INTERNATIONAL TELECOMMUNICATION UNION

# CCITT

# T.502

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

## TERMINAL EQUIPMENT AND PROTOCOLS FOR TELEMATIC SERVICES

---

# DOCUMENT APPLICATION PROFILE PM-11 FOR THE INTERCHANGE OF CHARACTER CONTENT DOCUMENTS IN PROCESSABLE AND FORMATTED FORMS

---

## Recommendation T.502

Geneva, 1991

Printed in Switzerland

## 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.502 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 conciseness to indicate both a telecommunication Administration and a recognized private operating agency.

**PAGE BLANCHE**

**RecommendationT.502**

# DOCUMENT APPLICATION PROFILE PM-11 FOR THE INTERCHANGE OF CHARACTER CONTENT DOCUMENTS IN PROCESSABLE AND FORMATTED FORMS

(revised 1990)

## 0 Introduction

This is the definition of an ODA document application profile (DAP) named PM-11. This profile is suitable for interchanging documents in formatted form, processable form or formatted processable form and has been defined in accordance with [ISO 8613-1/CCITT Recommendation T.411]. The format of this profile is in accordance with the standardized proforma and notation defined in the 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 content.

This Recommendation PM-11 is the replacement of Recommendation T.502 (1988). The documents that can be interchanged using this profile range from memos and letters to simple structured documents. 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: Office document interchange format (ODIF) (1989).

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

ISO 8613-1 – 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 architecture.

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-bits 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 graphic 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) (1988).

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

CCITT Recommendation T.505 – Document application profile PM-26 for the interchange of mixed content documents in processable and formatted forms (to be published).

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

CCITT Recommendation T.512 – Implementation requirements for document application profile PM-11 (to be published).

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

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

ISP FOD36 – Office document format profile – FOD36 (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 and RectoPage.

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 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 § 6.6.1). Thus in an application using this profile, the constituents may be known to the user by different names.

## 4 Relationships with other profiles

This profile belongs to a series of hierarchically related profiles which includes PM-26 (see Recommendation T.505) and PM-36 (see Recommendation T.506).

The features supported by this profile are a subset of the features supported by the profile PM-26 and PM-36 and thus all data streams that are conformant to this profile are also conformant to PM-26 and PM-36.

The profile defined in this Recommendation is identical to the profile defined in FOD11. Similarly the profiles PM-26 and PM-36 are identical to the profiles defined in FOD26 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, shall be structured in accordance with one of the document 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.512.

## 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 section 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 form:

– 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.

The constituents that may make up these three forms of document are defined in §§ 6.1.2, 6.1.3 and 6.1.4. 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.

6.1.2 *Formatted form documents*

a) *Required constituents*:

– a document profile;

– layout object descriptions representing a specific layout structure;

– content portion descriptions representing a specific layout structure.

b) *Optional constituents*:

– layout object class descriptions representing a "factor" generic layout structure;

– presentation styles.

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;

– content portion 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 representing a generic logical structure.

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.

Note that when the generic layout structure is present, layout style for the constituent constraint of the type Passage is required.

6.1.4 *Formatted 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;

– layout object class descriptions representing a "complete" generic layout structure;

– layout object descriptions representing a specific layout structure;

– content portion descriptions representing a specific logical/layout structure;

– layout styles.

b) *Optional constituents*:

– presentation styles;

– content portion descriptions representing a generic logical structure.

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

#### 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 and index.

The immediate subordinates of a Passage consist of a sequence of one or more constituent constraints of the type BodyText.

A document may contain only one class definition of the type Passage, which defines the common characteristics of sets of Passages within the document such as layout properties. For example, when the generic layout structure is present, Passage must be entirely laid out in the pages of one page set.

#### 6.2.3.3 *BodyText*

BodyText is a constituent constraint which represents the lowest level of logical subdivision of a document. This constituent constraint is a subdivision of Passages. This allows the layout and presentation characteristics of different parts of the document to be specified.

This is a basic logical constituent that directly refers to content portions that contain character content. BodyText in the specific logical structure must refer to one or more content portions each containing processable or formatted processable character content. However, this constituent in the generic logical structure may not refer to generic content.

### 6.2.4 *Common 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 character 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.

When the generic layout structure is present, constituents of the type CommonContent and their associated constituent constraints 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 common character content that is to be laid out in the header and footer areas of a document. For example, header and 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 must refer one content portion containing processable or 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 and footer areas of a document. This constituent is specifically used when it is required to present a header or footer 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 at the document root or page set level.

The format of the content generators is defined in § 6.6.3.

## 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 be 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.

Page layout type supported by this profile 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, the header area and footer area. Portrait and landscape orientations of this page layout are illustrated in Figures 1/T.502 and 2/T.502 respectively.

### 6.3.2 *DocumentLayoutRoot*

DocumentLayoutRoot is a constituent constraint representing the top level in the document layout structure. Its immediate subordinates consist of a sequence of one or more constituents of the type PageSets. 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 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 which 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 are intended to be laid out alternatively on the "recto" and "verso" (or on the "verso" and "recto") sides of the presentation medium and are represented by the constituent constraints RectoPage and VersoPage respectively. All pages in this sequence 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 of a section within a document. It may be used, for example, for a title page whose layout requirements differ from the following pages.

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

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

ii) all the 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 positions 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.502. 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 nominal page size defined in Table 1/T.502, subject to the restrictions specified above, may be specified as the default value in the document profile.

Table 1/T.502 also includes the recommended assured reproduction area (ARA). Information loss may occur when a document is reproduced if the dimensions of constituent constraint of the type of page exceed the ARA for the specified nominal page size.

TABLE 1/T.502

**Nominal page sizes**

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

6.3.4.4 *Page offset*

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

6.3.4.5 *Page layout characteristics*

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.

For page layout type supported by this profile, the header and footer areas are placed above and below the body area respectively. The layout path in the header, body and footer area are implicitly specified as 270 degrees, as shown in Figures 1/T.502 and 2/T.502. Layout path supported by this profile is 270 degrees only which is the standard default value specified in [CCITT Recommendation T.412/ISO 8613-2]. Thus the layout path may not be specified in a document.

