailing-edge {ANY_INTEGER}},
PERM    Block-alignment                 {ANY_VALUE},
PERM    Synchronization                 {ANY_VALUE}}
```

7.5.3.6 L-Style6: ANY-LAYOUT-STYLE {

-- this style is used for the constituent constraint FootnoteText --

```
REQ     Layout-category                 {"Footnote"},
PERM    Concatenation                   {ANY_VALUE},
CASE DocumentProfile(Generic-layout-structure) OF {
    {'complete-generator-set'}:
    PERM  Indivisibility                {OBJECT_CLASS_ID_OF(FootnoteArea)
                                        |'page'|'null'}
VOID:
    PERM  Indivisibility                {ANY_STRING|'page'|'null'}
                    },
PERM    Offset                          {ANY_VALUE},
PERM    Block-alignment                 {ANY_VALUE},
PERM    Separation                      {PERM #leading-edge {ANY_INTEGER},
                                         PERM #trailing-edge {ANY_INTEGER}}}
```

7.5.3.7 L-Style7: ANY-LAYOUT-STYLE {

-- this style is used for the constituent constraint Footnote only --

```
PERM    Same-layout-object              {$SameLayoutObject}}
```

7.5.3.8 L-Style8: ANY-LAYOUT-STYLE {

-- this style is used for the constituent constraints CommonRaster and CommonGeometric --

```
PERM    Offset                          {ANY_VALUE},
PERM    Block-alignment                 {ANY_VALUE},
PERM    Separation                      {PERM #leading-edge {ANY_INTEGER},
                                         PERM #trailing-edge {ANY_INTEGER}}}
```

7.5.3.9 L-Style9: ANY-LAYOUT-STYLE {

-- this style is used for the constituent constraint FootnoteNumber --

```
REQ     Layout-category                 {"Footnote"},
PERM    Offset                          {ANY_VALUE},
PERM    Block-alignment                 {ANY_VALUE},
PERM    Separation                      {PERM #leading-edge {ANY_INTEGER},
                                         PERM #trailing-edge {ANY_INTEGER}}}
```

7.5.3.10 L-Style10: ANY-LAYOUT-STYLE {

-- this style is used for the constituent constraints FootnoteReference only --

```
CASE DocumentProfile(Generic-layout-structure) OF {
    {'complete-generator-set'}:
    PERM  Indivisibility                {$LayoutObjectClasses
                                        |ANY_STRING|'page'|'null'}
VOID:
    PERM  Indivisibility                {ANY_STRING|'page'|'null'}
                                    },
PERM    Layout-category                 {ANY_STRING},
PERM    Same-layout-object              {$SameLayoutObject},
PERM    Concatenation                   {ANY_VALUE},
PERM    Offset                          {ANY_VALUE},
PERM    Separation                      {PERM #leading-edge {ANY_INTEGER},
                                         PERM #trailing-edge {ANY_INTEGER}},
PERM    Block-alignment                 {ANY_VALUE}}
```

7.5.3.11 L-Style11: ANY-LAYOUT-STYLE {

-- this style is used for the constituent constraint Footnotebody --

```
PERM    Same-layout-object              {$SameLayoutObject},
CASE DocumentProfile(Generic-layout-structure) OF {
    {'complete-generator-set'}:
    PERM  Indivisibility                {OBJECT_CLASS_ID_OF(FootnoteArea)
                                        |'page'|'null'}
VOID:
    PERM  Indivisibility                {ANY_STRING|'page'|'null'}}}
```

7.6 Presentation style constituent contraints

7.6.1 Macro definitions

-- No macro definitions are applicable to this subclause. --

7.6.2 Factor constraints

7.6.2.1 FACTOR ANY-PRESENTATION-STYLE {

