INTERNATIONAL TELECOMMUNICATION UNION

CCITT T.505

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

TERMINAL EQUIPMENT AND PROTOCOLS FOR TELEMATIC SERVICES

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

Recommendation T.505

Geneva, 1991

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT 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 Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation T.505 was prepared by Study Group VIII and was approved under the Resolution No. 2 procedure on the 18 of January 1991.

CCITT NOTE

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

© ITU 1991

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

Recommendation T.505

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

CONTENTS

0 *Introduction*

1 *Scope and field of application*

2 *References*

3 *Definitions and abbreviations*

- 3.1 Definitions
- 3.2 Constituent names

4 *Relationships with other profiles*

5 *Conformance*

- 5.1 Data stream conformance
- 5.2 Implementation conformance

6 *Characteristics supported by this document application profile*

- 6.1 Overview
- 6.2 Logical characteristics
- 6.3 Layout characteristics
- 6.4 Document layout characteristics
- 6.5 Content layout and imaging characteristics
  - 6.5.1 Character content
  - 6.5.2 Raster graphics content
  - 6.5.3 Geometric graphics content
- 6.6 Miscellaneous features
- 6.7 Document management features

7 *Specification of constituent constraints*

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

8 *Interchange format*

Annex A – Font reference

Annex B – ISO 8632 (CGM) constraints

0 Introduction

This is the definition of an ODA document application profile (DAP) named PM-26. This profile is suitable for interchanging documents in formatted form, processable form or formatted processable form and has been defined in accordance with [CCITT Recommendation T.411/ISO 8613-1]. The format of this profile is in accordance with the standardized proforma and notation defined in draft addendum to [CCITT Recommendation T.411/ISO 8613-1] Annex F (to be published).

1 Scope and field of application

This profile specifies interchange formats for the transfer of structured documents between equipment 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.

2 References

ISO 8613-1 – Information processing – Text and office systems; Office document architecture (ODA) and interchange format – Part 1: Introduction and general principles (1989).

ISO 8613-2 – Information processing – Text and office systems; Office document architecture (ODA) and interchange format – Part 2: Document structures (1989).

ISO 8613-4 – Information processing – Text and office systems; Office document architecture (ODA) and interchange format – Part 4: Document profile (1989).

ISO 8613-5 – Information processing – Text and office systems; Office document architecture (ODA) and interchange format – Part 5: Open document interchange format (1989).

ISO 8613-6 – Information processing – Text and office systems; Office document architecture (ODA) and interchange format – Part 6: Character content architectures (1989).

ISO 8613-7 – Information processing – Text and office systems; Office document architecture (ODA) and interchange format – Part 7: Raster graphics content architectures (1989).

ISO 8613-8 – Information processing – Text and office systems; Office document architecture (ODA) and interchange format – Part 8: Geometric graphics content architectures (1989).

ISO 8613-10 – Information processing – Text and office systems; Office document architecture (ODA) and interchange format – Part 1: DAD – A document application profile proforma and notation (to be published).

Recommendation T.411 – Open document architecture (ODA) and interchange format: Introduction and general principles.

Recommendation T.412 – Open document architecture (ODA) and interchange format: Document structures.

Recommendation T.414 – Open document architecture (ODA) and interchange format: Document profile.

Recommendation T.415 – Open document architecture (ODA) and interchange format: Open document interchange format (ODIF).

Recommendation T.416 – Open document architecture (ODA) and interchange format: Character content architectures.

Recommendation T.417 – Open document architecture (ODA) and interchange format: Raster graphics content architectures.

Recommendation T.418 – Open document architecture (ODA) and interchange format: Geometric graphics content architectures.

Recommendation T.4 – Standardization of Group 3 facsimile apparatus for document transmission.

Recommendation T.6 – Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus.

ISO 8859-1 – Information processing – 8-bit single-byte coded graphic character sets – Part 1: Latin Alphabet No. 1 (1987).

ISO 646 – Information processing – ISO 7-bit coded character sets for information interchange (revised 1990).

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

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

ISO 7350 – Text communication – Registration of graphic character subrepertoires (1984).

Recommendation X.209 – Specification of basic encoding rules for abstract syntax notation one (ASN.1).

ISO 8825 – Information processing systems – Open systems interconnection – Basic encoding rules for abstract syntax notation one (ASN.1) (1987).

Recommendation T.502 – Document application profile PM-11 for the interchange of character content documents in processable and formatted forms (to be published).

Recommendation T.506 – Document application profile PM-36 for the interchange of enhanced open content documents in processable and formatted forms (to be published).

Recommendation T.515 – Implementation requirements for document application profile PM-26 (to be published).

ISP FOD11 – Office document format profile (to be published).

ISP FOD26 – Office document format profile (to be published).

ISP FOD36 – Office document format profile (to be published).

3 Definitions and abbreviations

3.1 *Definitions*

The definitions given in [CCITT Recommendation T.411/ISO 8613-1] are applicable to this profile.

3.2 *Constituent names*

Each constituent that may be included in a document that conforms to this profile has been given a unique name which serves to identify that constituent throughout this profile.

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 names used in this profile are BodyText, Footnote, RectoPage and ColumnFixed.

In § 6 of this profile, each constituent provided by this profile is underlined once at the point in the text at which the purpose of that constituent is defined. This also serves to identify all the constituents provided by this profile.

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

Although the constituent names relate to the purpose of the constituents, the semantics of constituents must not be implied from the actual names that are used. Also, these names do not appear in an interchanged document but a mechanism for identifying constituents in an interchange document 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 (see Recommendation T.502) and PM-36 (see Recommendation T.506).

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.

Also the features supported by this profile are a subset of those supported by the profile PM-36 and thus all data streams that are conformant to this profile are also conformant to PM-36.

The profile defined in this Recommendation is identical to the profile defined in FOD26. Similarly, the profiles PM-11 and PM-36 are identical to the profiles defined in FOD11 and FOD36 respectively.

5 Conformance

In order to conform to this profile, a data stream representing a document must meet the requirements specified in § 5.1.

This Recommendation does not define implementation or service requirements. These requirements are defined in other Recommendations that make use of this profile.

5.1 *Data stream conformance*

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

a) The data stream shall be encoded in accordance with the ASN.1 encoding rules defined in [CCITT Recommendation X.209/ISO 8825].

b) The data stream shall be structured in accordance with the interchange format defined in § 8 of this profile.

c) 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 of this profile 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 § 7.

d) 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.

e) The attribute values specified shall be within the range of permissible values specified in this profile.

f) The encoded document shall be constructed in accordance with the abstract document architecture defined in [CCITT Recommendation T.412/ISO 8613-2].

g) The document shall be structured in accordance with the characteristics and constraints specified in § 6 of this profile.

5.2 *Implementation conformance*

The implementation requirements associated with this profile are defined in Recommendation T.515.

6 Characteristics supported by this document application profile

This section describes the characteristics of documents which can 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;

– generic-documents;

– document profile.

The constituents that may make up these forms of document are defined in §§ 6.1.2 to 6.1.6. Constituents defined as “required” must occur in any document that conforms to this profile. Constituents listed as “optional” may or may not be present in the document depending on the requirements of the particular document.

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

6.1.2 *Formatted form documents*

a) *Required constituents:*

- a document profile,
- layout object descriptions representing a specific layout structure.

b) *Optional constituents:*

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

6.1.3 *Processable form documents*

a) *Required constituents:*

- a document profile;
- logical object class descriptions representing a "complete" generic logical structure;
- logical object descriptions representing a specific logical structure.

b) *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 of this profile.

6.1.4 *Formatted processable documents*

a) *Required constituents:*

- a document profile;
- logical object class descriptions representing a "complete" 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.

b) *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" generic logical structure;
- layout styles whose presence are optional;
- presentation styles whose presence are optional;
- generic content portions whose presence are optional;

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

c)
- a document profile;
- logical object class descriptions which represent a "complete" generic logical structure;
- layout object class descriptions which represent a "complete" generic layout structure;
- layout styles whose presence are optional;
- presentation styles whose presence are optional;
- generic content portions whose presence are optional.

6.1.6 *Document profile*

This type of document contains a document profile only.

6.2 *Logical characteristics*

6.2.1 *Introduction*

This clause defines the logical constituent constraints 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 logical characteristics. This clause 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 may 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 main content of a document and is intended to be reproduced in the body area of the pages that make up the document. The "body" part must be included in all documents that are interchanged in accordance with this profile.

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 initialized 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. For example:

– 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 initialized 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 must at least have 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 of a Passage or another higher level NumberedSegment. It is used to represent the grouping of parts of a document that are distinguished by an identifier. These parts 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 initialized 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 constraint types:

– 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 constituent NumberedSegment.

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 cannot be recursively defined. That is, a NumberedSegment at one level cannot refer to a NumberedSegment at a higher level and there must be one NumberedSegment definition for each level of NumberedSegment in the specific logical structure of a document.

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 is 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 sequences 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 may be distinguished by means of “separators” characters such as spaces and full stops; a typical example is “6.2.3.4”.

Further details of the structure and generation of the identifiers are given in § 6.6.7.

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.

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. Alternatively, the character content may contain hard new line controls, which will cause parts of the content to be separated when laid out.

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 may refer to one or more content portions each containing processable, formatted or formatted processable character content. BodyRaster and BodyGeometric may 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 is 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 is a constituent constraint that is used to represent a footnote reference within the body of a document.

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

This 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.

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 "note5" 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.

The format of the content generator referred to above is described in § 6.6.8.

6.2.3.9 *FootnoteBody*

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

FootnoteBody is a composite logical constituent whose subordinates consist of the constituent constraint FootnoteNumber, which is mandatory and represents the footnote identifier, followed by one or more constituent constraints of the type FootnoteText 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 is 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 layout category name "Footnote" for this constituent; this ensures that this constituent is laid out in a FootnoteArea frame when a 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 is 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 layout category name “Footnote” for this constituent; this ensures that this constituent is laid out in a FootnoteArea frame when a 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 may consist of any combination of character, raster graphics and geometric 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 can be represented by this constituent.

CommonText is a basic logical object class that references one or more content portions each containing processable, formatted and formatted processable character content.

6.2.4.3 *PageNumber*

PageNumber is a constituent constraint that represents common character content that is to be laid out in the header or footer area of a document. This constituent 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 can start at 0 or any value greater than 0.

The format of the content generators is defined in § 6.6.6.

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 basic logical object class which references a single content portion containing formatted processable geometric graphics content.

6.3 *Layout characteristics*

This clause defines the layout constituent constraints 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 clause 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 may 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 subdivided into three areas; the body area, which is used to lay out the document body, and the header and footer areas, which may be used to lay out the common content.

Four page layout types are supported 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/T.505, 2/T.505, 3/T.505 and 4/T.505 respectively.

It is intended that all applications which use this profile should 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 lay out 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 vertically 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 subdivided 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 subdivided 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 can be initialized 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 page types concerns the values that can be specified for the parameter “side of sheet” in the attribute “medium type”. In the case of Page, the value of this parameter may be specified as “recto”, “verso” or “unspecified”. In the case of RectoPage, the value of this parameter may be specified as “recto” or “unspecified”; in the case of VersoPage, the value of this parameter may be specified as “verso” or “unspecified”.

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) a sequence of pages represented by the constituent constraint Page. All pages in this sequence must have the same layout characteristics (see Note) but these characteristics may differ from those of the initial page;

b) a sequence of pages which is intended to be laid out alternatively on the "recto" and "verso" (or on the "verso" and "recto") sides of the presentation medium and is represented by the constituent constraints RectoPage and VersoPage respectively. All pages in this sequence must have the same layout characteristics (see Note) but these characteristics may differ from those of the initial page.

A page set must contain at least one page.

An initial page is typically used at the beginning of a document or at 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:

i) all the pages must have the same dimensions and orientation (see § 6.3.4.2);

ii) all pages are to be laid out on the same size of presentation medium (see § 6.3.4.3).

Note – The layout characteristics of pages are specified in § 6.3.4.5. Pages having the same layout characteristics are pages for which the body area, header area (if present) and footer area (if present) have the same dimensions and position within the page (see § 6.3.4.3). However, pages having the same layout characteristics do not necessarily have the same position on the presentation medium (see § 6.3.4.4).

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.

Dimensions equivalent to or less than the common assured reproduction area of ISO A4 and North American Letter (NAL) in portrait or landscape orientation are basic values. Larger page sizes are non-basic and their use must be indicated in the document profile.

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

6.3.4.3 *Nominal page sizes*

The nominal page sizes that may be specified are listed in Table 1/T.505. 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 must be indicated in the document profile.

Any of the nominal page sizes defined in Table 1/T.505, subject to the restrictions specified above, may be specified as the default value in the document profile.

Table 1/T.505 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	7015 × 9920	not defined
ISO A4	210 mm × 297 mm	9920 × 14030	9240 × 13200
ISO A3	297 mm × 420 mm	14030 × 19840	13200 × 18480
ANSI legal	8.5" × 14"	10200 × 16800	9240 × 18480
ANSI A	8.5" × 11"	10200 × 13200	9240 × 12400
ANSI B	11" × 17"	13200 × 20400	12744 × 19656
Japan-legal	257 mm × 364 mm	12141 × 17196	11200 × 15300
Japan-letter	182 mm × 257 mm	8598 × 12141	7600 × 10200

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 must occur on every page in a document. The header and footer areas are both optional.

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

Four types of page layout are supported as defined below.

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/T.505.

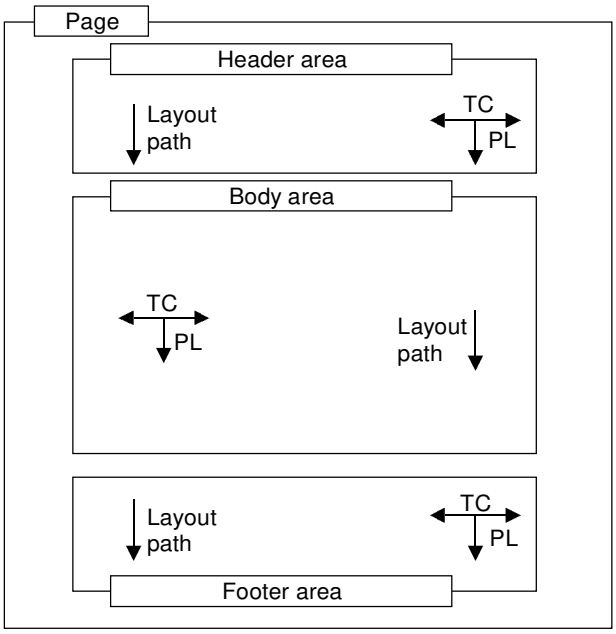

FIGURE 1/T.505

Page layout type A

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/T.505.