FIGURE 1/T.502 = 11,5 cm

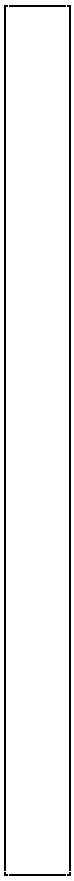
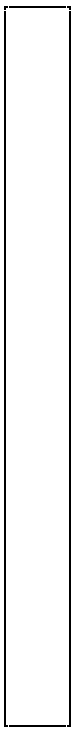


FIGURE 2/T.502 = 9,5 cm



### 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. This body area is represented by a BasicBody frame.

#### 6.3.5.2 *BasicBody*

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

The position and dimensions of this frame are fixed. The layout path of BasicBody is implicitly specified as 270° (see § 6.3.4.5).

### 6.3.6 *Header and footer area characteristics*

#### 6.3.6.1 *General characteristics*

The header and footer areas may consist of basic 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.

The content allocated to these areas is derived from the common part of the logical structure of a document.

#### 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 layout path of these frames is implicitly specified as 270° (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.

### 6.3.7 *SpecificBlock*

SpecificBlock is a constituent constraint that defines a specific block.

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.

Each SpecificBlock in a BasicBody frame must refer to only one content portion. A SpecificBlock in a BasicHeader frame or BasicFooter frame must refer to one or more content portions.

## 6.4 *Document layout characteristics*

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

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

### 6.4.1 *Flow controls*

Various mechanisms are provided to control the allocation of constituent constraints representing the "body" parts of the logical structure of a document to page 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*

In this profile, the following method of allocating the constituent constraint associated with the "body" part of the document to page sets is provided.

Layout object class is used to specify that a particular logical constituent constraint in a document is to be laid out entirely within a specified page set. This is specified for a constituent constraint of the type Passage using the attribute "layout object class" which specifies the object identifier of the required page set.

6.4.1.2 *Allocation of content to page*

In this profile, the following method of allocating the constituent constraint associated with the "body" part of the document to pages is provided.

6.4.1.2.1 *New layout object*

New layout object provides the ability to specify that a particular logical constituent constraint in a document is 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 the logical constituent constraint of the type BodyText.

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.

6.4.1.2.2 *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. It may be specified for logical constituent constraint of the types Passage and BodyText. The attribute "indivisibility" is used to specify this feature.

6.4.1.2.3 *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 the previous basic logical constituent constraint are to be regarded as an unbroken stream of content within a page. This may be specified for basic logical constituent constraint of the type BodyText.

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.

6.4.1.3 *Allocation of content to body areas*

In this profile, the page to which the content is allocated contains a basic body area which is represented by a constituent constraint of the type BasicBody (see § 6.3.5.2). The content is laid out in sequential order in that body area in the form of a single column.

6.4.1.4 *Allocation of content to header-footer areas*

In this profile, a basic header or footer area may be used (see § 6.3.6), the frame representing that area specifies the attribute "logical source" which indicates the particular instance of the logical constituent constraint of the type CommonContent (see§ 6.2.4.1) 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.

6.4.1.4.1 *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 basic logical constituent constraints of the type CommonText and PageNumber. The attribute "concatenation" is used to specify this feature.

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*

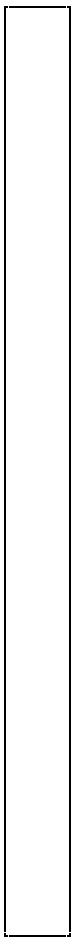
The margins are the minimum distances or offsets between a part of the document content and the edge of the particular area in which that content is laid out. The margins define the maximum extents of the available area into which the content can be positioned.

Margins may be specified for the basic logical constituent constraints of the type BodyText, CommonText and PageNumber; different margin values may be specified for different basic logical constituent constraints without restriction.

Four margins may be independently specified for each logical constituent constraint (see Figure 3/T.502), namely:

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

FIGURE 3/T.502 = 12,5 cm



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

6.4.2.2 *Separation*

Separation is the minimum distance between one basic logical constituent constraint and the next when they are laid out. It may be specified for basic logical constituent constraints of the types BodyText, CommonText and PageNumber. 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.

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

In processable form document 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 as defined in § 6.4.1.1.

b) It is possible to specify page breaks as defined in § 6.4.1.2.1 but is is only possible to indicate that the layout should begin on a new page. It is not possible to specify a particular page class.

   Indivisibility as defined in § 6.4.1.2.2 and same layout object as defined in § 6.4.1.2.3 can all be specified.

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 § 6.6.1). 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 two).

   Concatenation as defined in § 6.4.1.4.1 can all be specified.

d) Margins and separation as defined in § 6.4.3 can all be specified.

6.5 *Content layout and imaging characteristics*

A document may contain character content.

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

6.5.1 *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 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;

– graphic rendition, including the parameters:

  default rendition, bold, italicized, underlined, crossed out, normal intensity, not italicized, not underlined, not crossed out;

– tabulation;

– indentation;

– alignment;

– first line format;

– itemization;

– widow size;

– orphan size;

– initial point.

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

6.5.2 *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 or formatted processable character content.

6.5.3 *Character repertoire*

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 § 6.5.4.

Using these code extension techniques, the graphic character sets designated and/or invoked at the beginning of a content portion containing character content are specified by the presentation attribute "graphic character sets". The graphic character sets may 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 6397 (under revision) respectively.

6.5.4 *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 Figures 4/T.502 and 5/T.502.

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 may be specified by the presentation attribute "code extension announcers". The default code extension announcers used throughout a document may be specified in the document profile using the presentation attribute "code extension announcers".

*Note* – In accordance with [CCITT Recommendation T.416/ISO 8613-6], there is no restriction concerning the number of graphic character sets which are 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 the previous designation to that set, and invocation to GL or GR overrides the previous invocation to GL or GR respectively. Thus the sequential order of designation and/or invocation sequences in the attribute "graphic character sets" is significant.

FIGURE 4/T.502 = 11,5 cm

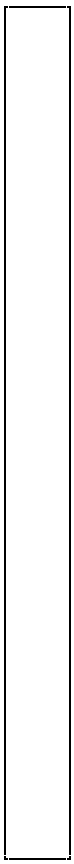
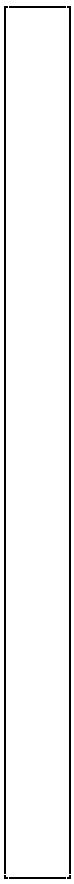


FIGURE 5.T.502 = 12,5 cm



6.5.5 *Line spacing*

Line spacing values of 100, 150, 200, 300 and 400 BMUs may be specified. The values of 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 function SVS.

6.5.6 *Character spacing*

Character spacing values of 80, 100, 120, 160 and 200 BMUs may be specified. The value of 120 BMUs is 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 presentation attribute "character spacing". The value may be changed anywhere within the content portion using the control functions SHS and SCS.