```
REQ     Presentation-style-identifier   {ANY_VALUE},
PERM    User-visible-name               {ANY_STRING},
PERM    Border                          {ANY_VALUE},
PERM    User-readable-comments          {ANY_STRING}}
```

7.6.3 Constituent constraints

7.6.3.1 P-Style1: ANY-PRESENTATION-STYLE {

-- this style is used for the constituent constraints BodyText, Number, FootnoteNumber, FootnoteReference,
-- FootnoteText, GenericBlock and SpecificBlock --

```
PERM  Presentation-attributes  {
  PERM #character-attributes   {
    PERM #alignment                          {ANY_VALUE},
    PERM #character-fonts                    {ANY_VALUE},
    PERM #character-orientation              {'0-degrees'
                                             |'90-degrees'},
    PERM #character-path                     {ANY_VALUE},
    PERM #character-spacing                  {ANY_VALUE},
    PERM #code-extension-announcers          {$CDEXTEN},
    PERM #first-line-offset                  {ANY_VALUE},
    PERM #graphic-character-sets             {$PERMIT-GRCHAR},
    PERM #graphic-character-sub-repertoire   {ANY_VALUE},
    PERM #graphic-rendition                  {$GRAPHICRENDITIONS},
    PERM #indentation                        {ANY_VALUE},
    PERM #itemization                        {ANY_VALUE},
    PERM #kerning-offset                     {ANY_VALUE},
    PERM #line-layout-table                  {ANY_VALUE},
    PERM #line-progression                   {ANY_VALUE},
    PERM #line-spacing                       {ANY_VALUE},
    PERM #orphan-size                        {ANY_VALUE},
    PERM #proportional-line-spacing          {ANY_VALUE},
    PERM #widow-size                         {ANY_VALUE}}}}
```

7.6.3.2 P-Style2: ANY-PRESENTATION-STYLE {

-- this style is used for the constituent constraints CommonText, PageNumber and SpecificBlock --

```
PERM  Presentation-attributes  {
  PERM #character-attributes   {
    PERM #alignment                          {ANY_VALUE},
    PERM #character-fonts                    {ANY_VALUE},
    PERM #character-orientation              {'0-degrees'
                                             |'90-degrees'},
    PERM #character-path                     {'0-degrees'
                                             |'180-degrees'
                                             |'270-degrees'},
    PERM #character-spacing                  {ANY_VALUE},
    PERM #code-extension-announcers          {$CDEXTEN},
    PERM #first-line-offset                  {ANY_VALUE},
    PERM #graphic-character-sets             {$PERMIT-GRCHAR},
    PERM #graphic-character-sub-repertoire   {ANY_VALUE},
    PERM #graphic-rendition                  {$GRAPHICRENDITIONS},
    PERM #indentation                        {ANY_VALUE},
    PERM #itemization                        {ANY_VALUE},
    PERM #kerning-offset                     {ANY_VALUE},
    PERM #line-progression                   {ANY_VALUE},
    PERM #line-spacing                       {ANY_VALUE},
    PERM #line-layout-table                  {ANY_VALUE},
    PERM #proportional-line-spacing          {ANY_VALUE}}}}
```

7.6.3.3 P-Style3: ANY-PRESENTATION-STYLE {

-- this style is used for the constituent constraints BodyRaster, CommonRaster, GenericBlock and SpecificBlock --

```
PERM  Presentation-attributes  {
  PERM #raster-graphics-attributes   {
    PERM #image-dimensions                   {ANY_VALUE},
    PERM #clipping                           {ANY_VALUE},
    PERM #pel-spacing                        {ANY_VALUE},
    PERM #spacing-ratio                      {ANY_VALUE}}}}
```

7.6.3.4 P-Style4: ANY-PRESENTATION-STYLE {

-- this style is used for the constituent constraints BodyGeometric, CommonGeometric, GenericBlock and SpecificBlock --