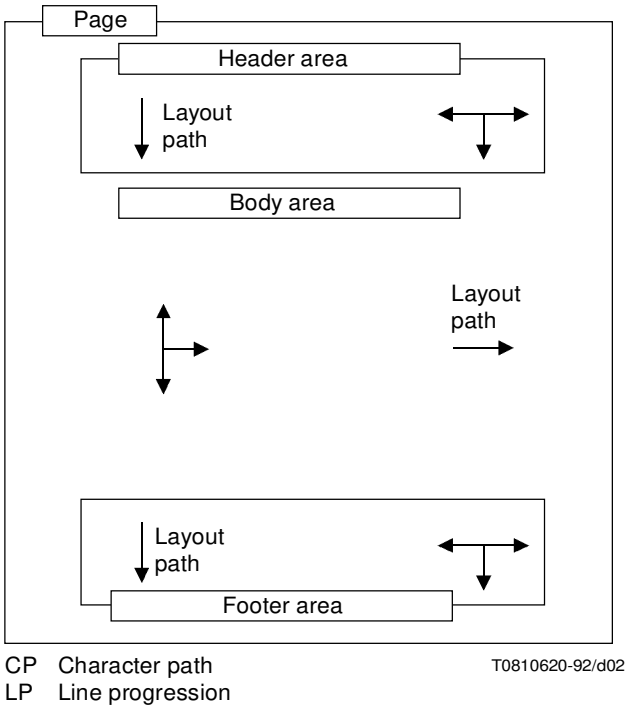

FIGURE 2/T.505

Page layout type B

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/T.505.

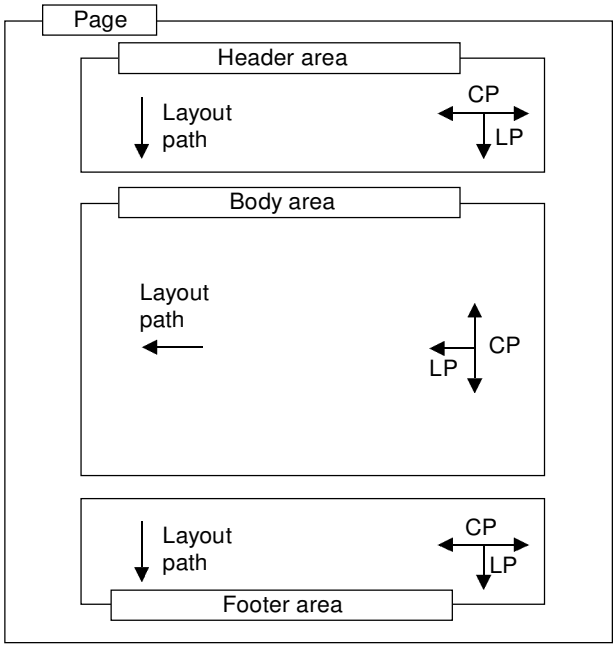

FIGURE 3/T.505

Page layout type C

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/T.505.

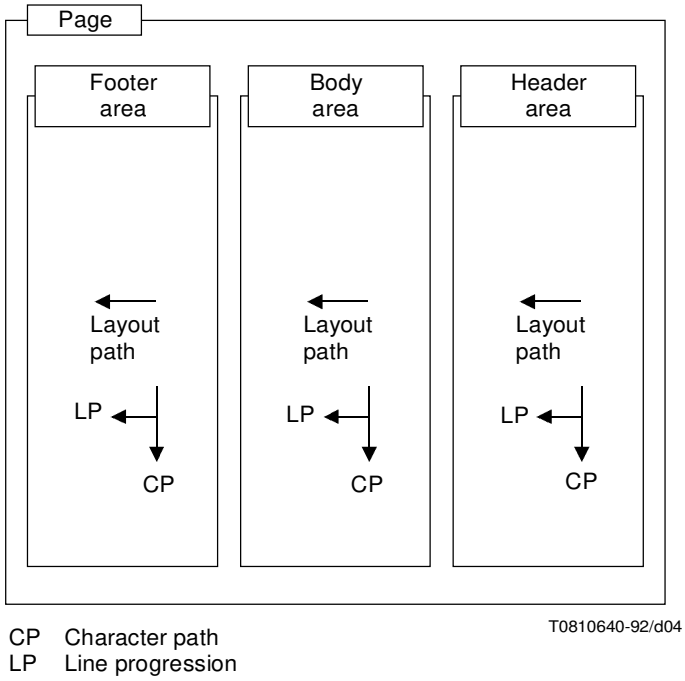

FIGURE 4/T.505

Page layout type D

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 regions to provide for combinations of single or multiple column layout and the layout of footnotes. 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. 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 frames of this type consist of an arbitrary ordered sequence of one or more frames of the following types:

– BasicFloat;

– SnakingColumns;

– SynchronizedColumns.

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

The subordinate frames are all variably positioned and have variable dimensions. Frames of the type BasicFloat, SnakingColumns and SynchronizedColumns are laid out in the direction of the layout path of the body area (i.e their positioning fill order is “normal”), whereas FootnoteArea frames are laid out in the direction opposite to that of the body area layout path (i.e in “reverse” positioning fill order).

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 body area.

Figures 5/T.505, 6/T.505 and 7/T.505 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 can 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.

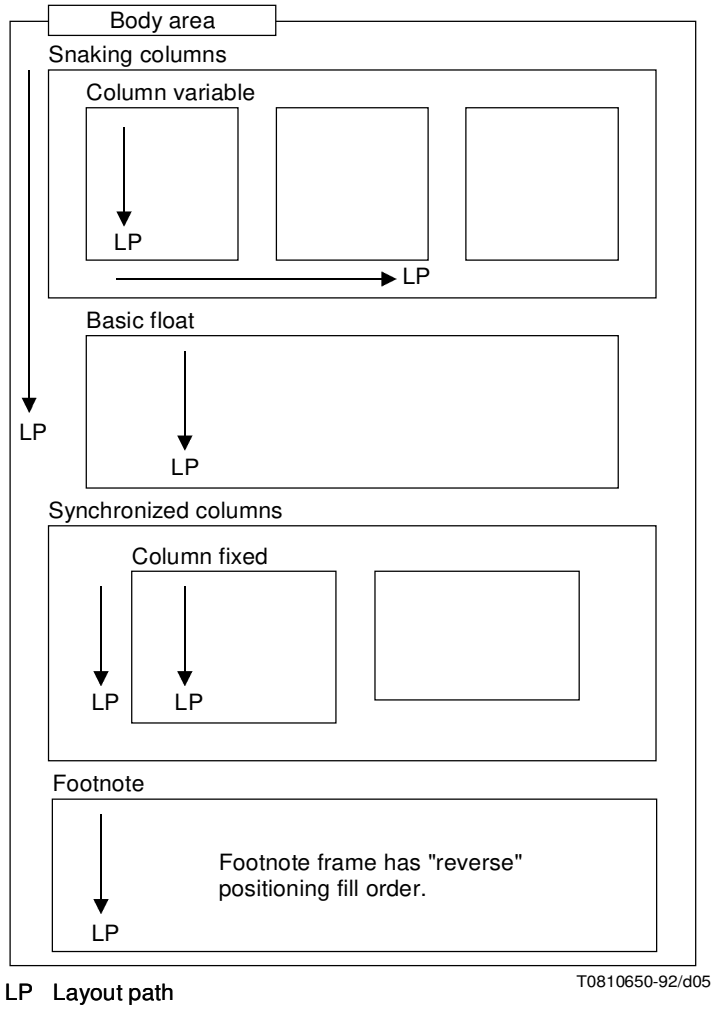

FIGURE 5/T.505

Example of body area layout for page layout A

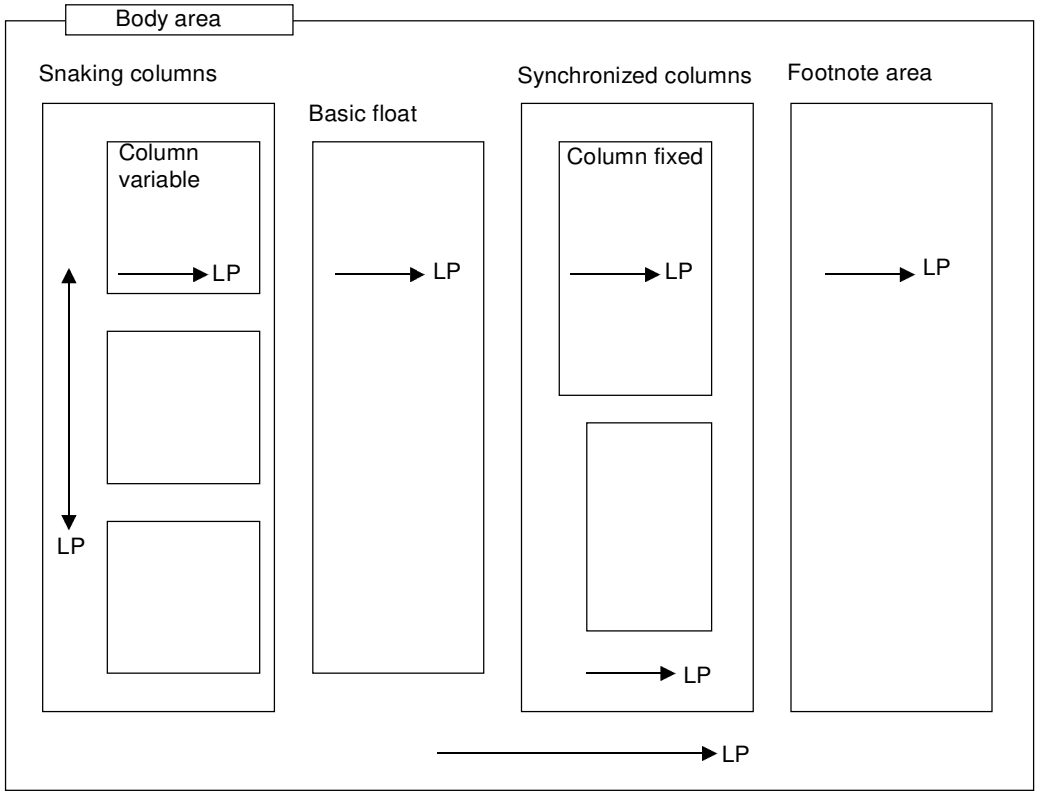

FIGURE 6/T.505

Example of body area layout for page layout B

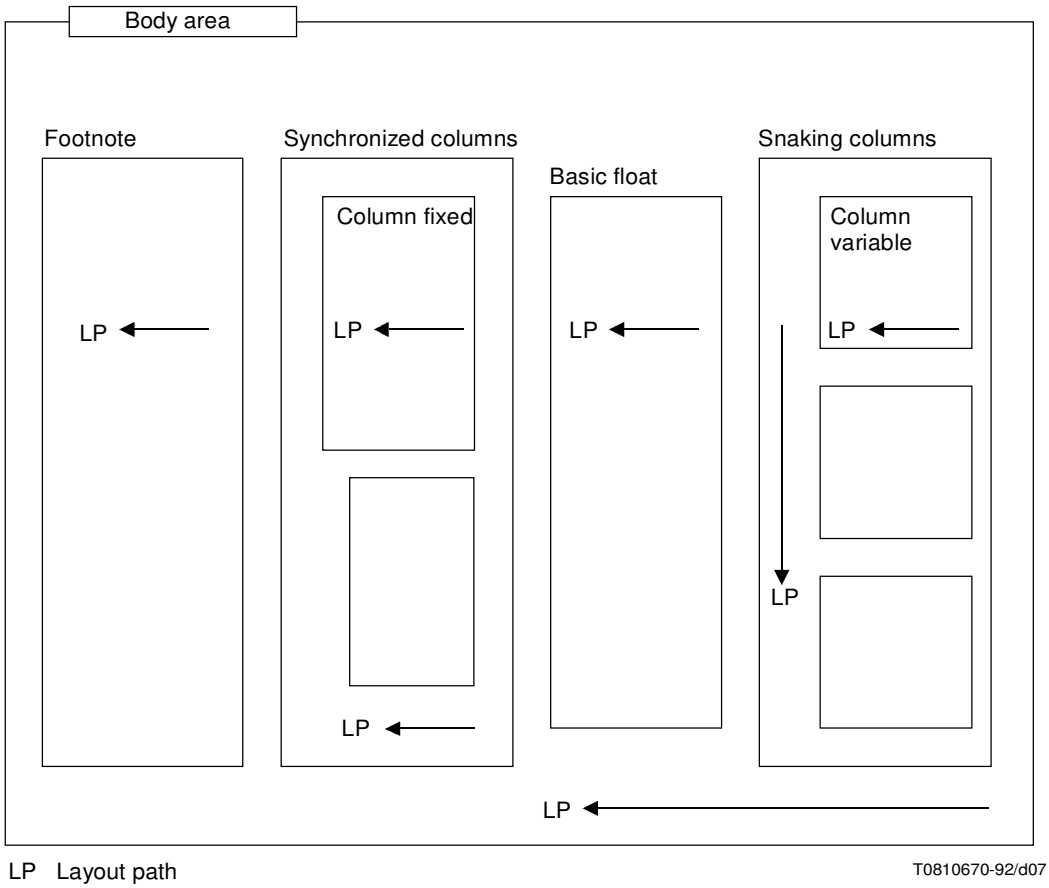

FIGURE 7/T.505

Example of body area layout for page layouts C and D

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 lay out content in the form of a single column. This is a variably positioned frame.

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

The dimensions of the edge in the direction parallel to the layout path of the body area is specified as "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 may 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 synchronized column area within a body area. A synchronized 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 is a frame which 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/T.505.