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

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

6.5.7 *Character path and line progression*

Both from left to right and from right to left horizontal writing directions may be specified within a document. The line progression is then from top to bottom only on the page.

Character path values of 0 and 180 degrees may be specified. The value of 0 degree is basic. The value of 180 degrees is non-basic and must be indicated in the document profile.

The values of character path may be specified at the beginning of the content associated with a basic component using the presentation attribute "character path". The value cannot be changed within a content portion.

The line progression supported by this profile is 270 degrees only which is the standard default value specified in [CCITT Recommendation T.416/ISO 8613-6]. Thus the line progression may not be specified.

6.5.8 *Character orientation*

The character orientation supported by this profile is 0 degree only which is the standard default value specified in [CCITT Recommendation T.416/ISO 8613-6]. Thus the character orientation may not be specified.

6.5.9 *Emphasis*

The following modes of emphasizing graphic characters may be specified as basic:

– normal rendition;

– normal intensity;

– increase intensity (bold);

– italicized;

– not italicized;

– underlined;

– not underlined.

The following modes of emphasizing graphic characters may be specified as non-basic:

– crossed-out,

– not crossed-out.

Above non-basic modes must be indicated in the document profile. 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.

### 6.5.10 *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.

### 6.5.11 *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 indentation specified determines the line home position (as defined in [CCITT Recommendation T.416/ISO 8613-6]).

Indentation acts a 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.

6.5.12 *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.

6.5.13 *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 defined in § 6.5.11). 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 feature provided corresponds 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.

6.5.14 *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 to the next frame or page.

Any value of widow or orphan size may be specified using the presentation attribute "widow size" and "orphan size" respectively.

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

6.5.15 *Reverse character string*

Bidirectional writing is supported by this profile (see § 6.5.7). 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.

6.5.16 *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].

6.5.17 *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.

6.5.18 *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.

6.5.19 *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.

6.5.20 *Use of control functions*

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

SHS – select horizontal spacing
(allowed parameter values: 0, 1, 2, 3)

SCS – set character spacing
(allowed parameter values: 80, 100, 120, 160, 200 BMUs)

SVS – select line spacing
(allowed parameter values: 0, 1, 2, 3, 4)

SGR – set graphic rendition
(allowed parameter values: 0, 1, 3, 4, 9, 22-24, 29)

STAB – selective tabulation
(allowed parameter values: any)

SRS – start reverse string
(allowed parameter values: any)

PLD – partial line down

PLU – partial line up

BPH – break permitted here

NBH – no break here

JFY – no justified

SUB – substitute character

SP – space

CR – carriage return

LF – line feed

SOS – start of string

ST – string terminator

– code extension control functions (see § 6.5.4)

The use of all these control functions, with the exception of SP, CR, LF, SOS and ST, are described in §§ 6.5.3 to 6.5.19.

### 6.5.21 *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.

## 6.6 *Miscellaneous features*

### 6.6.1 *Application comments*

Specification of the attributes "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.502.

The second field is optional and may contain any information that is relevant to the application or users. 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.502

**List of number string identifiers**

| Logical constituent | Numeric string identifier |
|---|---|
| DocumentLogicalRoot | 0 |
| Passage | 1 |
| BodyText | 14 |
| CommonContent | 19 |
| CommonText | 20 |
| PageNumber | 40 |
| **Layout constituent** | **Numeric string identifier** |
| DocumentLayoutRoot | 0 |
| PageSet | 1 |
| Page | 2 |
| RectoPage | 3 |
| VersoPage | 4 |
| BasicHeader | 27 |
| BasicBody | 28 |
| SpecificBlock | 30 |
| BasicFooter | 33 |

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

### 6.6.2 *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.

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 set cannot be changed in the attributes "alternative representation".

### 6.6.3 *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 § 7.3.1).

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 INITIALIZEPGNUM in § 7.4.1) and automatically incremented on each successive page (see the macro PAGENUMBER in § 7.4.1).

The content associated with logical object classes of the type PageNumber is laid out in a frame of the following types: BasicHeader or BasicFooter (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 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 numerical (lower case) for the number "4".

### 6.6.4 *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 "comment 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 function is allowed.

6.6.5 *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 "comment 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 function is 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.

#### 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. It is mandatory to specify this information 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 class used in the document (see§ 6.5.2). 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 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/[ISO 8613].

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

– document architecture attributes,

– character content 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:

– profile character sets;

– comment character sets;

– alternative representation character sets;

– page dimensions;

– medium-type;

– character presentation features.

Further information concerning document profile, comment profile 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 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 further specific character 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 § 6.6.4) and “user visible name” (see § 6.6.5) 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 attribute “user readable comments” and “user visible name”.

If this attribute is specified, the code extension techniques which may be used in the “user readable comments” and “user visible name” should 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 restriction on the use of code extension techniques as defined in § 6.5.4 is also applied.

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

There are no restrictions concerning the numbers of graphic character sets which are designated and/or invoked in the “comment character sets”; hence, designation to the same G set overrides the previous G set and invocation to the same GL or GR overides the previous GL or GR.

If the attribute “comment 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 *Document management attributes*

Document management attributes contain information about the content or 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 function is allowed; hence 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 definitions of the constituent constraints which can be represented by data streams conforming to this profile.

### 7.1 *Introduction*

The structure diagram illustrating the relationships between the constituents in the logical structures are shown in Figures 6/T.502 and 7/T.502. The macros indicated on these diagrams are defined in § 7.3.1. These macros define the permissible values for the "generator for subordinates" that are applicable to the constituents and, in effect, define the allowed structures that are supported by this profile.

The structure diagram illustrating the layout structures are shown in Figures 8/T.502 and 9/T.502. The macros indicated on these diagrams are defined in § 7.4.1.