```
PERM  Presentation-attributes  {
  PERM #geometric-graphics-attributes {
    PERM #picture-dimensions                 {ANY_VALUE},
    PERM #picture-orientation                {ANY_VALUE},
    PERM #text-rendition                     {PERM #fonts-list {ANY_VALUE},
```

PERM #character-set-list {ANY_VALUE}}}}}

7.7 Content portion constituent constraints

7.7.1 Macro definitions

-- No macro definitions are applicable to this subclause. --

7.7.2 Factor constraints

7.7.2.1 FACTOR ANY-CONTENT {

CASE $DAC OF {
$FDA:
REQ Content-identifier-layout {ANY_VALUE}

$PDA:
REQ Content-identifier-logical {ANY_VALUE}
-- This attribute is specified, if the content portion is associated with a basic logical object
-- or a basic logical object class. --
|REQ Content-identifier-layout {ANY_VALUE}
-- This attribute is specified, if the content portion is associated with a basic layout object class. --

$FPDA:
REQ Content-identifier-layout {ANY_VALUE},
REQ Content-identifier-logical {ANY_VALUE}
-- Both attributes are specified, if the content portion associated with a basic logical object
-- and a basic layout object. --
|REQ Content-identifier-layout {ANY_VALUE}
-- This attribute is specified, if the content portion is associated with a basic layout object class. --
|REQ Content-identifier-logical {ANY_VALUE}
-- This attribute is specified, if the content portion is associated with a basic logical object class --
}}

7.7.3 Constituent constraints

7.7.3.1 Character-content-portion: ANY-CONTENT {

PERM Type-of-coding {ASN.1{2 8 3 6 0}},
PERM Alternative-representation {ANY_STRING},
PERM Content-information
{CHARACTER {#STAB {ANY_VALUE}
|#SHS {0,1,2,3,4}
|#SGR {$GRAPHICRENDITIONS}
|#SVS {ANY_VALUE}
|#SLS {ANY_VALUE}
|#SCS {ANY_VALUE}
|#SRS {ANY_VALUE}
|#JFY {0}
|#CR
|#LF
|#VPB
|#VPR
|#PLD
|#PLU
|#SUB
|#BPH
|#NBH
|#SOS
|#ST
|#SP
|#LS0
|#LS1R
|#LS2R
|#LS3R
|#SS2
|#SS3
|#ESC{$DEG-CORE-G0}
|#ESC{$DEG-646-G0}

|#ESC{$DEG-ANY-G1}
|#ESC{$DEG-ANY-G2}
|#ESC{$DEG-ANY-G3}
|#ESC{$DEG-EMPTY-G1}
}...}}

7.7.3.2 Raster-graphics-content-portion: ANY-CONTENT {

PERM Type-of-coding {ASN.1{2 8 3 7 0} -- *T.6 encoding* --
|ASN.1{2 8 3 7 1} -- *T.4 one-dimensional encoding* --
|ASN.1{2 8 3 7 2} -- *T.4 two dimensional encoding* --
|ASN.1{2 8 3 7 3} -- *bitmap encoding* --**},**
PERM Coding-attributes{
PERM #raster-graphics-coding-attributes{
PERM #number-of-lines {>0},
REQ #number-of-pels-per-line {>=0},
PERM #compression {ANY_VALUE}}},
PERM Alternative-representation {ANY_STRING},
PERM Content-information {RASTER}}

7.7.3.3 Geometric-graphics-content-portion: ANY-CONTENT {

PERM Type-of-coding {ASN.1{2 8 3 8 0}},
PERM Alternative-representation {ANY_STRING},
PERM Content-information {GEOMETRIC}}

8 Interchange format

8.1 Interchange format

For conformance to this Profile, the ODIF interchange format class A shall be used. This form of ODIF is defined in CCITT Rec T.415 | ISO/IEC 8613-5. The value of the document profile attribute “interchange format” is 'if-a'.

8.2 Document application profile object identifier

The value for the document profile attribute “document application profile” for this interchange format is represented by the following object identifier: ASN.1 {2 8 4 0 26 0}

8.3 Encoding of application comments

The encoding of the attribute “application comments” is defined in this encoding as an octet string as specified in CCITT Rec. T.415 | ISO/IEC 8613-5. This document application profile requires that the encoding within that octet string be in accordance with the ASN.1 syntax specified in the following module definition:

```
FOD-DAPSpecification
DEFINITIONS ::= BEGIN
EXPORTS Appl-Comm-Encoding;

Appl-Comm-Encoding ::= SEQUENCE {
constraint-name [0] IMPLICIT PrintableString OPTIONAL,
external-data [1] IMPLICIT OCTET STRING OPTIONAL }
END
```

8.4 Data lengths

The maximum length of data values of the type OCTET STRING, as defined in CCITT Rec. X.208 | ISO/IEC 8824 in data streams which may be encoded in accordance with this document application profile is 32 767 octets. If it is required to encode an octet string of greater length than this, constructed type encoding shall be used. That is, data values greater than 32 767 in length shall be split into a sequence of strings shorter than 32 767, each of which is encoded using a primitive type.

Annex A

Amendments and technical corrigenda

(This annex forms an integral part of this Recommendation)

A.1 Amendments

A.1.1 Amendments to the base standard

The amendments applicable to this Recommendation include text to be included in CCITT Rec. T.411 | ISO/IEC 8613-1 as the following annexes:

- Annex E: Use of ISO/IEC 10021 (MOTIS) to interchange documents conforming to CCITT Rec. T.410 Series | ISO 8613 – published as [First extension to the CCITT Rec. T.410 Series (1988), Addendum II | ISO 8613-1 (1989): Amendment 1];
- Annex F: Document Application Profile Proforma and Notation – published as CCITT Rec. T.411 – Annex F (1991) | ISO 8613-1 (1989): Addendum 1;
- Annex G: Conformance testing methodology – published as ISO 8613-1 (1989): Amendment 2;
- Annex H: Recording of documents conforming to CCITT Rec. T.410 Series | ISO/IEC 8613 on flexible disk cartridges conforming to ISO 9293 – currently a Draft Amendment 5 to ISO 8613-1 (1989) – Annex H.

This Recommendation does not include the following features of the amendment:

- Addendum on security – published as [First extension to the CCITT Rec. T.410 Series (1988), Addendum V | ISO 8613 (1989) Addendum 4];
- Addendum on styles – published as [First extension to the CCITT Rec. T.410 Series (1988), Addendum IV | ISO 8613 (1989) Addendum 6];
- Addendum on alternative representation – published as [First extension to the CCITT Rec. T.410 Series (1988), Addendum III | ISO 8613 (1989) Addendum 3];
- Addendum on colour – published as [Revision of the CCITT Rec. T.410 Series contained in COM VIII – R30-E Addendum 2 | ISO 8613 (1989) – Amendment 2];
- Addendum on tiled raster graphics – published as [First extension to the CCITT Rec. T.410 Series (1988) Addendum I | ISO 8613 (1989) – Addendum 1];
- Addendum on streams – published as ISO 8613 (1989) – Amendment 5.

A.1.2 Proposed changes to standards due to defects

This amendment addresses the inclusion of the CCITT Rec. T.410 Series (1988) | ISO/IEC 8613 (1988) – Technical corrigenda 1, 2 and 3.

A.2 Technical corrigenda

A.2.1 Technical corrigenda to this Recommendation

There are no technical corrigenda specific to this Recommendation.

A.3 Versions of the ODA Specification

The version of the ODA International Standard defined in A.1 is known as version 1.1. The associated date is 1992-01-01. The version of the ODA International Standard including all features of the amendments, addenda and technical corrigenda mentioned in A.1 is known as version 1.2. The associated date is 1992-07-01.

The entries for ODA version for this document application profile is:

- standard or recommendation: “CCITT Rec. T.410 Series (1988) | ISO 8613 (1989); version 1.1”;
- publication date: “1992-01-01”.

Annex B

Recommended practices

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

B.1 Transfer methods for ODA

B.1.1 Conveyance of ODA over CCITT Recommendation X.400-1984

This describes how ODA body parts are to be encoded for transmission over a CCITT X.400-1984 service.

An ODA body part is encoded as OdaBodyPart in the definition given below:

```
OdaBodyPart ::= SEQUENCE { OdaBodyPartParameters, OdaData }
OdaBodyPartParameters ::= SET {
    document-application-profile
        [0] IMPLICIT OBJECT IDENTIFIER,
    document-architecture-class
        [1] IMPLICIT INTEGER {
            formatted (0),
            processable (1),
            formatted-processable (2) }
OdaData ::= SEQUENCE OF Interchange-Data-Element
```

NOTE – It is recommended to transfer an ODA document as a single body part with tag 12:

Oda [12] IMPLICIT OCTETSTRING

The content of the octet string is encoded as OdaBodyPart, defined above. However, this is out of the scope of this Profile.

B.1.2 Conveyance of ODA over FTAM

This describes the FTAM Document Type to be used for minimal storage and transfer capabilities of ODA data streams. It is recognized that enhanced capabilities may at some point be added.

When using FTAM to transfer an ODA file, the FTAM-3, "ISO FTAM Unstructured Binary", document type shall be specified.

However, since files that do not contain ODA data streams can have the same document type, it is left up to the user of application programs that remotely access files using FTAM to know that a given file contains an ODA data stream.

B.1.3 Conveyance of ODA over DTAM

This provides for information concerning the interchange of ODA based documents with DTAM protocols.

DTAM (Document Transfer and Manipulation) is defined in the T.430-Series of CCITT Recommendations and is, like ODA, an integral part of the T.400-Series of CCITT Recommendations named Open Document Architecture, Transfer and Manipulation.

The T.520-Series of CCITT Recommendations contain Communication Application Profiles (CAP). CCITT Recommendation T.522 describes the Communication Application Profile BT1 for document bulk transfer. CCITT Recommendation T.522 is applicable for the Profile published in this Recommendation.

NOTE – The use of BT1 within the end-to-end oriented Telematic Services Telefax 4 and Teletex is described in 7.1/T.561 and in 7.1/T.562.

B.1.4 Conveyance of ODA over flexible disks

The recommended method for interchanging ODA documents between systems by the exchange of magnetically recorded Flexible Disk Cartridges by the use of an annex to CCITT Rec. T.411 | ISO/IEC 8613-1 (to be published), "*Recording of Documents Conforming to ISO 8613 on Flexible Disk Cartridges Conforming to ISO 9293"*. This annex provides for recording each ODA document as a separate file as defined by ISO 9293.

B.2 Font reference

The recommended method for specifying a font reference is to be based on ISO/IEC 9541.

Font sizes from 6 to 72 points (100 to 1200 BMU) are intended to be supported by implementation conforming to this Recommendation. All other values of font sizes may additionally be supported, but implementations may also support using some form of fallback.

The minimum font properties and values from ISO/IEC 9541 that are to be specified in a Font-Attribute-Set are those specified by the following document application profile notation.