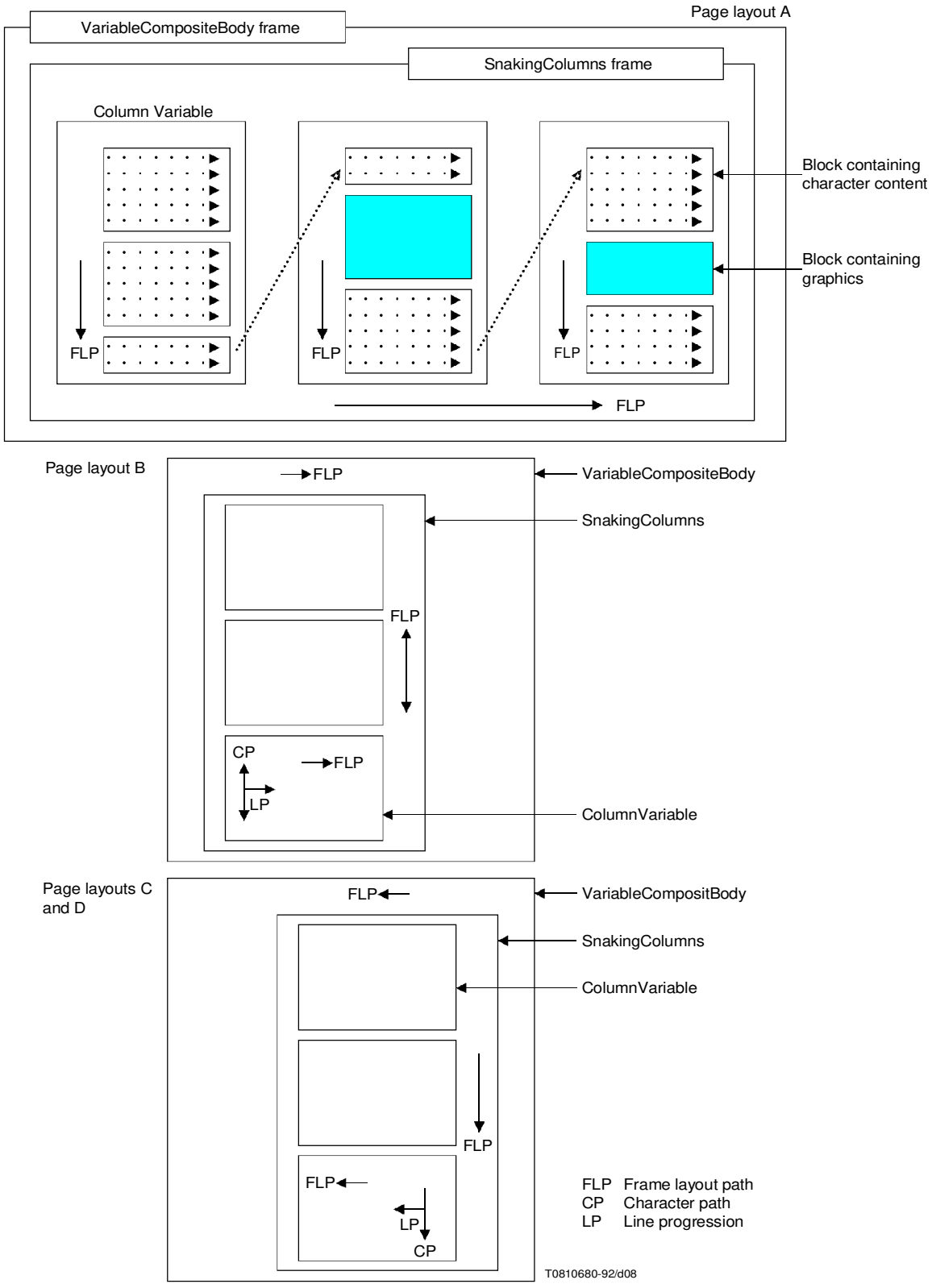

FIGURE 8/T.505

Examples of the layout of snaking columns

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

The dimensions of the edge of this frame in the direction parallel to the layout path of the body area is specified as "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 equal in length in the vertical dimension in the case of page layout A and equal in length in the horizontal dimension in the cases of page layouts B, C and D (see Note).

Note – It is intended that the attribute "balance" may be ignored when the subordinate ColumnVariable frames have unequal widths.

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/T.505.

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.

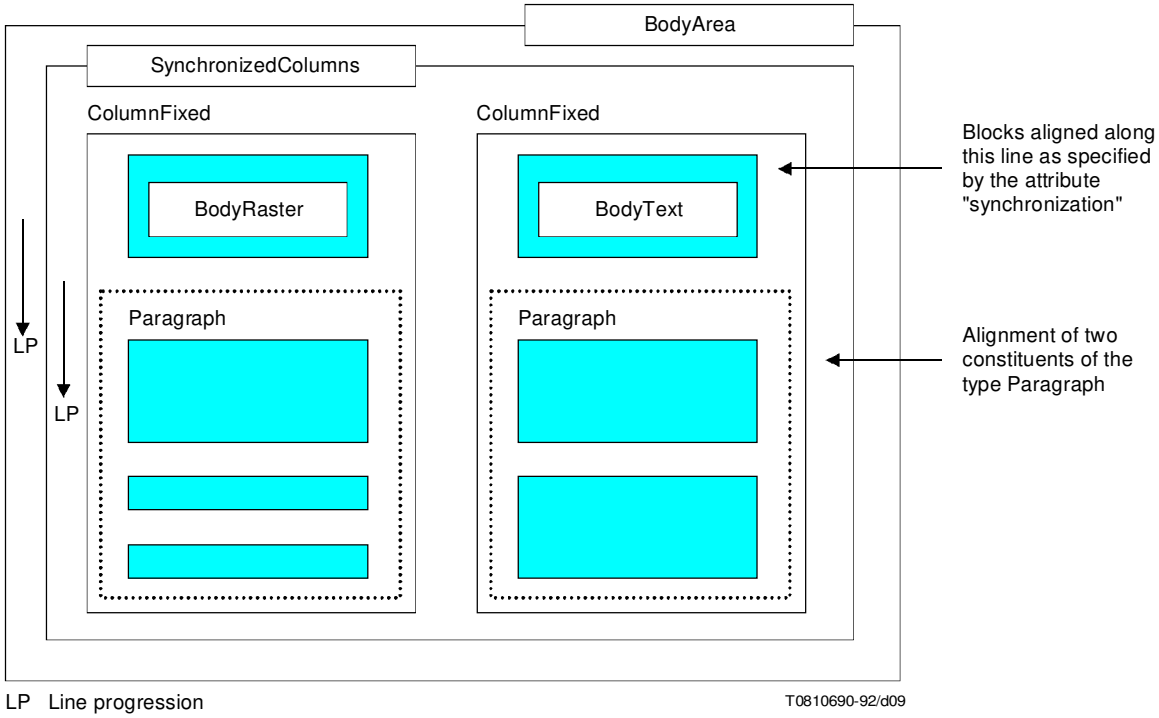

FIGURE 9/T.505

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 as “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 must 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 fixed or specified as "maximum-size" (see Note) 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 must 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 as "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 may only be specified as "maximum-size".

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

The layout path for ColumnFixed frames must 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 as "synchronized" by using the attribute "Synchronization".

Note – The value "maximum-size" may only be specified for the last ColumnFixed frames 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 fixed or specified as "maximum-size". In the direction of the layout path, the dimension is specified by "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 only content that may be laid out in this frame is content which is associated with basic logical objects that are subordinate to the composite logical object "FootnoteBody". To achieve this, the category name "Footnote" is specified for this frame.

6.3.6 *Header and footer area characteristics*

6.3.6.1 *General characteristics*

The header and footer areas may consist of 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 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 layout paths 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.

The subordinate frames within a CompositeHeader or CompositeFooter frame 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 the edge of this frame which is orthogonal to the direction of the layout path of the superior frame is fixed or specified as “maximum-size”. The dimensions of the edge in the direction parallel to the layout path of the superior frame is specified as either fixed 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 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 predefined common content contained in the generic layout structure. The positioning, dimensioning and layout path characteristics of this frame are the same as that for SourcedContentVariable frames.

This frame references 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 predetermined 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 dimensions and its layout path is equal to that of the containing header or footer area.

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 predefined common content derived from the generic layout structure. This frame references 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 lay out common content at predetermined positions in the header or footer areas.

The positioning, dimensioning and layout path characteristics of this frame are the same as that for SourcedContentFixed frames.

6.3.7 *GenericBlock and SpecificBlock*

Two types of constituent constraints of the type "block" are defined, namely GenericBlock and SpecificBlock.

Objects of the type GenericBlock may occur in the generic layout structure as subordinates to 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 therefore restricted to occur within the header and footer areas of the page.

Objects of the type SpecificBlock may 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 relating to documents for a generic layout structure are 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.

a) *Layout in a nominated page set*

This 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 and NumberedSegment using the attribute "Layout object class" which specifies the object identifier of the required page set.

b) *Starting a new page set*

This provides the ability to specify that a particular logical 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 constituent constraints of the types Passage, NumberedSegment, Number, FootnoteReference, BodyText, BodyRaster and BodyGeometric using the attribute "New layout object" which specifies the object identifier of the required page set.

6.4.1.2 *Page breaks*

This provides the ability to specify that a particular logical 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 must belong to the page set in which the immediate preceding logical constituent constraint is laid out (see Note).

This may be specified for logical constituent constraints of the types Passage, NumberedSegment, Number, FootnoteReference, BodyText, BodyRaster and BodyGeometric.

This is achieved using the attribute "New layout object". This attribute may specify the value "page" indicating that the logical 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 logical constituent constraint is to be laid out starting on a page of a particular class; this is achieved by specifying the object identifier of the required page class.

Note – The specification of a page break must not be used to lay out part of a document in a new page set. If a new page set is required, then this should be explicitly specified as described in § 6.4.1.1.

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 content into a composite body area, it is necessary to indicate the type of column area that is to be used.

Logical constituent constraints of the types NumberedSegment, Number, FootnoteReference, BodyText, BodyRaster and BodyGeometric can be specified to be laid out starting at the beginning of single columns area, snaking columns area or synchronized columns area. When a particular type of area has been specified the document content continues to be laid out in this area until a different area is selected. This can occur at any point in the document for the above logical constituent constraint types.

If there is insufficient area on one page to lay out all the content allocated to a particular type of area, then the layout of the content will automatically continue in the same type of area in the next 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 next page.

However, if a new page set or new page is explicitly specified as in § 6.4.1.1 or 6.4.1.2 then it is necessary to explicitly specify the type of column area required.

The selection of the required type of column area is achieved using the attribute “New layout object” which specifies the identifier of the frame class that represents the single, snaking or synchronized column area required. In the case of single column areas, the attribute “New layout object” may indicate the category name corresponding to the frame class of the single column area that is required.

When layout occurs in a snaking columns area, column breaks can be indicated by also using the attribute “New layout object”. This attribute may specify the identifier or the category name of the frame corresponding to the specific column in which the layout is to continue.

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 lay out the logical entities. Each column within a synchronized columns area must have a different category name and each basic logical entities to be laid out in this particular area must 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*

Basic logical 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 “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 footnote area on the next 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 predefined 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 entity 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 can 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/T.505).

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.

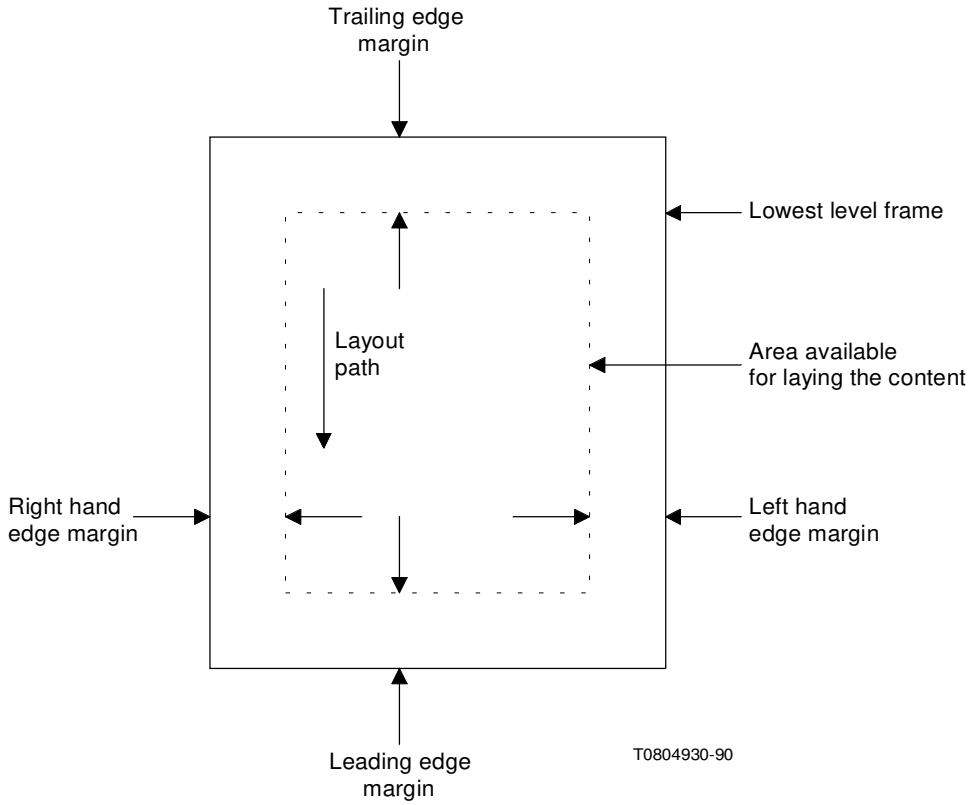

FIGURE 10/T.505

Specification of margins

6.4.2.2 *Separation*

Separation is the minimum distance between one basic logical constituent constraint and the next when they are laid out. It may be specified for any constituent constraint. This distance is 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 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, Number, FootnoteReference and BodyText. The attribute "Indivisibility" is used to specify this feature.

6.4.2.4 *Same layout object*

Same layout object provides the means to specify that the content associated with a basic logical constituent constraint and the content associated with a previous basic logical constituent constraint are to be regarded as an unbroken stream of content within a page. This may be specified for constituent constraints of the types NumberedSegment, Paragraph, Number, Footnote, FootnoteReference, BodyText, BodyRaster and BodyGeometric.