FIGURE 6/T.502 = 11,5 cm

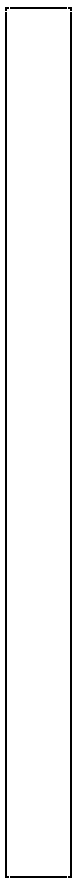


FIGURE 7/T.502 = 8,5 cm

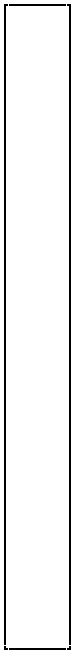


FIGURE 8/T.502 = 10,5 cm

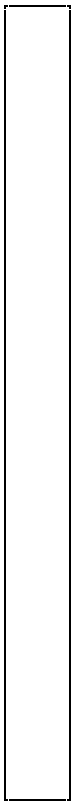
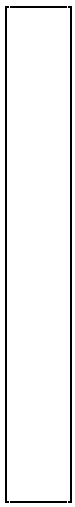


FIGURE 9/T.502 = 6,5 cm



## 7.2 *Document profile constraints*

### 7.2.1 *Macro definitions*

```
DEFINE(FC,          "ASN.1{28260}"          -- formatted character content --)
DEFINE(PC,          "ASN.1{28261}"          -- processable character content --)
DEFINE(FPC,         "ASN.1{28262}"          -- formatted processable character content --)

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

DEFINE (DAC,                "DocumentProfile (Document-architecture-class)")

DEFINE(GLAS,                "DocumentProfile(Generic-layout-structure)")

DEFINE(COMPLETE,            "{'complete-generator-set'} ")

DEFINE(BasicPageDimensions,  "

        {REQ #horizontal-dimension
              {REQ #fixed-dimension {<=9240}},
         REQ #vertical-dimension
              {REQ #fixed-dimension {<=12400}}}
       | {REQ #horizontal-dimension
              {REQ #fixed-dimension {<=12400}},
         REQ #vertical-dimension
              {REQ #fixed-dimension {<=9240}}}
              ")
```

*-- Any size equal to or smaller than CARA (Common Assured Reproduction Area) of ISO A4 and --*
*-- NAL. Both Landscape and Portrait may be specified. --*

```
DEFINE(NonBasicPageDimensions,          "

        {REQ #horizontal-dimension
              {REQ #fixed-dimension {9241..14030}},
         REQ #vertical-dimension
              {REQ #fixed-dimension {12401..19840}} }
       | {REQ #horizontal-dimension
              {REQ #fixed-dimension {12401..19840}},
         REQ #vertical-dimension
              {REQ #fixed-dimension {9241..14030}} }
       | {REQ #horizontal-dimension
              {REQ #fixed-dimension {9241..13200}},
         REQ #vertical-dimension
              {REQ #fixed-dimension {12401..20400}} }
       | {REQ #horizontal-dimension
              {REQ #fixed-dimension {12401..20400}},
         REQ #vertical-dimension
              {REQ #fixed-dimension {9241..13200}} }

              ")
```

*-- Any size equal to or smaller than the full size of ISO A3 or ANSI-B and larger than the range --*
*-- of basic values. Both Landscape and Portrait may be specified. --*

```
DEFINE(PermissiblePageDimensions, "$BasicPageDimensions
        $NonBasicPageDimensions")

DEFINE(NonBasicNominalPageSize,

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

        ")

DEFINE(GRAPHICRENDITIONS, "

    {'cancel' | 'increased-intensity' | 'italicised' | 'underlined' | 'crossed-out'
    | 'normal-intensity' | 'not-italicised' | 'not-underlined' | 'not-crossed-out'} ...
        ")

    -- Permissible values of graphic renditions --

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

DEFINE(CDEXTAN,

    "ESC 02/00 05/00,                -- Use G0 & LS0 --
    [ESC 02/00 05/03],               -- Use G1 & LS1R --
    [ESC 02/20 05/05],               -- Use G2 & LS2R --
    [ESC 02/00 05/07],               -- Use G3 & LS3P --
    [ESC 02/00 05/10],               -- Use G2 & SS2 --
    [ESC 02/00 05/11],               -- Use G3 & SS3 --

        ")
```

-- *Macro defining code extension announcer for DAP defaults* --

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

-- *Same constraint as CDEXTAN* --

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

**DEFINE(FCORE, "04/02"**

-- *The 94 characters of the IRV of ISO 646 (revised 1990) plus the character space (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 multti 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 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 -> GL --

    DEFINE(LS1R,      "ESC 07/14")
        -- Locking shift invoking G1 -> GR --

    DEFINE(LS2R,      "ESC 07/13")
        -- Locking shift invoking G2 -> GR --

    DEFINE(LS3R,      "ESC 07/12")
        -- Locking shift invoking G3 -> GR --

    DEFINE(SS2,       "08/14")
        -- Single shift invoking G2 -> GL --

    DEFINE(SS3,       "08/15")
        -- Single shift invoking G3 -> GL --

-- Macro defining permissible graphic character sets. This macro may be used in each constituent --
-- constraint or presentation style constraint. --

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

-- Macro defining graphic character sets for DAP defaults --

    DEFINE(DAP-DEFAULT-GRCHAR,          "$PERMIT-GRCHAR")
        -- Same constraint as PERMIT-GRCHAR --

-- Macro defining basic graphic 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 graphic character sets --

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

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

    DEFINE(PROFCHAR,
        "{$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 comment character sets --

DEFINE(COMCHAR,
        "{ESC 02/00 05/00          -- Use G0 & LS0 --
         [ESC 02/00 05/03],        -- Use G1 & LS1R --
         [ESC 02/00 05/05],        -- Use G2 & LS2R --
         [ESC 02/00 05/07],        -- Use G3 & LS3R --
         [ESC 02/00 05/10],        -- Use G2 & SS2 --
         [ESC 02/00 05/11]   },    -- Use G3 & SS3 --

         {$DEG-CORE-G0 [$LS0] | $DEG-646-G0 [$LS0]},

        {{$DEG-ANY-G1 [$LS1R]
         | $DEG-ANY-G2 [$LS1R]
         | $DEG-ANY-G3 [$LS1R]} ...
         | $DEG-EMPTY-G1 $LS1R}
                ")

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

DEFINE(ALTCHAR, "$PROFCHAR")
        -- Same constraint as PROFCHAR --
```

7.2.2 *Constituent constraints*

7.2.2.1 *DocumentProfile* {

```
CASE $DAC OF {

        $FDA: PERM      Generic-layout-structure        {'factor-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'}

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

-- Document characteristics --

        REQ   Document-application-profile
              {-- to be supplied --},

        PERM Document-application-profile-defaults {
              CASE $DAC OF {
                    {$FDA}:  {PERM   #content-architecture-classe    {$FC | $FPC}
                    {$PDA}:  {PERM   #content-architecture-classe    {$PC | $FPC | $FC}
                    {$FPDA}: {PERM   #content-architecture-classe    {$FPC | $FC}
                    },

              PERM  #dimensions       {$PermissiblePageDimensions},
                    -- Any ARA (Assured Reproduction Area) of permissible values --