```
Font-Attribute-Set {
PERM        Fontname                        {ANY_VALUE},
PERM        Standard-Version                {ANY_VALUE},
PERM        Data-Source                     {ANY_VALUE},
PERM        Design-Source                   {ANY_VALUE},
PERM        Font-Family-Name                {ANY_VALUE},
PERM        Posture                         {'upright' | 'italic-forward'},
PERM        Weight                          {'light' | 'medium' | 'bold'},
PERM        Proportionate-Width             {ANY_VALUE},
PERM        Glyph-Complement  {
   PERM #Included-Glyph-Collections         {ANY_VALUE},
   PERM #Excluded-Glyph-Collections         {ANY_VALUE},
   PERM #Included-Glyphs                    {ANY_VALUE},
   PERM #Excluded-Glyphs                    {ANY_VALUE}
                        },
PERM        Design-Size                     {ANY_VALUE},
PERM        Min-Size    {
   PERM #Numerator                          {100 .. 1200},
   PERM #Denominator                        {1}
                        },
PERM        Max-Size    {
   PERM #Numerator                          {100 .. 1200},
   PERM #Denominator                        {1}
                        },
   -- BMU Units equivalent to range of 6..72 point sizes --
PERM        Design-Group  {
   PERM #Class                              {ANY_VALUE},
   PERM #Subclass                           {ANY_VALUE},
   PERM #Specific-Group                     {ANY_VALUE}
                        },
PERM        Structure                       {ANY_VALUE},
PERM        Writing-Modes {
            MUL             {
   REQ  #Writing-Mode-Name                  {ANY_VALUE},
   PERM #Nominal-Escapement-Direction       {ANY_VALUE},
   PERM #Escapement-Class                   {ANY_VALUE},
   PERM #Average-Escapement-X               {ANY_VALUE},
   PERM #Average-Escapement-Y               {ANY_VALUE}
                }
            }
```

B.3 ISO/IEC 8632 (CGM) constraints for this document application profile

It is recommended that geometric graphics content information contain only those elements listed in this portion of the document, in addition to the constraints imposed by CCITT Rec. T.415 | ISO/IEC 8613-5. It is believed that this subset of the CGM is sufficiently implemented to enable interworking of geometric graphics for application conforming to this document application profile.

Where an element has parameters, recommended constraints on the values are given. The -- symbol indicates that there is no recommended constraint.

Requirements in ISO/IEC 8632 and CCITT Rec. T.415 | ISO/IEC 8613-5 concerning mandatory elements, parameters shall be fulfilled.

B.3.1 Delimeter elements

No-op	An arbitrary sequence of n octets. Where n = 0, 1, ..., 32 767. The sequence of zero or more octets is for padding purposes.
Begin Metafile	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets.

End Metafile	
Begin Picture	Support will be provided for strings up with a length to 254 octets, except for data records which will support strings with a length up to 32 767 octets.
Begin Picture Body	
End Picture	

B.3.2 Metafile descriptor elements

Metafile Version	1
Metafile Description	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets. The METAFILE DESCRIPTION string parameter will be used to include the sub-string "ISO FOD26" to the label content information as conforming to this Profile. In addition, generators of content are encouraged to append a sub-string that identifies the company and product that produced the CGM.
VDC Type	Integer
Integer Precision	16
Real Precision	(0, 9, 23), (1, 16, 16)
Index Precision	16
Colour Precision	8, 16
Colour Index Precision	8, 16
Maximum Colour Index	0 .. 63
Colour Value Extent	--
Metafile Element List	--
Metafile Defaults Replacement	--
Font List	All fonts referenced in the metafile shall be defined. Font referencing in FONT LISTS using ISO/IEC 9541 names is preferred, but font names may be specified using proprietary font names.
Character Set List	All character sets referenced in the metafile shall be defined in CHARACTER SET LIST. Permissible character sets are the same as for character content architecture.
Character Coding Announcer	--

B.3.3 Picture descriptor elements

Scaling Mode	The Scale Factor parameter of SCALING MODE element is always a 32-bit floating point value, even when the REAL PRECISION has selected fixed point for other real numbers. It is not apparent in ISO/IEC 8632 what the precision of this floating point value is when fixed point has been selected. Its precision shall be (0, 9, 23).
Colour Selection Mode	Indexed
Line Width Specification Mode	Scaled
Marker Size Specification Mode	Scaled
Edge Width Specification Mode	Scaled
VDC Extent	--
Background Colour	--

B.3.4 Control elements