The attribute "Same layout object" is used to specify this feature. This attribute contains an expression which indicates that the previous logical constituent constraint and the constituent constraint to which the attribute applies are to be laid out starting on the same page.

6.4.2.5 *Concatenation*

Concatenation provides the means to specify that the content associated with a basic logical constituent constraint and the content associated with the previous basic logical 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, FootnoteNumber, 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 entity 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 BodyText and CommonText, when they contain formatted character content, BodyRaster, and 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 can be specified within the document. These restrictions are indicated below.

a) 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.

b) It is possible to specify page breaks as defined in § 6.4.1.2 but it is only possible to indicate that the layout should begin on a new page. It is not possible to specify a particular page class.

c) 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 can be distinguished by means of application comments (see § 6.6.4). An exception is that it is not possible to distinguish whether the common content is to be placed in a header or footer area (or split between the two).

d) It is not possible to indicate the type of layout area to be used to lay out 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 must be decided by the receiver.

e) Footnotes within the body part of a document can 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.

f) Margins, separation, indivisibility, same layout object, concatenation, and block alignment, as defined in § 6.4.2 can all be specified. Only one restriction applies. Indivisibility (see § 6.4.2.3) may be used to specify that a logical constituent constraint is not to be split over more than one page but indivisibility cannot 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 clause 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, 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 underlined, not crossed out;

– tabulation;

– indentation;

– alignment;

– first line format;

– 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 must 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 can 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 the IRV of ISO 646 (revised 1990) 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 "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 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 can be specified in the document profile using the presentation attribute “graphic character sets”.

If the character set defined in ISO 6937-2 is designated and invoked, then the use of any subrepertoire registered according to ISO 7350 may be specified using the presentation attribute “graphic character subrepertoire”. All subrepertoires are non-basic and their use must be indicated in the document profile. The subrepertoire shall not be changed within a content portion.

Note 1 – The basic character repertoire supported by this profile is not the standard default value specified in [CCITT Recommendation T.416/ISO 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.

Note 2 – Revised Recommendations T.50 and T.51 and new Recommendation T.52 are under preparation. T.50 and T.51 are intended to be completely compatible with ISO 646 (revised 1990) and ISO 6937 (under revision) respectively.

6.5.1.4 *Code extension techniques*

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

i) G0 set: only the IRV of ISO 646 (revised 1990), the primary set of ISO 6937-2 and a version of ISO 646 may be designated for this set; these character sets may only be invoked in GL.

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

iii) 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.)

iv) 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 must be designated into G0 (see item i) above) and invoked into GR. It is also necessary to invoke a character set into GR which has been designated into G1, G2 or G3 set.

v) The empty set must 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 Figure 11/T.505.

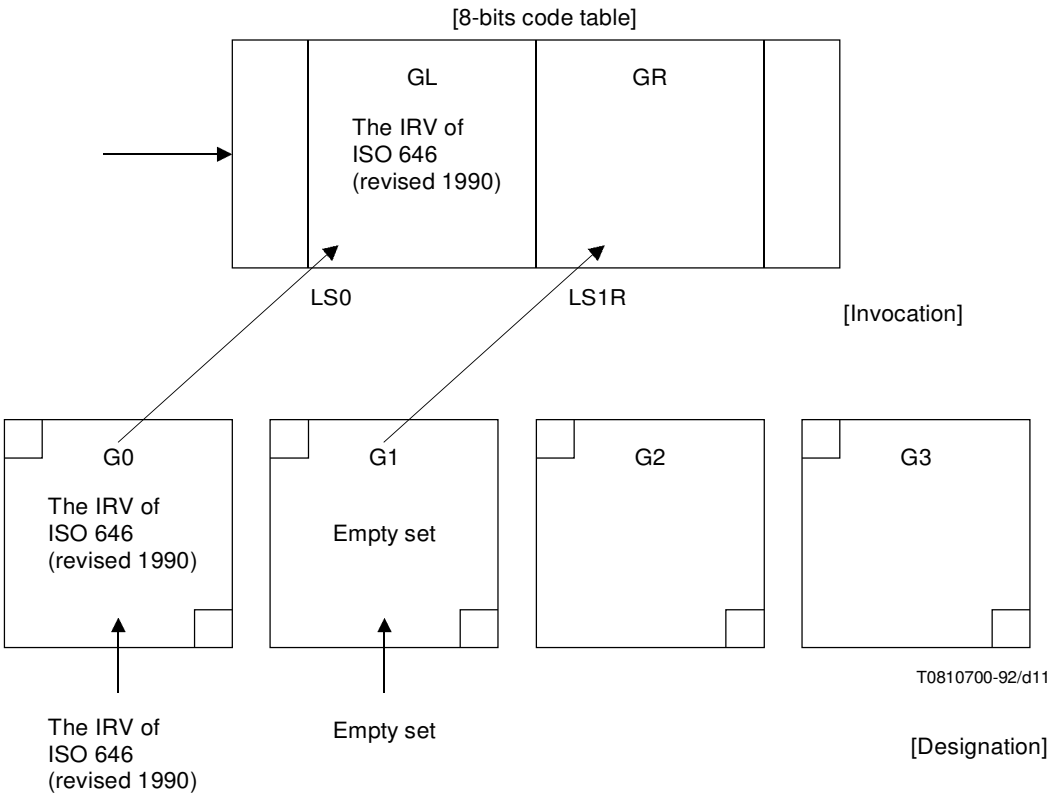

FIGURE 11/T.505

Code extension features (in the case of basic character sets)

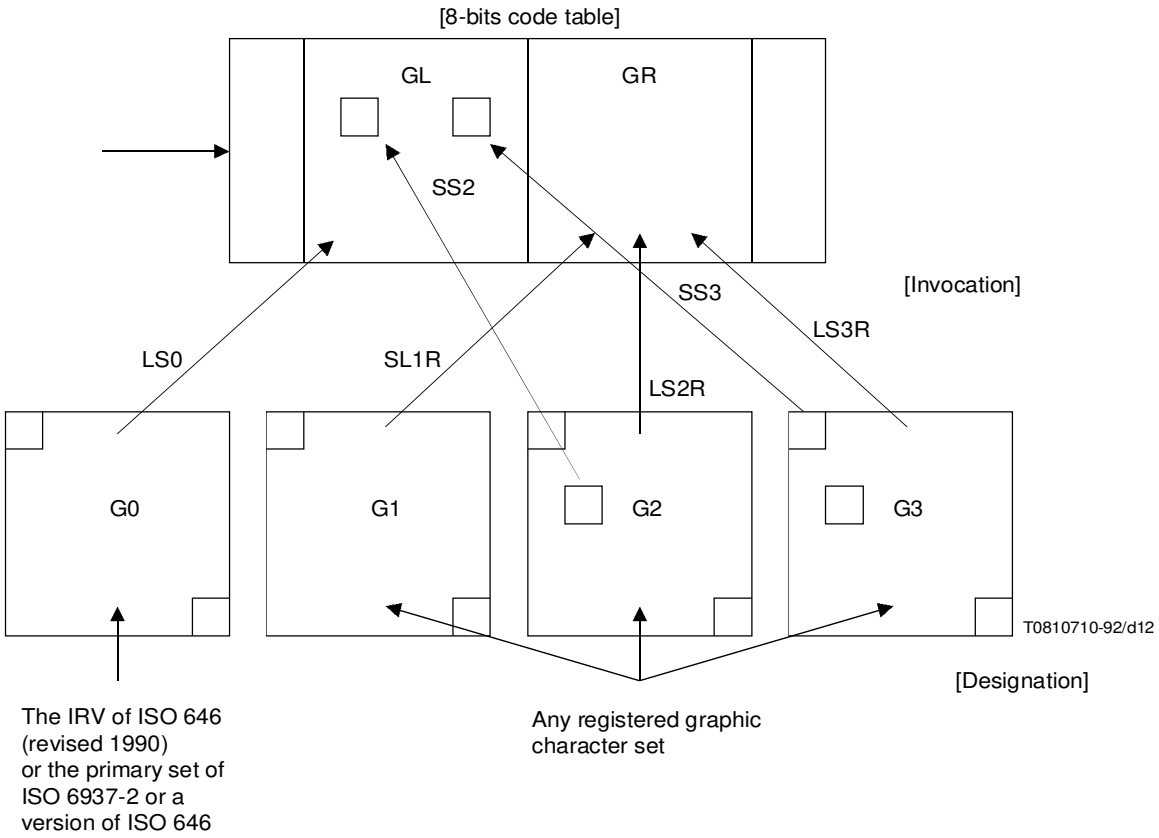

FIGURE 12/T.505

Code extension features (in the case of all possible character sets)

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 must 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, also using the presentation attribute "code extension announcers".

Note – In accordance with [CCITT Recommendation T.416/ISO 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 clause 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. Values of 150, 200, 300 and 400 BMUs are basic; the use of any other value in a document is non-basic and must 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.

6.5.1.6 *Character spacing*

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

Note 1 – A character spacing of 160 BMUs is provided for use with Korean Han-gul characters.

Note 2 – SHS parameters of 0, 1, 2, 3 and 4 are currently provided. The use of parameters 5 and 6 is currently being studied for use with Chinese characters.

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 must be indicated in the document profile.

The values of character path and line spacing 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 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 must 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.

Note – A character orientation of 0° in the case of vertical writing is typically used when it is required to mix ideogram characters with Latin characters. In order to achieve this, character strings orientated at 0° and 90° must be defined in separate basic components.

6.5.1.9 *Emphasis*

The following modes of emphasizing graphic characters may be distinguished:

– normal rendition;

– normal intensity;

– increased intensity (bold);

– italicized;

– not italicized;

– underlined;

– doubly underlined;

– not underlined;

– crossed out;

– not crossed out.

All the above modes of emphasis are basic. If no default mode is explicitly specified in the document profile, then the default mode is normal 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 "normal rendition". Mutually exclusive modes are normal/increased intensity, italicized/not italicized, 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.

Normal rendition cancels the effect of all methods of emphasis that are currently in operation and specifies that the text should 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.

6.5.1.10 *Tabulation*

Tabulation stop positions may be specified at any character 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 Recommendation T.416/ISO 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 can 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.

6.5.1.11 *Indentation*

Indentation is the distance between the first character on a line of content and the position of the margin position 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 Recommendation T.416/ISO 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 may not be changed within a content portion.

6.5.1.12 *Alignment*

This feature is concerned with how the first and last characters on each line of character content is 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 Recommendation T.416/ISO 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.

6.5.1.13 *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 itemization 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 § 6.5.1.11). 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 CR (Carriage Return) is used as the separator.

The features provided correspond to examples 10.1 to 10.4 shown in Figure 10 of [CCITT Recommendation T.416/ISO 8613-6].

First line format is specified by the presentation attributes “first line offset” and “itemization”; there are no restrictions on the values that may be specified.

6.5.1.14 *Widow and orphan sizes*

The widow size specifies the minimum number of lines of content that must 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 must 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 must 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 the body area of pages.

6.5.1.15 *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 Annex A.

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.

If the use of a particular font is explicitly specified, the character spacing used is determined from the attributes of the font. In this case constant or variable character spacing may be used, depending on the particular font specified. If no font is explicitly specified, then constant character spacing is used as specified by the presentation attribute “character spacing” (see § 6.5.1.6).

6.5.1.16 *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 can be specified by the control function SRS as defined in [CCITT Recommendation T.416/ISO 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 should ignore this control function when reverse character string layout and presentation is not supported.

6.5.1.17 *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.

6.5.1.18 *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 is determined from the attributes associated with the font used and may vary from one line to the next. This process is application dependent.

6.5.1.19 *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 Recommendation T.416/ISO 8613-6].

6.5.1.20 *Line breaks*

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

6.5.1.21 *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.

6.5.1.22 *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.

6.5.1.23 *Use of control functions*

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

SHS – select horizontal 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 – set 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);

PTX – parallel texts
(allowed parameters: any);

PLD – partial line down;

PLU – partial line up;

BPH – break permitted here;

NBH – no break here;

JFY – no justify;

SUB – substitute character;

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 points in § 6.5.1.

6.5.1.24 *Formatting the content*

All formatting of the content must be carried out by the imaging process and not by the content layout process (see [CCITT Recommendation T.416/ISO 8613-6]). Thus 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 clause 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;

– clipping;

– image dimensions.

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

6.5.2.2 *Raster graphics content architectures*

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 can 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 T.4, either the one dimensional or two dimensional encoding scheme may be used. Also the "bit-map encoding" scheme defined in [CCITT Recommendation T.417/ISO 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, it is required that both the coding attributes "number of lines" and "number of pels per line" are specified. The value of these attributes shall be a positive number; otherwise no restriction is placed on the values that may be specified. 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 raster graphics content architecture class, then the default encoding method is assumed to be T.6.

6.5.2.4 *Pel path and line progression*

The pel path and line progression supported by this profile 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 must 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".

Note 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 25.4 mm respectively when the BMU is interpreted as 1/1200 inch.

Note 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 graphic content*

A document may contain graphic images composed of geometric graphic controls encoded as CGM metafiles in accordance with ISO 8632. Each CGM figure must 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 in a resource document.

6.6.2 *External documents*

In the case of processable and formatted processable, either the generic logical structure or the generic layout structure or both of these structures may be contained in an external document.

6.6.3 *Borders*

Borders may be specified for all the frame types defined in §§ 6.3.5 and 6.3.6 using the attribute “Borders”. All the features of borders specified in [CCITT Recommendation T.412/ISO 8613-2], § 5.4.3, may be specified. The use of borders is a non-basic feature and must be indicated in the document profile. Borders cannot 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 optional for objects.

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 for which the attribute is specified. This facilitates the processing of documents. A list of these identifiers is given in Table 2/T.505.

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 § 8.3.

TABLE 2/T.505

List of number string identifiers

Logical constituent	*Numeric string identifier*
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	*Numeric string identifier*
DocumentLayoutRoot	0
PageSet	1
Page	2
RectoPage	3
VersoPage	4
CompositeHeader	5
CompositeBodyVariable	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

Note – The value of each numeric string identifier is unique for constituents within either the logical or layout structures. Also the number string identifiers are unique within the series of hierarchically related profiles to which this profile belongs.