              PERM  #medium-type      {PERM #nominal-page-size
                                            {$NonBasicNominalPageSize},
                                     PERM #side-of-sheet{ANY_VALUE}
                    },
                    -- Any permitted medium type. Both landscape and portrait may be specified --
```

```
PERM  #character-content-defaults {
    PERM        #alignment                          {ANY_VALUE},
    PERM        #character-path                     {'0-degrees' | '180-degrees'},
    PERM        #character-spacing                  {80 | 100 | 120 | 160 | 200},
    PERM        #code-extension-announcers          {$DAP-DEFAULT-CDEXTAN},
    PERM        #first-line-offset                  {ANY_VALUE},
    PERM        #graphic-character-sets             {$DAP-DEFAULT-GRCHAR},
    PERM        #graphic-character-subrepertoire    {ANY VALUE}
    PERM        #graphic rendition                  {$GRAPHICRENDITIONS},
    PERM        #itemisation                        {ANY_VALUE},
    PERM        #line-layout-table                  {ANY_VALUE},
    PERM        #line-spacing                       {100 | 150 | 200 | 300 | 400},
    PERM        #initial-offset                     {ANY_VALUE},
    PERM        #indentation                        {ANY_VALUE},
    PERM        #orphan-size                        {ANY_VALUE},
    PERM        #widow-size                         {ANY_VALUE} }
        },

REQ   Document-architecture-class       {$FDA | $PDA | $FPDA },
REQ   Content-architecture-classes      {[$FC], [$PC], [$FPC]},
REQ   Interchange-format-class          {'if-a'},
REQ   Oda-version                       {REQ #standard-or-recommendation{"ISO 8613"},
                                         PERM #publication-date{"-- to be supplied --}},
```

*-- Non-basic document characteristics --*

```
PERM        Profile-character-sets          {$PROFCHAR},
PERM        Comments-character-sets         {$COMCHAR},
PERM        Alternative-representation
                        -character-sets     {$ALTCHAR},
PERM        Page-dimensions                 {PMUL {$NonBasicPageDimensions}},
PERM        Medium-types                    {PMUL {PERM #nominal-page-size
                                                   {$NonBasicNominalPageSize},
                                                   PERM #side-of-sheet{'recto' | 'verso'}}
        },

PERM Presentation-features      {
    PERM    #character-presentation-features {
        PERM    #character-path                 {'180-degrees'},
        PMUL    #graphic-character-sets         {$NON-BASIC-GRCHAR},
        PMUL    #graphic-character-subrepertoire    {ANY_VALUE},
        PMUL    #character-spacing              {80 | 100 | 160 | 200},
        PMUL    #line-spacing                   {100 | 150},
        PMUL    #graphic-rendition              {'crossed-out'
                                                | 'not-crossed-out'} }
        },
```

*-- Document management attributes --*

*-- Document description --*

```
PERM        Title                   {ANY_VALUE},
PERM        Subject                 {ANY_VALUE},
PERM        Document-type           {ANY_VALUE},
PERM        Abstract                {ANY_VALUE},
PERM        Keywords                {ANY_VALUE},
REQ         Document-reference      {ANY_VALUE},
```

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

-- Originators --
    PERM    Organizations               {ANY_VALUE},
    PERM    Preparers                   {ANY_VALUE},
    PERM    Owners                      {ANY_VALUE},
    PERM    Authors                     {ANY_VALUE},

-- Other user information --
    PERM    Copyright                   {ANY_VALUE},
    PERM    Status                      {ANY_VALUE},
    PERM    User-specific-codes         {ANY_VALUE},
    PERM    Distribution-list           {ANY_VALUE},
    PERM    Additional-information      {ANY_VALUE},

-- External references --
    PERM    Reference-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_VALUE},
    PERM    Number-of-pages             {ANY_VALUE},
    PERM    Languages                   {ANY_VALUE},

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

## 7.3 *Logical constituent constraints*

### 7.3.1 *Macro definitions*

```
DEFINE(DocumentLogicalRootGFS, "
    <construction-expr>::=          REP     OBJECT_CLASS_ID_OF(Passage);
            ")

DEFINE(PassageGFS,"
    <construction-expr>::=          REP     OBJECT_CLASS_ID_OF(Body Text);
            ")

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

    <construction-factor>::=        OBJECT_CLASS_ID_OF_(PageNumber)
                                    | OBJECT_CLASS_ID_OF_(CommonText);
            ")
```

```
DEFINE(HEADERFOOTERSTRING, "
    <string-expr>::=[ANY_STRING]{<str-exp>}[ANY_STRING];

    <str-exp>       ::=     MAKE-STRING(<num-exp>)
                          | UPPER-ALPHA(<num-exp>)
                          | LOWER-ALPHA(<num-exp>)
                          | UPPER-ROMAN(<num-exp>)
                          | LOWER-ROMAN(<num-exp>);

    <num-exp>     ::= B_REF(SUP(CURR-INST8FRAME,CURR-OBJ)))(""PGnum"");
            ")

DEFINE(DocumentLogicalRoot,
    "REQ                #constraint-name {""0""},
    PERM                #external-data {ANY_VALUE}
            ")

DEFINE(Passage,
    "REQ                #constraint-name {""1""},
    PERM                #external-data {ANY_VALUE}
            ")

DEFINE(BodyText,
    "REQ                #constraint-name {""14""},
    PERM                #external-data {ANY_VALUE}
            ")

DEFINE(CommonContent,
    "REQ                #constraint-name {""19""},
    PERM                #external-data {ANY_VALUE}
            ")

DEFINE(CommonText
    "REQ                #constraint-name {""20""},
    PERM                #external-data {ANY_VALUE}
            ")

DEFINE(PageNumber,
    "REQ                #constraint-name {""40""},
    PERM                #external-data {ANY_VALUE}
            ")
```

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},
    PERM        Object-identifier               {ANY_VALUE},
    SPEPERM     Object-class                    {VIRTUAL}

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

7.3.3 *Constituent constraints*

7.3.3.1 *DocumentLogicalRoot: ANY-LOGICAL* {

```
GENERIC:
    REQ     Object-type                 {'document-logical-root'},
    REQ     Generator-for-subordinates  {$DocumentLogicalRootGFS},
    REQ     Application-comments        {$DocumentLogicalRoot}

SPECIFIC:
    PERM    Object-type                 {'document-logical-root'},
    REQ     Object-class                {OBJECT_CLASS_ID_OF
                                            (DocumentLogicalRoot)},
    REQ     Subordinates                {SUB_ID_OF(Passage)+},
    PERM    Application-comments        {$DocumentLogicalRoot}
        }
```

7.3.3.2 *Passage: ANY-LOGICAL* {