VDC Integer Precision	16
VDC Real Precision	(0, 9, 23), (1, 16, 16)
Auxiliary Colour	--
Transparency	Transparent
Clip Rectangle	--
Clip Indicator	--

B.3.5 Graphical primitive elements

Polyline	Support for point lists with up to 255 vertices.
Disjoint Polyline	Support for point lists with up to 255 vertices.
Polymarker	Support for point lists with up to 255 vertices.
Text	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets. Format effector control characters are disallowed in the string parameter.
Restricted Text	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets. Format effector control characters are disallowed in the string parameter.
Append Text	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets. Format effector control characters are disallowed in the string parameter.
Polygon	Support for point lists with up to 255 vertices.
Polygon Set	Support for point lists with up to 255 vertices.
Cell Array	--
Rectangle	--
Circle	--
Circular Arc 3 Point	--
Circular Arc 3 Point Close	--
Circular Arc Centre	--
Circular Arc Centre Close	--
Ellipse	--
Elliptical Arc	--
Elliptical Arc Close	--

B.3.6 Attribute elements

Line Bundle Index	1
Line Type	1-5
Line Width	--
Line Colour	--
Marker Bundle Index	1
Marker Type	1-5
Marker Size	--
Marker Colour	--

Text Bundle Index	1
Text Font Index	All fonts referenced (indexed by TEXT FONT INDEX) in the metafile shall be defined in FONT LIST either in presentation parameters of CCITT Rec. T.410 Series | ISO/IEC 8613 or in ISO/IEC 8632.
Text Precision	0 (string)
Character Expansion Factor	1.0
Character Spacing	0.0
Text Colour	--
Character Height	--
Character Orientation	--
Text Alignment	--
Character Set Index	All character sets referenced in the metafile (indexed by CHARACTER SET INDEX) shall be defined in CHARACTER SET LIST. The only character sets which may be designated in G0 are ISO/IEC 646:1991 IRV or versions of ISO/IEC 646:1991. Other character sets shall be designated in G1, G2 or G3.
Alternate Character Set Index	All character sets referenced in the metafile (indexed by ALTERNATE CHARACTER SET INDEX) shall be defined in CHARACTER SET LIST.
Fill Bundle Index	1
Interior Style	--
Fill Colour	--
Hatch Index	Negative values are prohibited.
Pattern Index	1
Edge Bundle Index	1
Edge Type	1
Edge Width	1.0
Edge Colour	--
Edge Visibility	0 (off)
Fill Reference Point	--
Pattern Table	The PATTERN TABLE element has an unspecified effect when it appears in a picture subsequent to any graphical primitives. The PATTERN TABLE element shall appear prior to any graphical primitive element to assure that interpreting systems without dynamic pattern update can render the intended effect. The minimum support for the length of the Colour Array parameter for the PATTERN TABLE element is 2048. This will support 8 patterns of 16×16, 2 patterns of 32×32 or 1 pattern of 32×64. All indexes which are used in the metafile shall be defined.
Pattern Size	--
Colour Table Specification	The COLOUR TABLE element has an unspecified effect when it appears in a picture subsequent to any graphical primitives. The COLOUR TABLE element shall appear prior to any graphical primitive elements to assure that interpreting systems without dynamic colour update can render the intended effect. The minimum support for the length of the Colour List parameter in the COLOUR TABLE element is 63. This will support a 64 (0..63) entry colour table. All indexes which are used in the metafile shall be defined.
Aspect Source Flags	Individuals

B.3.7 External elements

Message	The presentation of a message string may not be appropriate for all applications. No requirement for the formatted presentation of the message string has been placed on the Interpreter. Only the No Action Flag needs to be supported. Support for string lengths up to 254.
Application Data	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets.

Annex C

Bibliography

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

– ITU-T Recommendation T.52 (1993), *Non-latin coded character sets for telematic services.*

– CCITT Recommendation T.502 (1991), *Document application profile PM-11 for the interchange of character content documents in processable and formatted forms.*

– CCITT Recommendation T.505 (1991), *Document application profile PM-26 for the interchange of mixed content documents in processable and formatted forms.*

– ISO 8571:1988, *Information processing systems – Open systems interconnection – File transfer access and management.*

– ISO/IEC 9070:1991, *Information technology – SGML support facilities – Registration procedures for public text owner identifiers.*

– ISO/IEC TR 9573:1988, *Information processing – SGML support facilities – Techniques for using SGML.*

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