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 must 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 character set is the minimum subrepertoire of ISO 6937-2. The control functions 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.

6.6.6 *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 HEADERFOOTERSTRING (see Note).

The <string-literal> fields are optional and are predefined 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 subrepertoire if appropriate, may be used provided that it is designated and invoked by the appropriate code extension announcer and indicated in the document profile as a non-basic value. 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 initialized 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).

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 can 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".

Note – Unless otherwise stated, the macros referred to in this clause are defined in § 7.3.1.

6.6.7 *Segment numbering*

As described in § 6.2.3.4, the constituent Number contains a content generator which when evaluated during the layout process produces an identifier which serves to identify the Numbered Segment to which the constituent Number belongs.

The format of this identifier 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 subrepertoire if appropriate, may be used provided that it is designated and invoked by the appropriate code extension announcer 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 and separator 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 and the document may contain any number of different bindings of these types. For example, "prefix-1" and "suffix-2" may be used to carry the prefix and suffix strings used in the first and second numbered segments.

These bindings can be initialized 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 can also be respecified at any level within the numbering scheme. The initialization of bindings is specified by the macro INITIALIZEANY.

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 can 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 initialized 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 optional and 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 can 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 segment. This binding is also referred to in a content generator carried by the constituent 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 clause are defined in § 7.3.1.

6.6.8 *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 predefined 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 subrepertoire if appropriate, may be used provided that it is designated and invoked by the appropriate code extension announcer 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 provided to represent footnote numbers. This may be initialized to any non-negative number at the logical root or on any Passage (see specification of the macro INITIALIZEFNOTE).

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.

6.6.9 *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 must 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 character set is the minimum subrepertoire of ISO 6937-2. The control functions CR and LF and code extension control functions may also be used within the character string but no other control functions are allowed.

6.6.10 *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 must 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 character set is the minimum subrepertoire of ISO 6937-2. The control functions CR, LF 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 DocumentProfile. This constituent must be specified in every document.

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

i) document constituent information;

ii) document identification information;

iii) document default information;

iv) non-basic characteristics information;

v) 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 generic layout structure or both of these 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 class information*

This information indicates the content architecture classes used in the document (see §§ 6.5.1.2, 6.5.2.2 and 6.5.3.2). It is mandatory to specify this information using the attribute "Content architecture class".

6.7.2.4 *Interchange format class information*

This information indicates the interchange format class used to represent the document (see § 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 ISO standard or CCITT Recommendation to which the document conforms. It also specifies a calendar date, which indicates that the document conforms to the version of the ISO standard or CCITT 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 § 6 of this profile. 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 [T.410-Series of CCITT Recommendations and ISO 8613].

Default values for the following groups of attributes can 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 in the document profile when such a value is used in the document.

The following types of non-basic attributes can be specified:

– document profile character sets;

– comment character sets;

– alternative representation character sets;

– page dimensions;

– medium-type;

– layout path;

– 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 assumed to be designated and invoked at the beginning of these character strings are specified by the document profile attribute "profile character sets".

The character sets that are designated and invoked by the attribute “profile character sets” are subject to the following restrictions:

i) G0 set: only the IRV of ISO 646 (revised 1990), the primary set of ISO 6937-2 and a version of ISO 646 may be designated for this set; these graphic character sets may only be invoked in GL.

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

iii) The empty set must be designated into G1 and invoked into GR if no other specific set is invoked into GR.

If the attribute “profile character sets” is not specified, then the character set designated and invoked is assumed to be the minimum subrepertoire of ISO 6937-2.

When the Teletex subrepertoire of ISO 6937-2 is needed, the primary set and the supplementary set of Recommendation T.61 are designated and invoked in this attribute.

6.7.4.2 *Comment character sets*

The character sets assumed to have been designated and invoked at the beginning of the character strings specified by the attributes “user readable comments” (see § 6.6.9) and “user visible name” (see § 6.6.10) are specified using the document profile attribute “comment character sets”.

It also specifies 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” should be announced by appropriate code extension announcers. The use of G0 and LS0 should always be announced by appropriate code extension announcers. The use of G0 set and LS0 should always be announced. Other code extension announcers are to be specified according to the requirements of a particular document. The restrictions on the use of code extension techniques as defined in § 6.5.1.4 also apply.

All the graphic character sets which may be used in the attribute “user readable comments” and “user visible name” should be designated in the “comments character sets”.

There are no restrictions concerning the numbers 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 and invocation to the same GL or GR overrides the previous GL or GR.

If the attribute “comments character sets” is not specified, then the character set designated and invoked is assumed to be the minimum subrepertoire of ISO 6937-2.

When the Teletex subrepertoire of ISO 6937-2 is needed, the primary set and the supplementary set of Recommendation T.61 are designated and invoked in this attribute.

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 graphic character sets described in § 6.7.4.1 is also applied. If this attribute is not explicitly specified in the document profile, the minimum subrepertoire of ISO 6937-2 is used in the attribute "alternative representation".

When the Teletex subrepertoire of ISO 6937-2 is needed, the primary set and the supplementary set of Recommendation T.61 are designated and invoked in this attribute.

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".

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 Note);
- 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 Recommendation T.414/ISO 8613-4].

The string of characters used in the document management attributes must belong to the character set 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 subrepertoire of ISO 6937-2.

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 section specifies the definition of the constituent constraints which can 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 Figures 13/T.505, 14/T.505 and 15/T.505. 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, in effect, define the allowed structures that are supported by this profile.

The structure diagrams illustrating the layout structures are shown in Figures 16/T.505, 17/T.505 and 18/T.505. The macros indicated in these diagrams are defined in § 7.4.1.