```
GENERIC:
    REQ     Object-type                 {'composite-logical-object'},
    REQ     Generator-for-subordinates  {$PassageGFS},
    REQ     Application-comments        {$Passage}

SPECIFIC:
    PERM    Object-type                 {'composite-logical-object'},
    REQ     Object-class                {OBJECT_CLASS_ID_OF(Passage)},
    REQ     Subordinates                {SUB_ID_OF(BodyText)+},
    CASE $GLAS OF {
        $COMPLETE:
            REQ     Layout-style        {STYLE_ID_OF(L-Style1)}
        VOID:
            PERM    Layout-style        {STYLE_ID_OF(L-Style1)}},
    PERM    Application-comments        {$Passage}
        }
```

7.3.3.3 *Bodytext: ANY-LOGICAL* {

```
GENERIC:
    REQ     Object-type                 {'basic-logical-object'}
    REQ     Application-comments        {$BodyText},

SPECIFIC:
    PERM    Object-type                 {'basic-logical-object'},
    REQ     Object-class                {OBJECT_CLASS_ID_OF(BodyText)},
    REQ     Content-portions            {CONTENT_ID_OF
                                            (Character-content-portion)+},
    PERM    Presentation-style          {STYLE_ID_OF(P-Style1)},
    PERM    Content-architecture-classe {$PC | $FPC | $FC},
    PERM    Layout-style                {STYLE_ID_OF(L-Style2)},
    PERM    Application-comments        {$BodyText}
        }
```

7.3.3.4 *CommonContent* {

```
GENERIC:
    REQ     Object-type                 {'composite-logical-object'},
    REQ     Object-class-identifier     {ANY_VALUE},
    REQ     Generator-for-subordinates  {$CommonContentGFS},
    REQ     Application-comments        {$CommonContent},
    PERM    User-readable-comments      {ANY_VALUE},
    PERM    User-visible-name           {ANY_VALUE}
        }
```

7.3.3.5 *CommonText* {

```
GENERIC:
    REQ     Object-type                     {'basic-logical-object'},
    REQ     Object-class-identifier         {ANY-VALUE},
    REQ     Content-portions                {CONTENT_ID_OF
                                                (Character-content-portion)},
    PERM    Presentation-style              {STYLE_ID_OF(P-Style2)},
    PERM    Content-architecture-class      {$PC | $FPC | $FC},
    PERM    Layout-style                    {STYLE_ID_OF(L-Style3)},
    PERM    Application-comments            {$CommonText},
    PERM    User-readable-comments          {ANY_VALUE},
    PERM    User-visible-name               {ANY_VALUE}
        }
```

7.3.3.6 *PageNumber* {

```
GENERIC:
    REQ     Object-type                     {'basic-logical-object'},
    REQ     Object-class-identifier         {ANY-VALUE},
    REQ     Content-generator               {$HEADERFOOTERSTRING},
    PERM    Presentation-style              {STYLE_ID_OF(P-Style2)},
    PERM    Content-architecture-classe     {$PC | $FPC | $FC},
    PERM    Layout-style                    {STYLE_ID_OF(L-Style3)},
    REQ     Application-comments            {$PageNumber},
    PERM    User-readable-comments          {ANY_VALUE},
    PERM    User-visible-name               {ANY_VALUE}
        }
```

7.4 *Layout constituent constraint*

7.4.1 *Macro definitions*

```
DEFINE(DocumentLayoutRootGFS, "
    <construction-expr> ::=     REP  CHO({OBJECT_CLASS_ID_OF(PageSet)}...);
        ")

DEFINE(PageSetGFS , "<construction-expr>  ::= <constraint-1>
                                            | <constraint-2>
                                            | <constraint-3>
                                            | <constraint-4>
                                            | <constraint-5>;

<constraint-1>      ::=     OBJECT_CLASS_ID_OF(Page);

<constraint-2>      ::=     REP OBJECT_CLASS_ID_OF( Page);

<constraint-3>      ::=     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));

<constraint-4>      ::=     SEQ(OBJECT_CLASS_ID_OF(Page)
                                OPT REP OBJECT_CLASS_ID_OF(Page)  );

<constraint-5>      ::=     SEQ(OBJECT_CLASS_ID_OF(Page)
                                OPT REP (SEQ(OBJECT_CLASS_ID_OF(RectoPage)
                                        OPT OBJECT_CLASS_ID_OF(VersoPage)))      )
                            | SEQ(OBJECT_CLASS_ID_OF(Page)
                                OPT REP (SEQ(OBJECT_CLASS_ID_OF(VersoPage)
                                        OPT OBJECT_CLASS_ID_OF(RectoPage)))      );
        ")
```

```
DEFINE(PageGFS,          "
    <construction-expr>  ::=   SEQ([OBJECT_CLASS_ID_OF(BasicHeader)]
                                    OBJECT_CLASS_ID_OF(BasicBody)
                                    [OBJECT_CLASS_ID_OF(BasicFooter) ]);
            ")

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

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

DEFINE(DocumentLayoutRoot,  "REQ      #constraint-name {""0""},
                            PERM     #external-data {ANY_VALUE}
            ")

DEFINE(PageSet,             "REQ      #constraint-name {""1""},
                            PERM     #external-data {ANY_VALUE}
            ")

DEFINE(Page,                "REQ      #constraint-name {""2""},
                            PERM     #external-data {ANY_VALUE}
            ")

DEFINE(RectoPage,           "REQ      #constraint-name {""3""},
                            PERM     #external-data {ANY_VALUE}
            ")

DEFINE(VersoPage,           "REQ      #constraint-name {""4""},
                            PERM     #external-data {ANY_VALUE}
            ")

DEFINE(BasicHeader,         "REQ      #constraint-name {""27""},
                            PERM     #external-data {ANY_VALUE}
            ")

DEFINE(BasicBody,           "REQ      #constraint-name {""28""},
                            PERM     #external-data {ANY_VALUE}
            ")

DEFINE(SpecificBlock        "REQ      #constraint-name {""30""},
                            PERM     #external-data {ANY_VALUE}
            ")

DEFINE(BasicFooter,         "REQ      #constraint-name {""33""},
                            PERM     #external-data {ANY_VALUE}
            ")
```

7.4.2 *Factor constraints*

7.4.2.1 *Factor: ANY-COMPOSITE-LAYOUT {*

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

SPECIFIC:
    PERM         Object-type              {VIRTUAL},
    PERM         Object-identifier        {ANY_VALUE}
```