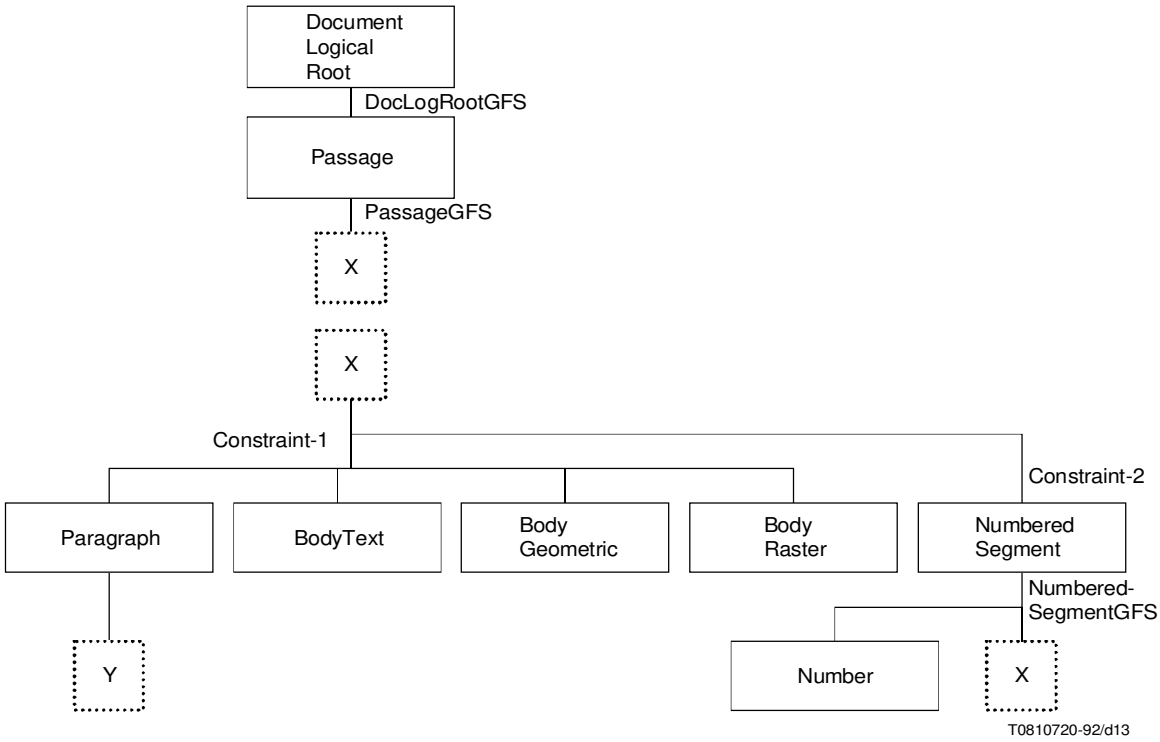

FIGURE 13/T.505

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

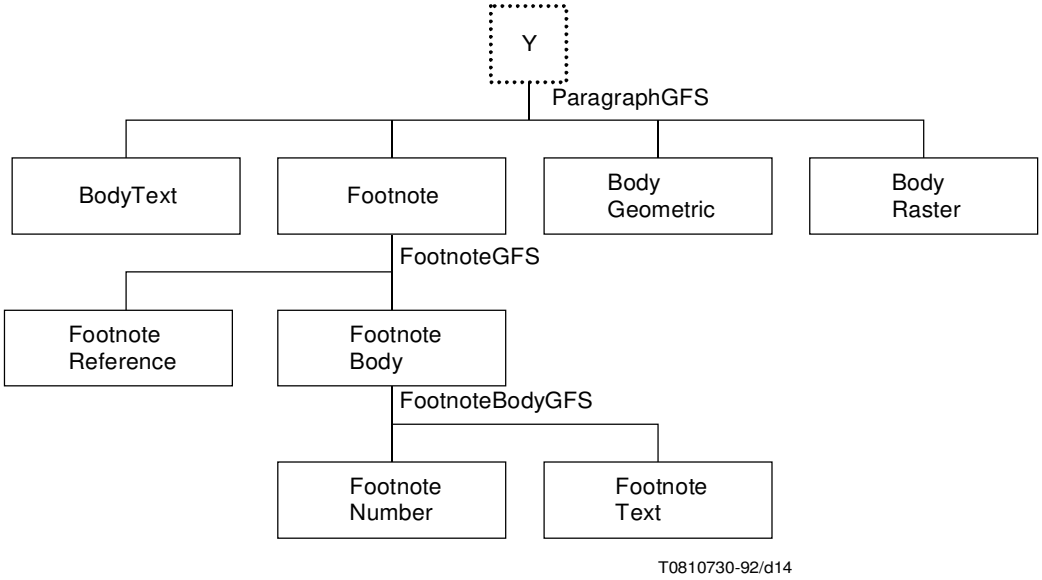

FIGURE 14/T.505

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

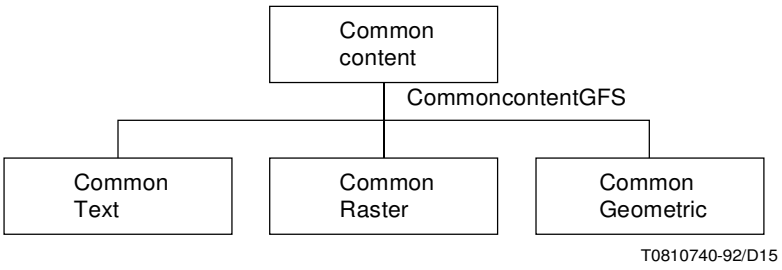

FIGURE 15/T.505

The "common" part of the generic logical structure

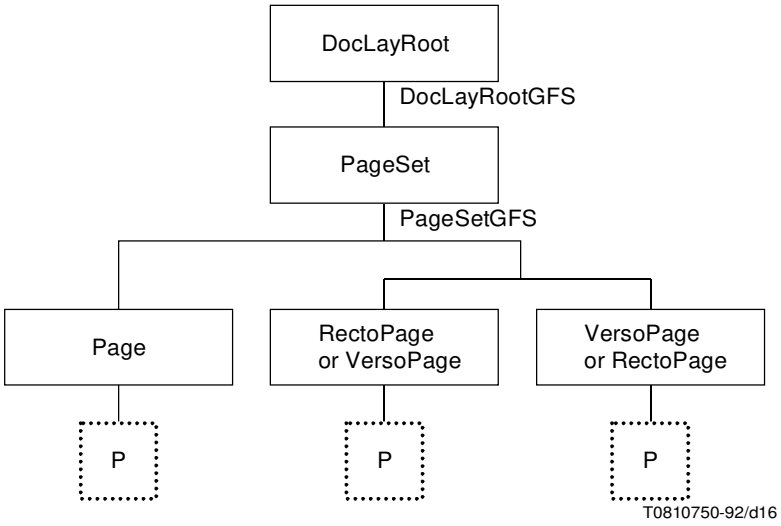

FIGURE 16/T.505

The layout structure – document root and page sets

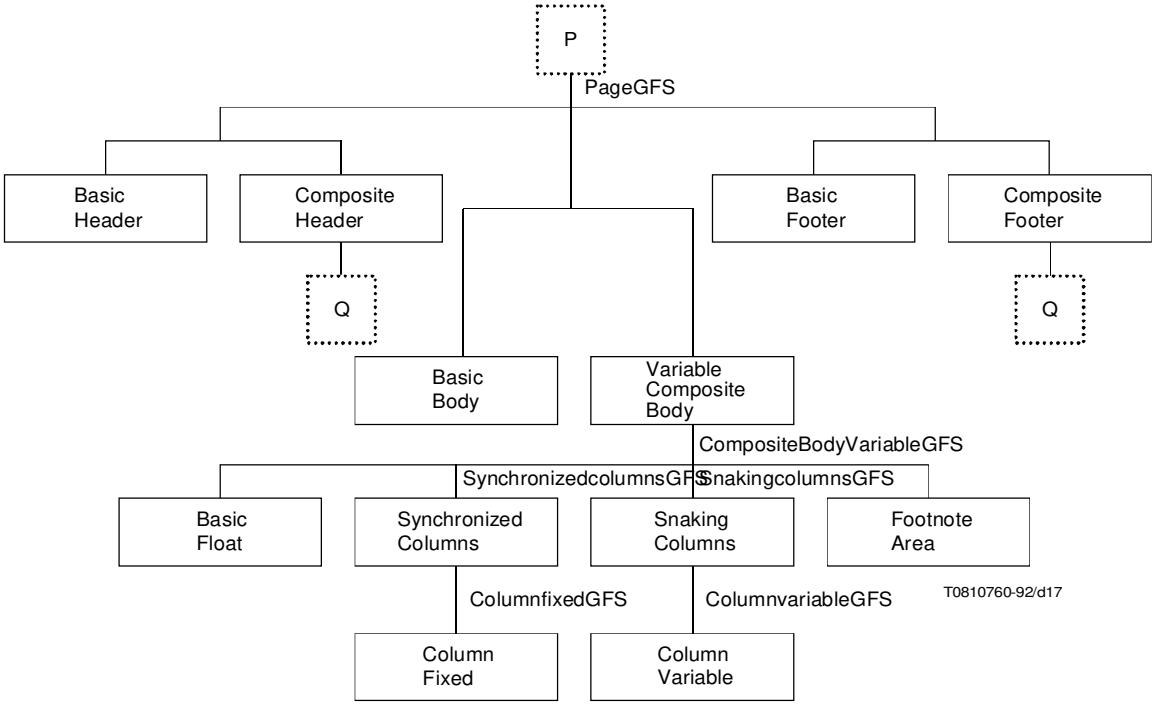

FIGURE 17/T.505

The layout structure – the page structure

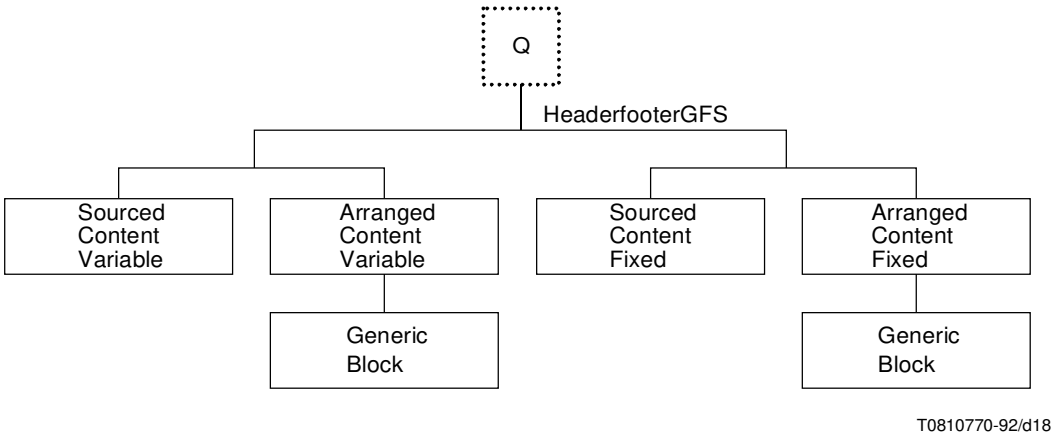

FIGURE 18/T.505

The layout structure – the header and footer frame structure

7.2 *Document profile 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(DAC,     "DocumentProfile (Document-architecture-class)")
```

```
DEFINE(NominalPageSizes,        "
        {REQ #horizontal-dimension {6922},
         REQ #vertical-dimension {9920}        -- ISO A5 portrait --}
        |{REQ #horizontal-dimension {9920},
         REQ #vertical-dimension {6922}        -- 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 {14031},
         REQ #vertical-dimension {19843}       -- ISO A3 portrait --}
        |{REQ #horizontal-dimension {19843},
         REQ #vertical-dimension {14031}       -- ISO A3 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 --}
        |{REQ #horizontal-dimension {12141},
         REQ #vertical-dimension {17196}       -- Japanese legal portrait --}
        |{REQ #horizontal-dimension {17196},
         REQ #vertical-dimension {12141}       -- Japanese legal landscape --}
        |{REQ #horizontal-dimension {8598},
         REQ #vertical-dimension {12141}       -- Japanese letter portrait --}
        |{REQ #horizontal-dimension {12141},
         REQ #vertical-dimension {8598}        -- Japanese letter landscape --}
                                        ")

DEFINE(GRAPHICRENDITIONS   "
       {'cancel'|'increased-intensity'
        |'italicised'|'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-italicised'|'not-underlined'|'not-crossed-out'}...
                                        ")
```

-- *Macro defining permissible code extension announcers. This macro may be used in each constituent constraint or presentation style constraint. Note that all the values are basic.* --

```
DEFINE(CDEXTEN, "
        ESC 02/00 05/00,            -- LS0 --
        [ESC 02/00 05/03]   ,       -- LS1R --
        [ESC 02/00 05/05]   ,       -- LS2R --
        [ESC 02/00 05/07]   ,       -- LS3R --
        [ESC 02/00 05/10]   ,       -- SS2 --
        [ESC 02/00 05/11]           -- SS3 --
                        ")
```

-- *Macro defining code extension announcers for profile default values* --

```
DEFINE(DAP-DEFAULT-CDEXTAN,             "$CDEXTAN")
```

-- *Macros defining final character for designation* --

```
DEFINE(FCORE,     "04/02
        -- the 94 characters of the IRV of ISO 646 (revised 1990) (i.e ASCII) -- ")

DEFINE(F646,      "
        -- a final character designating any version of ISO 646, except 04/02 -- ")

DEFINE(F94S,      "
        -- a final character designating any registered 94 single byte graphic character set -- ")

DEFINE(F94M,      "
        -- a final character designating any registered 94 multi byte graphic character set -- ")

DEFINE(F96S,      "
        -- a final character designating any registered 96 single byte graphic character set -- ")

DEFINE(F96M,      "
        -- a final character designating any registered 96 multi byte graphic character set -- ")

DEFINE(FEMPTY,    "07/14
        -- the empty set -- ")
```

-- *Macros defining designation sequences* --

```
DEFINE(DEG-CORE-G0,    "ESC 02/08 $FCORE")
        -- designate the 94 characters of the IRV of ISO 646 to G0 --

DEFINE(DEG-646-G0,      "ESC 02/08 $F646")
        -- designate any version of ISO 646, except 04/02, to G0 --

DEFINE(DEG-ANY-G1,      "
        {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 02/10 $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 02/11 $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 --

-- Macros defining shift functions --

DEFINE(LS0,         "00/15")
        -- locking shift invoking G0 to GL --

DEFINE(LS1R,        "ESC 07/14")
        -- locking shift invoking G1 to GR --

DEFINE(LS2R,        "ESC 07/13")
        -- locking shift invoking G2 to GR --

DEFINE(LS3R,        "ESC 07/14")
        -- locking shift invoking G3 to GR --

DEFINE(SS2,         "08/14")
        -- single shift invoking G2 to GL --

DEFINE(SS3,         "08/15")
        -- single shift invoking G3 to GL --

-- 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 default graphic character sets --

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 to be used in any other part of this DAP specification. --

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

-- Macro defining non-basic 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,        "
        {$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 comments character sets --

DEFINE(COMCHAR,         "
        {ESC 02/00 05/00,           -- LS0 --
         [ESC 02/00 05/03],         -- LS1R --
         [ESC 02/00 05/05],         -- LS2R --
         [ESC 02/00 05/07],         -- LS3R --
         [ESC 02/00 05/10],         -- SS2 --
         [ESC 02/00 05/11]},        -- SS3 --
        {$DEG-CORE-G0 [LS0]
         |$DEG-646-G0 [LS0]},
        {{$DEG-ANY-G1 [$LS1R]
         |$DEG-ANY-G2 [$LS2R]
         |$DEG-ANY-G3 [$LS3R]}...
         |$DEG-EMPTY-G1 $LS1R}}
                        ")

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

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

7.2.2 *Constituent constraints*

7.2.2.1 *DocumentProfile* {

```
CASE $DAC OF    {

    $FDA:       PERM    Generic-layout-structure        {'factor-generator-set'},
                REQ     Specific-layout-structure       {'present'},
                PERM    Presentation-styles             {'present'}

    $PDA :      PERM    Generic-layout-structure        {'complete-generator-set'},
                REQ     Generic-logical-structure       {'complete-generator-set'},
                REQ     Specific-logical-structure      {'present'},
                PERM    Presentation-styles             {'present'},
                PERM    Layout-styles                   {'present'}

    $FPDA:      REQ     Generic-layout-structure        {'complete-generator-set'},
                REQ     Specific-layout-structure       {'present'},
                REQ     Generic-logical-structure       {'complete-generator-set'},
                REQ     Specific-logical-structure      {'present'},
                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          {-- to be supplied --},

   PERM Document-application-profile-defaults  {

-- document architecture defaults --

   CASE $DAC OF     {
        $FDA:       PERM #content-architecture-class  {$FC|$FPC}}
        $PDA:       REQ  #content-architecture-class  {$FC|$PC|$FPC}}
        $FPDA:      REQ  #content-architecture-class  {$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 #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                           {'0-degrees'
                                                        |'90-degrees'
                                                        |'180 degrees'
                                                        |'270-degrees'},
        PERM #character-spacing                        {ANY_VALUE},
        PERM #character-orientation                    {'0-degrees'
                                                        |'90-degrees'},
        PERM #code-extension-announcers                {$CDEXTAN},
        PERM #first-line-format                        {ANY_VALUE},
        PERM #graphic-character-sets                   {$BASIC-GRCHAR,
                                                        $DAP-DEFAULT-GRCHAR},
```

```
        PERM #graphic-character-subrepertoire          {ANY_VALUE},
        PERM #graphic-rendition                        {$GRAPHICRENDITIONS},
        PERM #indentation                              {ANY_VALUE},
        PERM #initial-offset                           {ANY_VALUE},
        PERM #itemisation                              {ANY_VALUE},
        PERM #kerning-offset                           {ANY_VALUE},
        PERM #line-layout-table                        {ANY_VALUE},
        PERM #line-progression                         {'90-degrees'
                                                        |'270-degrees'},
        PERM #line-spacing                             {ANY_VALUE},
        PERM #orphan-size                              {ANY_VALUE},
        PERM #proportional-line-spacing                {ANY_VALUE},
        PERM #widow-size                               {ANY_VALUE}}
                                                    },
    PERM #raster-graphic-content-defaults           {
        PERM #clipping                                 {ANY_VALUE},
        PERM #image-dimensions                         {ANY_VALUE},
        PERM #pel-spacing                              {ANY_VALUE},
        PERM #spacing-ratio                            {ANY_VALUE},
        PERM #compression                              {ANY_VALUE}}
                                                    },
    REQ  Document-architecture-class                {$FDA|$PDA|$FPDA},
    REQ  Content-architecture-classes               {[$FC],[$PC],[$FPC],[$FPR],[$FPG]},
    REQ  Interchange-format                         {'if-a'},
    REQ  Oda-version                                {REQ #standard-or-recommendation("ISO 8613"),
                                                     REQ #publication-date(-- to be supplied --)}},

-- 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 {9241..14030}},
                              REQ #vertical-dimension
                                 {REQ #fixed-dimension {12401..19840}}}
                                                    -- up to ISO A3 portrait --
                            |{REQ #horizontal-dimension
                                 {REQ #fixed-dimension {12401..19840}},
                              REQ #vertical-dimension
                                 {REQ #fixed-dimension {9241..14030}}}
                                                    -- up to ISO A3 landscape --
                            |{REQ #horizontal-dimension
                                 {REQ #fixed-dimension {9241..13200}},
                              REQ #vertical-dimension
                                 {REQ #fixed-dimension {12401..20400}}}
                                                    -- up to ANSI B portrait -
                            |{REQ #horizontal-dimension
                                 {REQ #fixed-dimension {12401..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 NAL
   is non-basic --

   PERM Medium-type          {PMUL
                                 {PERM #nominal-page-size{$NominalPageSizes},
                                  PERM #side-of-sheet{'recto'|'verso'}}},

-- all values of "medium type" are non-basic --

   PERM Layout-path          {'0-degrees','90-degrees','180-degrees'},

   PERM Border               {ANY_VALUE},

   PERM Ra-gr-coding-attributes    {
        PERM #compression                    {ANY_VALUE}},

   PERM Presentation-features      {
      PERM      #character-presentation-features          {
        PERM #character-orientation                  {'90-degrees'},
        PMUL #character-path                         {'90-degrees'
                                                       |'180-degrees'
                                                       |'270-degrees'},
        PMUL #character-spacing                      {<100},
        PMUL #graphic-character-sets                 {ANY_EXCEPT $BASIC-GRCHAR},
        PMUL #graphic-character subrepertoire        {ANY_VALUE},
        PMUL #line-spacing                           {ANY_EXCEPT 150,200,300,400},
        PERM #line-progression                       {'90-degrees'}},

   PERM #raster-graphics-presentation-features {
        PMUL #pel-spacing         {REQ #length{ANY_EXCEPT 16,12,8,6,5,4,3,2,1},
                                   REQ #pel-spaces{ANY_EXCEPT 1}}},

-- additional document characteristics --

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

-- the format of the parameter "font-reference" is defined in § 8.4 --

-- 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_STRING},
   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 Preparers                          {ANY_VALUE},
    PERM Owners                             {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)
                                |CHO({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(<constraint-3>[<constraint-1>]
                                [<constraint-2>]);

<constraint-3>              ::= 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  <constraint-4>);

<constraint-4>                  ::= 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(CommonText)
                                    |OBJECT_CLASS_ID_OF(PageNumber)
                                    |OBJECT_CLASS_ID_OF(CommonRaster)
                                    |OBJECT_CLASS_ID_OF(CommonGeometric);
                                ")

DEFINE(N,    "
<n>                         ::=
        -- any string of characters from the set of characters: "0"..."9"  --
                ")

DEFINE(PREFIXES,"
<prefixes>                      ::=  'prefixes-' <n>;
$N
                        ")

DEFINE(SUFFIXES,"
<suffixes>                      ::=  'suffixes-'<n>;
$N

                        ")

DEFINE(SEPARATORS,    "
<separators>                    ::=  'separators-'<n>;
$N

                        ")

DEFINE(NUMBERS,"
<numbers>                       ::=  'numbers-'<n>;
$N

                        ")

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

                        ")

DEFINE(STRINGFUNCTION,      "
<string-function>               ::=  MK_STR|U_ALPHA|L_ALPHA|U_ROM|L_ROM;
                                ")
```

```
DEFINE(INITIALISEANY,   "
                         {REQ #binding-identifer{<prefixes>},
                          REQ #binding-value{ANY_STRING}
                         |{REQ #binding-identifer{<suffixes>},
                          REQ #binding-value{ANY_STRING}
                         |{REQ #binding-identifer{<separators>},
                          REQ #binding-value{ANY_STRING}
                         |{REQ #binding-identifer{<numbers>},
                          REQ #binding-value{{ANY_STRING}
                         |{REQ #binding-identifer{<numberstring>},
                          REQ #binding-value{ANY_STRING}

$PREFIXES
$SUFFIXES
$SEPARATORS
$NUMBERS
$NUMBERSTRING
                    ")

DEFINE(USENUMBERSTRINGS, "
                         {REQ #binding-identifer{<numberstring>},
                          REQ #binding-value{<hierarchic-expr>|<simple-expr>}

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

<simple-expr>              ::=  <string-function>(B_REF(CURR-OBJ)(<numbers>))
                                |<string-function>(ORD(CURR_OBJ));

$NUMBERSTRING
$SEPARATORS
$NUMBERS
$STRINGFUNCTION
                         ")

DEFINE(USENUMBERS,   "
                         {REQ #binding-identifer{numbers>},
                          REQ #binding-value
                                  {INC(B_REF(PREC(CURR_OBJ))(<numbers>)}

$NUMBERS
                    ")

DEFINE(SEGMENTNUMBER,    "
<string-expr>            ::=  [<pre-st>]<num-st>[suf-st>];
<num-str>                ::=  B_REF(SUP(CURR_OBJ))(<numberstring>);
<pre-st>                 ::=  B_REF(SUP(CURR_OBJ))(<prefixes>)
                              |ANY_STRING;

<suf-st>                 ::=  B_REF(SUP(CURR_OBJ))(<suffixes>)
                              |ANY_STRING;

$NUMBERSTRING
$PREFIXES
$SUFFIXES

                         ")

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