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

7.4.2.2 *Factor Any-Page: ANY-COMPOSITE-LAYOUT* {

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

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

SPECIFIC_AND_GENERIC:
    PERM        Dimensions                     {$PermissiblePageDimensions},
    PERM        Page-position                  {ANY_VALUE}
          }
```

7.4.2.3 *Factor Any-Frame: ANY-COMPOSITE-LAYOUT* {

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

SPECIFIC:
    PERM        Object-type            {'frame'}
    REQ         Subordinates           {SUB_ID_OF(SpecificBlock)+}

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

7.4.3 *Constituent constraints*

7.4.3.1 *DocumentLayoutRoot: ANY-COMPOSITE-LAYOUT* {

```
GENERIC:
    REQ         Object-type                    {'document-layout-root'},
    REQ         Generator-for-subordinates     {$DocumentLayoutRootGFS},
    CASE $DAC OF {
      $PDA-FPDA:     PERM Bindings             {$INITIALISEPGNUM} }
    REQ         Application-comments           {$DocumentLayoutRoot}

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

7.4.3.2 *PageSet: ANY-COMPOSITE-LAYOUT* {

```
GENERIC:
    REQ          Object-type                    {'pageset'},
    REQ          Generator-for-subordinates     {$PageSetGFS},
    CASE $DAC OF {
      $PDA-FPDA:      PERM Bindings             {$INITIALISEPGNUM} }
    REQ          Application-comments           {$PageSet}

SPECIFIC:
    PERM         Object-type                    {'pageset'},
    CASE $DAC OF {
      $FDA:           PERM Object-class         {OBJECT_CLASS_ID_OF (PageSet)}
      $FPDA:          REQ  Object-class         {OBJECT_CLASS_ID_OF (PageSet)} },
    REQ          Subordinates                   {{SUB_ID_OF (Page)+},
                                                 {SUB_ID_OF (RectoPage)+},
                                                 {SUB_ID_OF (VersoPage)+} },
    PERM         Application-comments           {$PageSet}
          }
```

7.4.3.3 *Page: ANY_PAGE* {

```
GENERIC:
    REQ          Application-comments           {$Page}

SPECIFIC:
    CASE $DAC OF {
      $FDA:           PERM Object-class         {OBJECT_CLASS_ID_OF(Page)}
      $FPDA:          REQ  Object-class         {OBJECT_CLASS_ID_OF(Page)} },
    PERM         Application-comments           {$Page}

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

          }
```

7.4.3.4 *RectoPage: ANY-PAGE* {

```
GENERIC:
    REQ          Application-comments           {$RectoPage},
    REQ          Medium-type                    {REQ #nominal-page-size
                                                      {NonBasicNominalPageSize},
                                                REQ #side-of-sheet{'unspecified' | 'recto'} }

SPECIFIC:
    CASE $DAC OF {
      $FDA:           PERM Object-class         {OBJECT_CLASS_ID_OF (RectoPage)}
      $FPDA:          REQ  Object-class         {OBJECT_CLASS_ID_OF (RectoPage)} },
    PERM         Application-comments           {$RectoPage},
    PERM         Medium-type                    {PERM #nominal-page-size
                                                      {NonBasicNominalPageSize},
                                                PERM #side-of-sheet{'unspecified' | 'recto'}}

          }
```

7.4.3.5 *VersoPage: ANY-PAGE* {

```
GENERIC:
    REQ          Application-comments           {$VersoPage},
    REQ          Medium-type                    {REQ #nominal-page-size
                                                      {NonBasicNominalPageSize},
                                                REQ #side-of-sheet{'unspecified' | 'verso'}}
```

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

7.4.3.6 *BasicHeader: ANY-FRAME* {

```
GENERIC:
    CASE $DAC OF {
      $PDA- FPDA:
                       REQ  Logical-source       {OBJECT_CLASS_ID_OF (CommonContent)},
    REQ         Application-comments            {$BasicHeader}

SPECIFIC:
    CASE $DAC OF {
      $FDA:          PERM Object-class          {OBJECT_CLASS_ID_OF (BasicHeader)}
      $FPDA:         REQ  Object-class          {OBJECT_CLASS_ID_OF (BasicHeader)} },
    PERM        Application-comments            {$BasicHeader}
            }
```

7.4.3.7 *BasicBody: ANY-FRAME* {

```
GENERIC:
    REQ         Application-comments            {$BasicBody}

SPECIFIC:
    CASE $DAC OF {
      $FDA:          PERM Object-class          {OBJECT_CLASS_ID_OF (BasicBody)}
      $FPDA:         REQ  Object-class          {OBJECT_CLASS_ID_OF (BasicBody)} },
    PERM        Application-comments            {$BasicBody}
            }
```

7.4.3.8 *BasicFooter: ANY-FRAME* {

```
GENERIC
    CASE $DAC OF {
      $PDA-FPDA:
                       REQ  Logical-source       {OBJECT_CLASS_ID_OF (CommonContent}}
    REQ         Application-comments            {$BasicFooter}

SPECIFIC:
    CASE $DAC OF {
      $FDA:          PERM Object-class          {OBJECT_CLASS_ID_OF (BasicFooter)}
      $FPDA:         REQ  Object-class          {OBJECT_CLASS_ID_OF (BasicFooter)} },
    PERM        Application-comments            {$BasicFooter}
            }
```

7.4.3.9 *SpecificBlock* {

```
SPECIFIC:
    REQ         Object-type                     {'block'},
    REQ         Object-identifier               {ANY_VALUE},
    REQ         Content-portions                {CONTENT_ID_OF
                                                      (Character-content-portion)+},
    PERM        Presentation-style              {STYLE_ID_OF(P-Style1)
                                                {STYLE_ID_OF(P-Style2)},
    PERM        Content-architecture-class      {$FC | $FPC},
```

```
PERM        Presentation-attributes                 {
    PERM        #character-attributes               {
    PERM        #alignment                                  {ANY_VALUE},
    PERM        #character-path                             {'0-degrees' | '180-degrees'},
    PERM        #character-spacing                          {80 | 100 | 120 | 160 | 200},
    PERM        #code-extension-announcers                  {$CEXTAN},
    PERM        #first-line-offset                          {ANY_VALUE},
    PERM        #graphic-character-sets                     {$PERMIT-GRCHAR},
    PERM        #graphic-character-subrepertoire            {ANY_VALUE},
    PERM        #graphic-rendition                          {$GRAPHICRENDITIONS},
    PERM        #itemisation                                {ANY_VALUE},
    PERM        #line-layout-table                          {ANY_VALUE},
    PERM        #line-spacing                               {100 | 150 | 200 | 300 | 400},
    PERM        #initial-offset                             {ANY_VALUE}             }},
PERM        User-readable-comments  {ANY_VALUE},
PERM        User-visible-name       {ANY_VALUE},
PERM        Position                {REQ #fixed-position
                                        {REQ #horizontal-position{ANY_INTEGER},
                                         REQ #vertical-position{ANY_INTEGER}}},
PERM        Dimensions              {REQ #horizontal-dimension
                                        {REQ #fixed-dimension{ANY_INTEGER}},
                                    REQ #vertical-dimension
                                        {REQ #fixed-dimension{ANY_INTEGER}}},
PERM        Application-comments    {$SpecificBloc}
        }
```