```
DEFINE(INCFNOTENUMBER       "
                            {REQ #binding-identifer{"fnotenumber"},
                             REQ #binding-value{INC(B_REF(PREC (CURR-OBJ))(fnotenumber)}
                            ")

DEFINE(FNOTENUMBERSTRING                "
                            {REQ #binding-identifer{"fnotestring"},
                             REQ #binding-value{<string-function>
                                                B_REF(CURR_OBJ)(fnotenumber)}

<string-function>           ::=  MK_STR|U_ALPHA|L_ALPHA|U_ROM|L_ROM;
                                 ")

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

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

<num-str>                   ::=  B_REF(SUP(CURR_OBJ))(fnotestring);
                        ")

DEFINE(HEADERFOOTERSTRING               "
<string-expr>               ::=  [ANY_STRING]{<string-function><num-exp>}[ANY_STRING];

<num-exp>                   ::=  B_REF(SUP(CURR_INST(<class-or-type1>,
                                   CURR_OBJ)))(PGnum)
                                   |B_REF(CURR_INST(<class-or-type2>,
                                   CURR_OBJ)))(PGnum);

<class-or-type1>            ::=  FRAME;

<class-or-type2>            ::=  PAGE
                                 |OBJECT_CLASS_ID_OF(Page)
                                 |OBJECT_CLASS_ID_OF(RectoPage)
                                 |OBJECT_CLASS_ID_OF(VersoPage);

$STRINGFUNCTION
                                 ")
```

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**	**{$NumberedSegmentGFS},**
REQ	**Application-comments**	**{REQ #constraint-name {"2"},**
		PERM #external-data {ANY_VALUE}},
PERM	**Bindings**	**{PMUL{$INITIALISEANY},**
		PERM{$USENUMBERS},
		PERM{$USENUMBERSTRING}}

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-Style2)}}

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	**Content-portions**	**{CONTENT_ID_OF(Character- content-portion)+},**
PERM	**Application-comments**	**{REQ #constraint-name {"3"},**
		PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM	**Layout-style**	**{STYLE_ID_OF(L-Style3)},**
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-Style2)}}

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-Style3)},**
PERM Presentation-style	**{STYLE_ID_OF(P-Style1)},**
PERM Content-architecture-class	**{$FC\|$PC|$FPC},**
PERM Content-portions	**{CONTENT_ID_OF(Character-content-portion)+}}**

-- the attribute "content portion" must be specified either in the specific or generic part, otherwise the attribute "resource" must 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-Style2)},**
PERM Content-portions	**{CONTENT_ID_OF(Geometric-content-portion)}}

-- the attribute "content portion" must be specified either in the specific or generic part, otherwise the attribute "resource" must 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-content-portion)}}**

-- *the attribute "content portion" must be specified either in the specific or generic part, otherwise the attribute "resource" must be specified* --

7.3.3.9 *Footnote: ANY-LOGICAL* {

GENERIC:

REQ	**Object-type**	**{'composite-logical-object'},**
REQ	**Generator-for-subordinates**	**{$FootnoteGFS},**
PERM	**Bindings**	**{{$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	**Application-comments**	**{REQ #constraint-name {"8"},** **PERM #external-data {ANY_VALUE}}**

SPECIFIC_AND_GENERIC:

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

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	**Content-portions**	**{CONTENT_ID_OF(Character-content-portion)+},**
PERM	**Application-comments**	**{REQ #constraint-name {"10"},** **PERM #external-data {ANY_VALUE}}**

SPECIFIC_AND_GENERIC:

PERM	**Layout-style**	**{STYLE_ID_OF(L-Style3)},**
PERM	**Presentation-style**	**{STYLE_ID_OF(P-Style1)},**
PERM	**Content-architecture-class**	**{$PC\|$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}}**

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}}}**

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	**Content-portions**	**{CONTENT_ID_OF(Character-content-portion)+},**
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-portion**	**{CONTENT_ID_OF(Character-content-portion)+},**
PERM	**Resource**	**{ANY_VALUE},**
PERM	**Layout-style**	**{STYLE_ID_OF(L-Style7)},**
PERM	**Presentation-style**	**{STYLE_ID_OF(P-Style4)},**
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 portion" or "resource" must be specified in the above constituent --

7.3.3.16 *PageNumber* {

GENERIC:

REQ	**Object-type**	**{'basic-logical-object'},**
REQ	**Object-class-identifier**	**{ANY_VALUE},**
REQ	**Content-generator**	**{$HEADERFOOTERSTRING},**
PERM	**Layout-style**	**{STYLE_ID_OF(L-Style7)},**
PERM	**Presentation-style**	**{STYLE_ID_OF(P-Style4)},**
PERM	**Content-architecture-class**	**{$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-content-portion)},
        PERM Resource                       {ANY_VALUE},
        PERM Layout-style                   {STYLE_ID_OF(L-Style8)},
        PERM Presentation-style             {STYLE_ID_OF(P-Style2)},
        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 portion" or "resource" must be specified in the above constituent --

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-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 portion" or "resource" must be specified in the above constituent --

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(VariableCompositeBody)
                                   |OBJECT_CLASS_ID_OF(BodyFrameVariable);

<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(PAGENUMBER,   "
                                 {REQ #binding-identifier{"PGnum"},
                                  REQ #binding-value{INC(B_REF(PREC(CURR-OBJ))("PGnum"))}
                                |{REQ #binding-identifier{"PGnum"},
                                  REQ #binding-value{ORD(CURR-OBJ)}
                            ")

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

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_VALUE},
        PERM User-visible-name           {ANY_VALUE}}
```

7.4.2.2 *Factor: ANY-PAGE: ANY-LAYOUT* {

```
GENERIC:
        REQ   Object-type                {'page'},
        REQ   Generator-for-subordinates {$PageGFS},
        CASE $DAC OF {
              $PDA:   PERM  Bindings     {$PAGENUMBER}
              $FPDA: PERM  Bindings      {$PAGENUMBER}
            }
SPECIFIC:
        PERM Object-type                 {'page'},
        REQ   Subordinates               {SUB_ID_OF(BasicHeader),
                                          SUB_ID_OF(CompositeHeader),
                                          SUB_ID_OF(VariableCompositeBody),
                                          SUB_ID_OF(BodyFrameVariable),
                                          SUB_ID_OF(BasicFooter),
                                          SUB_ID_OF(CompositeFooter)},
        CASE $DAC OF {
              $FPDA:  PERM  Bindings     {REQ #binding-identifier{"PGnum"},
                                          REQ #binding-identifier{>=0}}
            }
```

```
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'},
        REQ   Subordinates              {VIRTUAL}
SPECIFIC_AND_GENERIC:
        PERM Position                   {REQ #fixed-position
                                          {REQ #horizontal-position {ANY_VALUE},
                                           REQ #vertical-position {ANY_VALUE}}},
        PERM Dimension                  {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'},
        REQ   Subordinates              {VIRTUAL},
        CASE $DAC OF {
              $FPDA: REQ   Position     {REQ #fixed-position
                                          {REQ #horizontal-position {ANY_VALUE},
                                           REQ #vertical-dimension {ANY_VALUE}}},
                     REQ   Dimension    {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  Dimension       {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},
    REQ  Content-portions            {CONTENT_ID_OF(character-content-portion)+,
                                      CONTENT_ID_OF(raster-graphics-content-portion),
                                      CONTENT_ID_OF(geometric-graphics-content-portion)},
    PERM Presentation-style          {STYLE_ID_OF(P-style1)
                                      |STYLE_ID_OF(P-style2)
                                      |STYLE_ID_OF(P-style3)},
    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-orientation            ('0-degree|'90-degrees'),
            PERM #character-path                   ('0-degrees'|'90-degrees'
                                                   {|'180-degrees'|'270-degrees'}
            PERM #character-spacing                {ANY_VALUE},
            PERM #code-extension-announcers        {$CDEXTAN},
            PERM #first-line-offset                {ANY_VALUE},
            PERM #graphic-character-sets           {$BASIC-GRCHAR},
            PERM #graphic-character-subrepertoire  {ANY-VALUE},
            PERM #graphic-rendition                {$GRAPHICRENDITIONS},
            PERM #itemisation                      {ANY_VALUE},
            PERM #kerning-offset                   {ANY_VALUE},
            PERM #line-layout-table                {ANY_VALUE},
            PERM #line-spacing                     {150|200|300|400},
            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 Dimension                   {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'},
        REQ   Generator-for-subordinates        {$DocLayRootGFS},
        CASE $DAC OF {
                $PDA:   PERM   Bindings         {$INITIALISEPGNUM}
                $FPDA:  PERM   Bindings         {$INITIALISEPGNUM}
                        },
        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                       {'pageset'},
        REQ   Generator-for-subordinates        {$PageSetGFS},
        CASE $DAC OF {
                $PDA:   PERM   Bindings         {$INITIALISEPGNUM}
                $FPDA: PERM   Bindings          {$INITIALISEPGNUM}
                        },
        REQ   Application-comments              {REQ #constraint-name {"1"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC:
        PERM Object-type                        {'pageset'},
        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                   {REQ  #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                   {REQ  #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'}}},

        PERM Permitted-categories               {ANY_STRING}

        CASE SUPERIOR (VariableCompositeBody(Layout-Path)) OF {

        '270-degrees':      -- page layout A --
                REQ   Dimension                 {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   Dimension         {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   Dimension         {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'}}
                CASE SUPERIOR (VariableCompositeBody(Layout-Path)) OF {

                '270-degrees':      -- page layout A --
                        REQ   Dimension        {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   Dimension        {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   Dimension        {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'}},
        PERM Balance                       {ANY_VALUE}
        CASE SUPERIOR (VariableCompositeBody(Layout-Path)) OF {

        '270-degrees':        -- page layout A --
            REQ   Dimension          {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   Dimension          {REQ #horizontal-dimension
                                          {REQ #rule-b {ANY_VALUE}},
                                      REQ #vertical-dimension
                                          {REQ #fixed-dimension {ANY_VALUE}
                                          |REQ #maximum-size {'applies'}}},
            PER   Layout-path        {'90-degrees'|'270-degrees'}

        '180-degrees':        -- page layouts C and D --
            REQ   Dimension          {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'}}
```

```
            CASE SUPERIOR (VariableCompositeBody(Layout-Path)) OF {

            '270-degrees':      -- page layout A --
                  REQ  Dimension          {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  Dimension          {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  Dimension          {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:

            PERM 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  Dimension          {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  Dimension          {REQ #horizontal-dimension
                                              {REQ #rule-b {ANY_VALUE}
                                              |REQ #maximum-size {'applies'}},
                                           REQ #vertical-dimension
                                              {REQ #fixed-dimension {ANY_VALUE}
                                              |REQ #maximum-size {ANY_VALUE}}},
                  REQ   Layout-path       {'0-degrees'}

            '180-degrees':        -- page layouts C and D --
                  REQ   Dimension         {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'}},

            REQ   Permitted-categories {''Footnote''}

            CASE SUPERIOR (VariableCompositeBody(Layout-Path)) OF {

            '270-degrees':        -- page layout A --
                  REQ   Dimension         {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   Dimension         {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  Dimension          {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:
      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:
      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:
      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:
        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'}}

                CASE SUPERIOR (CompositeHeader|CompositeFooter(Layout-path)) OF {
                '270-degrees':
                        REQ   Dimension        {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   Dimension        {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
                                                {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'}},
                REQ   Dimension                 {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE},
                                                 REQ #vertical-dimension
                                                    {REQ #fixed-dimension {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 {"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}}}
                CASE SUPERIOR (CompositeHeader|Compositefooter(Layout-path)) OF {
                '270-degrees':
                        REQ   Dimension         {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}},
                                                 REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}
                                                    |REQ #rule-b {ANY_VALUE}}},
                        PERM  Layout-path       {'270-degrees'}
```

```
            '180-degrees':
                  REQ   Dimension          {REQ #horizontal-dimension
                                              {REQ #fixed-dimension {ANY_VALUE}
                                              |REQ #rule-b {ANY_VALUE}},
                                             REQ #vertical-dimension
                                              {REQ #fixed-dimension {ANY_VALUE}},
                  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:
      REQ   Generator-for-subordinates     {SEQ({OBJECT_CLASS_IS_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-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 Presentation-style              {STYLE_ID_OF(P-Style1)
                                            |STYLE_ID_OF(P-Style2)
                                            |STYLE_ID_OF(P-Style3)},
      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 Dimension                       {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)}
                        },
        PERM Application-comments               {REQ #constraint-name {"29"},
                                                 PERM #external-data {ANY_VALUE}}}
```