*-- Each block in a BasicBody must refernce only one content portion. --*

*-- A block in a BasicHeader or BasicFooter may reference any number of content portions. --*

## 7.5 *Layout style constraints*

### 7.5.1 *Macro definitions*

No macro definitions are applicable in this clause.

### 7.5.2 *Factor constraints*

#### 7.5.2.1 *Factor: ANY-LAYOUT-STYLE* {

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

### 7.5.3 *Constituent constraints*

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

*-- This style is used for the constituents Passage only. --*

```
CASE $GLAS OF {
    $COMPLETE:
      PERM      Indivisibility          {'page' | 'null'},
      REQ       Layout-object-class     {OBJECT_CLASS_ID_OF (PageSet) }
    VOID:
      PERM      Indivisibility          {'page' | 'null'} }
        }
```

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

-- *This style is used for the constituents BodyText only.* --

```
CASE $GLAS OF {
    $COMPLETE:
      PERM      Indivisibility          {'page' | 'null'},
      PERM      New-layout-object       {{OBJECT_CLASS_ID_OF (Page)
                                        | OBJECT_CLASS_ID_OF (RectoPage)
                                        | OBJECT_CLASS_ID_OF (VersoPage)
                                        | OBJECT_CLASS_ID_OF (BasicBody)}
                                        |'page' | 'null'},
      PERM      Offset                  {ANY_VALUE},
      PERM      Same-layout-object      {REQ #same-as {<object-id-expr>::=
                                              PREC-OBJ(CURR-OBJ); | 'null' },
                                        PERM #within {'page'}},
      PERM      Separation              {PERM #leading-edge  {ANY_INTEGER},
                                         PERM #trailing-edge {ANY_INTEGER}}
    VOID:
      PERM      Indivisibility          {'page' | 'null'},
      PERM      New-layout-object       {'page' | 'null'},
      PERM      Offser                  {ANY_VALUE},
      PERM      Same-layout-object      {REQ #same-as {<object-id-expr>::=
                                              PREC-OBJ(CURR-OBJ); | 'null' },
                                        PERM #within {'page'}},
      PERM      Separation              {PERM #leading-edge {ANY_INTEGER},
                                         PERM #trailing-edge {ANY_INTEGER}} }
          }
```

7.5.3.3 *L-Style3: ANY-LAYOUT-STYLE* {

-- *This style is used for the constituents CommonText and PageNumber.* --

```
PERM      Concatenation     {ANY_VALUE},
PERM      Offset            {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 in this clause.

7.6.2 *Factor constraints*

7.6.2.1 *Factor: ANY-PRESENTATION-STYLE* {

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

7.6.3 *Constituent constraints*

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

-- *This style is used for the constituent BodyText only.* --

```
PERM      Presentation-attibutes {
    PERM  #character-attributes             {
        PERM      #alignment                        {ANY_VALUE},
        PERM      #character-path                   {'0-degrees' | '180-degrees'},
        PERM      #character-spacing                {80 | 100 | 120 | 160 | 200},
```

```
        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    #itemisation                        {ANY_VALUE},
        PERM    #line-layout-table                  {ANY_VALUE},
        PERM    #line-spacing                       {100 | 150 | 200 | 300 | 400},
        PERM    #indentation                        {ANY_VALUE},
        PERM    #orphan-size                        {ANY_VALUE},
        PERM    #widow-size                         {ANY_VALUE} }}
            }
```

7.6.3.2 *P-Style2: ANY-PRESENTATION-STYLE* {

-- *This style is used for the constituent CommonText, PageNumber and SpecificBlock.* --

```
PERM    Presentation-atributes {
    PERM    #character-attributes           {
        PERM    #alignment                          {ANY_VALUE},
        PERM    #character-path                     {'0-degrees' | '180-degrees'},
        PERM    #character-spacing                  {80 | 100 | 120 | 160 | 200},
        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    #itemisation                        {ANY_VALUE},
        PERM    #line-layout-table                  {ANY_VALUE},
        PERM    #line-spacing                       {100 | 150 | 200 | 300 | 400},
        PERM    #indentation                        {ANY_VALUE} }}
            }
```

7.7 *Content portion constraints*

7.7.1 *Macro definitions*

No macro definitions are applicable in this clause.

7.7.2 *Factor constraints*

No factor constraints are applicable in this clause.

7.7.3 *Content portion constraints*

7.7.3.1 *Character-content-portion* {

```
PERM    Content-identifier-layout       {ANY_VALUE},
PERM    Content-identifier-logical      {ANY_VALUE},
PERM    Alternative-representation      {ANY_VALUE},
PERM    Content-information             {CHARACTER,
            {#STAB                          {ANY_VALUE}
            |#SHS                           {80 | 100 | 120 | 200}
            |#SCS                           {80 | 100 | 120 | 160 | 200}
            |#SGR                           {$GRAPHICRENDITIONS}
            |#SVS                           {100 | 150 | 200 | 300 | 400}
            |#SRS                           {ANY_VALUE}
            |#JFY                           {0}
```

```
                |#CR
                |#LF
                |#PLD
                |#PLU
                |#SP
                |#SUB
                |#BPH
                |#NBH
                |#SOS
                |#ST
                |#$LSO
                |#$LS1R
                |#$LS2R
                |#$LS3R
                |#$SS2
                |#$SS3
                |#$DEG-CORE-G0
                |#$DEG-646-G0
                |#$DEG-ANY-G1
                |#$DEG-ANY-G2
                |#$DEG--ANY-G3
                |#$DEG-EMPY-G1}... }
        }
```

## 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 *Data value length*

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 32 767 octets. If it is required to encode a data value string of 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
```