7.4.3.23 *SpecificBlock: BLOCK* {

```
SPECIFIC:
        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 #sameas{<object-id-expr> ::=  PREC-OBJ(CURR-OBJ);
                        |'null'},
        REQ #within{'page'}

                                        ")
```

7.5.2 *Factor: ANY-LAYOUT-STYLE* {

```
        REQ   Layout-style-identifier           {ANY_VALUE},
        PERM User-visible-name                  {ANY_STRING},
        PERM User-readable-comments             {ANY_STRING}}
```

7.5.3 *Layout style constituent constraints*

7.5.3.1 *L-Style1: ANY-LAYOUT-STYLE* {

-- this style is used for the constituent Passage only --

```
CASE Document-profile(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'}
                                                        }}
```

7.5.3.2 *L-Style2: ANY-LAYOUT-STYLE {*

-- this style is used for the constituents NumberedSegment and Paragraph --

```
CASE Document-profile(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}}
```

7.5.3.3 *L-style3: ANY-LAYOUT-STYLE {*

-- this style is used for the constituents BodyText, Number and FootnoteReference --

```
CASE Document-profile(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}}
```

7.5.3.4 *L-Style4: ANY-LAYOUT-STYLE {*

-- this style is used for the constituent Footnote only --

```
PERM  Indivisibility                            {'page'|'null'},
PERM  Same-layout-object                        {$SameLayoutObject}}
```

7.5.3.5 *L-Style5: ANY-LAYOUT-STYLE {*

-- this style is used for the constituents BodyRaster and BodyGeometric --

```
CASE Document-profile(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'}
                                                        },
```

```
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 FootnoteText* --

```
REQ    Layout-category                {"Footnote"},
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}}}
```

7.5.3.7 *L-Style7: ANY-LAYOUT-STYLE* {

-- *this style is used for the constituents 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}}}
```

7.5.3.8 *L-Style8: ANY-LAYOUT-STYLE* {

-- *this style is used for the constituents 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 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}}}
```

7.6 *Presentation style constraints*

7.6.1 *Macro definitions*

No macro definitions are applicable to this clause.

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   User-readable-style            {ANY_STRING}}
```

7.6.3 *Presentation style constituent constraints*

7.6.3.1 *P-Style1: ANY-PRESENTATION-STYLE {*

-- this style is used for the constituents BodyText, Number, FootnoteNumber, FootnoteReference and FootnoteText --

```
PERM  Presentation attributes  {
        PERM #character-attributes   {
                PERM #alignment                          {ANY_VALUE},
                PERM #character-spacing                  {ANY_VALUE},
                PERM #character-fonts                    {ANY_VALUE},
                PERM #character-orientation              {'0 degrees'
                                                          |'90-degrees'},
                PERM #character-path                      {'0-degrees'
                                                          |'90-degrees'
                                                          |'180-degrees'
                                                          |'270-degrees'},
                PERM #code-extension-announcers          {$CDEXTAN},
                PERM #first-line-offset                  {ANY_VALUE},
                PERM #graphic-character-sets             {$PERMIT-GRCHAR},
                PERM #graphic-character-subrepertoire    {ANY_VALUE},
                PERM #graphic-rendition                  {$GRAPHICRENDITIONS},
                PERM #indentation                        {ANY_VALUE},
                PERM #itemisation                        {ANY_VALUE},
                PERM #kerning-offset                     {ANY_VALUE},
                PERM #line-progression                   {'90-degrees'
                                                         |'270-degrees'},
                PERM #line-spacing                       {ANY_VALUE},
                PERM #line-layout-table                  {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 constituents BodyGeometric and CommonGeometric --

```
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}}
```

7.6.3.3 *P-Style3: ANY-PRESENTATION-STYLE {*

-- this style is used for the constituents BodyRaster and CommonRaster --

```
PERM  Presentation attributes  {
        PERM #raster-graphics-attributes   {
                PERM #image-dimensions                   {ANY_VALUE},
                PERM #clipping                           {ANY_VALUE},
                PERM #pel-spacing                        {{REQ #length {ANY_VALUE},
                                                           REQ #pel-spaces{ANY_VALUE}}
                                                          |'null'},
        PERM #spacing-ratio                              {ANY_VALUE}}}
```

7.6.3.4 *P-Style4: ANY-PRESENTATION-STYLE* {

-- his style is used for the constituents CommonText and PageNumber --

```
PERM  Presentation attributes  {
      PERM #character-attributes   {
            PERM #alignment                              {ANY_VALUE},
            PERM #character-spacing                      {ANY_VALUE},
            PERM #character-fonts                        {ANY_VALUE},
            PERM #character-orientation                  {'0-degrees'
                                                          |'90-degrees'},
            PERM #character-path                         {'0-degrees'
                                                          |'180-degrees'
                                                          |'270-degrees'},
            PERM #code-extension-announcers              {$CDEXTAN},
            PERM #first-line-offset                      {ANY_VALUE},
            PERM #graphic-character-sets                 {$PERMIT-GRCHAR},
            PERM #graphic-character-subrepertoire        {ANY_VALUE},
            PERM #graphic-rendition                      {$GRAPHICRENDITIONS},
            PERM #indentation                            {ANY_VALUE},
            PERM #itemisation                            {ANY_VALUE},
            PERM #kerning-offset                         {ANY_VALUE},
            PERM #line-progression                       {'90-degrees'
                                                          |'270-degrees'},
            PERM #line-spacing                           {ANY_VALUE},
            PERM #line-layout-table                      {ANY_VALUE},
            PERM #proportional-line-spacing              {ANY_VALUE}}}
```

7.7 *Content portion constraints*

7.7.1 *Macro definitions*

No macro definitions are applicable to this clause.

7.7.2 *Factor constraints*

7.7.2.1 *Factor: ANY-CONTENT* {

```
            PERM Content-identifier-logical              {ANY_VALUE},
            PERM Content-identifier-layout               {ANY_VALUE}}
```

7.7.3 *Content portion 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    {0 1 2 4}
                         |#SLS    {ANY_VALUE}
                         |#SCS    {ANY_VALUE}
                         |#SRS    {ANY_VALUE}
                         |#JFY    {0}
                         |#CR
                         |#LF
                         |#PLD
                         |#PLU
```

```
                    |#SP
                    |#SUB
                    |#BPH
                    |#NBH
                    |#SOS
                    |#ST
                    |#$LS0
                    |#$LS1R
                    |#$LS2R
                    |#$LS3R
                    |#$SS2
                    |#$SS3
                    |#$DEG-CORE-G0
                    |#$DEG-646-G0
                    |#$DEG-ANY-G1
                    |#$DEG-ANY-G2
                    |#$DEG-ANY-G3
                     #$DEG-EMPTY-G1
                      }...}}
```

7.7.3.2 *Raster-graphics-content-portion: ANY-CONTENT* {

```
PERM  Number-of-lines              {>0},
REQ   Number-of-pels-per-line      {>=0},
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  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_VALUE},
PERM  Content-information          {GEOMETRIC}}
```

8 Interchange format

8.1 *Document interchange format*

Interchange format class 'A' is to be used in this profile, as defined in [CCITT Recommendation T.415/ISO 8613-5].

8.2 *Octet string lengths*

The maximum length of data values of any universal type (as defined in [CCITT Recommendation X.208/ISO 8824] in data streams which may be encoded in accordance with this profile is 32767 octets. If it is required to encode a data value of string greater length than this, constructed type encoding must be used.

8.3 *Encoding of application comments*

The encoding of the attribute “Application comments” is defined as an octet string in [CCITT Recommendation T.415/ISO 8613-5]. This 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
DEFINITION                    ::= BEGIN
EXPORTS    Object-Class-Appl-Comm-Encoding,
           Object-Appl-Comm-Encoding;

 -- the next two definitions are not ambiguous because they are not used  in the same context --

Object-Class-Appl-Comm-Encoding    ::= SEQUENCE {
      Constraint-name    [0]    IMPLICIT PrintableString,
      External-data      [1]    IMPLICIT OCTETSTRING OPTIONAL }

Object-Appl-Comm-Encoding          ::= SEQUENCE {
      Constraint-name    [0]    IMPLICIT PrintableString OPTIONAL,
      External-data      [1]    IMPLICIT OCTETSTRING OPTIONAL }
END
```

Note – The encoding of the attribute "application comments" is simple (not structured).

ANNEX A

(to Recommendation T.505)

Font References
(informative)

The specification of font references and the font attribute set referred to in this annex are under study and thus this annex is informative at present.

A.1 *Font reference*

The method for specifying a font reference is to be based on ISO 9541. Such a reference is to be specified by the following ASN.1 encoding:

```
Font-reference      ::=  SET {
 user-visible-name                [0]   IMPLICIT Comment-string OPTIONAL,
 user-readable-comment            [1]   IMPLICIT Comment-string OPTIONAL,
 reference-attributes             [2]   IMPLICIT SEQUENCE OF SET {
   precedence-number                    [0]   IMPLICIT INTEGER OPTIONAL,
   attributes                           [1]   IMPLICIT Font-attribute-set,
   user-readable-comment                [2]   IMPLICIT Comment-string OPTIONAL}
```

Font sizes from 6 to 72 points (100 to 1200 BMU) are intended to be supported by implementations conforming to this informative annex. All other values font sizes may additionally be supported, but implementations may also support using some form of "fallback".

The minimum font properties and values from ISO 9541 that are to be specified in a Font-attribute-set are those specified below using the the document application profile notation.

```
Font-attribute-set  {
        PERM  Fontname                          {ANY_VALUE},
        PERM  Standardversion                   {-- to be supplied --},
        PERM  Dsnsource                         {ANY_VALUE},
        PERM  Fontfamily                        {ANY_VALUE},
        PERM  Posture                           {'upright'|'italic-forward'},
        PERM  Weight                            {'light'|'medium'|'bold'},
        PERM  Propwidth                         {ANY_VALUE},
        PERM  Glyphcomp                         {
          PERM #incgyphols                      {ANY_VALUE},
          PERM #exclglyphols                    {ANY_VALUE},
          PERM #inclgyphs                       {ANY_VALUE},
          PERM #exclgyphs                       {ANY_VALUE}),
        PERM  Dsnsize                           {ANY_VALUE},
        PERM  Minsize                           {
          PERM #numerator                       {100..200},
          PERM #denominator                     {1}},
        PERM  Maxsize                           {
          PERM #numerator                       {100..200},
          PERM #denominator                     {1}},
              -- BMUs equivalent to a range of 6..72 point sizes --
        PERM  Dsngroup                          {
          PERM #group-code                      {ANY_VALUE},
          PERM #subgroup-code                   {ANY_VALUE},
          PERM #specifc-group-code              {ANY_VALUE}},
        PERM  Structure                         {ANY_VALUE},
        PERM  Wrmodes                           {
          PERM #wrmodename                      {ANY_VALUE},
          PERM #nomescdir                       {'0-degrees'|'90-degrees'
                                                 |'180-degrees'|'270-degrees'},
          PERM #esclass                         {ANY_VALUE},
          PERM #avlescx                         {ANY_VALUE},
          PERM #avgescy                         {ANY_VALUE}}}
```

ANNEX B

(to Recommendation T.505)

ISO 8632 (CGM) constraints
(informative)

It is intended to transfer the information in this annex to Recommendation T.515 (Implementation Requirements for T.515) which is under study. Hence this annex is for information at present.

It is recommended that geometric graphics content information contain only those elements listed in this annex, in addition to the constraints imposed by ISO 8613-8. It is believed that this subset of CGM is sufficient 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 8632 and ISO 8613-8 concerning mandatory elements and parameters must be fulfilled.

No requirements are placed on how an Interpreter may optionally support features not supported by this profile.

B.1 *Delimiter elements*

Begin Metafile	Metafile name recommended to be the same as file name. Support for string length up to 255.
End Metafile	
Begin Picture	Support for string length up to 255
Begin Picture Body	
End Picture	

B.2 *Metafile descriptor elements*

Metafile Version	Must always be 1
Metafile Description	Support for string length up to 255. String should begin with *ISO FOD26* to identify conformance to this profile.
Real Precision	32 bit floating point (0,9,32) or 32 bit fixed point (1,16,16).
Colour Precision	8 or 16. A MDR is required for a default other than 8.
Colour Index Precision	8 or 16. A MDR is required for a default other than 8.
Maximum Colour Index	0 . . . 255
Colour Value Extent	A 3-tuple in the range [0, 32767]
Metafile Element List	A suitable short hand or list of each element supported by this profile shall be included in the element generated.
Font List	--
Character Set List	Any registered character set. At a minimum, support should be provided for the 94 characters specified in the IRV of ISO 646 (revised 1990).

B.3 *Picture descriptor elements*

VDC Extent	Two points with (x,y) in the range [-32767, 32767]
Background Colour	A 3-tuple in the range [0, 32767]

B.4 *Control elements*

Transparency	- -
Clip Rectangle	Any value within the VDC Range
Clip Indicator	- -

B.5 *Graphical primitive elements*

Polyline	Support for points lists with up to 255 vertices.
Polymarker	Support for points lists with up to 255 vertices.
Text	Support for string lengths up to 255. Only the graphical characters must be supported. No requirements are placed on how an interpreter may support control characters in the string parameter.
Polygon	Support for points lists with up to 255 vertices.
Rectangle	- -
Circle	- -
Circular arc centre	- -
Circular arc centre close	- -
Ellipse	- -
Elliptical arc	- -
Elliptical arc close	- -

B.6 *Attribute elements*

Line Type	1-5
Line Width	- -
Line Colour	- -
Marker Type	1-5
Marker Size	- -
Marker Colour	- -
Text Font Index	- -
Text Colour	- -
Character Height	- -
Character Orientation	- -

Text Alignment	Horizontal: 0, 1, 2, 3; Vertical: 0, 1, 2, 3, 4, 5
Character Set Index	1, 2
Interior Style	0, 1, 3, 4
Fill Colour	- -
Hatch Index	1-6
Colour Table	Minimum colour table support for 64 entries.

B.7 *External entries*

Message	The presentation of the message string may not be appropriate for all applications. No requirement for the formatted presentation of the messsage string has been placed on the interpreter. Only the *No Action* action flag need be supported. Support for string length up to 255.