INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.506

(08/93)

TERMINALS FOR TELEMATIC SERVICES

DOCUMENT APPLICATION PROFILE PM-36 FOR THE INTERCHANGE OF EXTENDED DOCUMENT STRUCTURES AND MIXED CONTENT DOCUMENTS IN PROCESSABLE AND FORMATTED FORMS

ITU-T Recommendation T.506

(Previously “CCITT Recommendation”)

FOREWORD

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

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

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

ITU-T Recommendation T.506 was prepared by ITU-T Study Group 8 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 31st of August 1993.

NOTE

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

CONTENTS

Page

1 Scope 1

2 Normative references 1

2.1 Paired Recommendations | International Standards equivalent in technical content 1

2.2 Additional references 2

3 Definitions 3

4 Relationship with other International Standardized Profiles 4

5 Conformance 4

5.1 Data stream conformance 4

5.2 Implementation conformance 4

6 Characteristics supported by this document application profile 4

6.1 Overview 5

6.1.1 General 5

6.1.2 Formatted form documents 5

6.1.3 Processable form documents 5

6.1.4 Formatted processable form documents 5

6.1.5 Generic documents 6

6.1.6 Document profile 6

6.2 Logical characteristics 6

6.2.1 Introduction 6

6.2.2 Overview of the logical structure 6

6.2.3 Body part of the logical structure 7

6.2.3.1 DocumentLogicalRoot 7

6.2.3.2 Passage 7

6.2.3.3 Numbered Segment 8

6.2.3.4 Number.... 8

6.2.3.5 Paragraph. 9

6.2.3.6 BodyText, BodyRaster and BodyGeometric 9

6.2.3.7 Footnote... 10

6.2.3.8 FootnoteReference 10

6.2.3.9 FootnoteBody 10

6.2.3.10 FootnoteNumber 11

6.2.3.11 FootnoteText 11

6.2.3.12 Title 11

6.2.3.13 Phrase 12

6.2.3.14 Constituents that provide for a general referencing mechanism 12

6.2.3.15 Constituents representing illustrations 13

6.2.3.16 Constituents used to represent tables 15

6.2.3.17 Constituents representing lists 16

6.2.4 Common content part of the logical structure 18

6.2.4.1 CommonContent 18

6.2.4.2 CommonText 18

6.2.4.3 PageNumber 18

6.2.4.4 CommonRaster 18

6.2.4.5 CommonGeometric 18

6.2.4.6 CommonReference 19

6.2.4.7 CurrentInstance 19

6.2.4.8 TableNumber 19

6.2.4.9 CommonNumber 19

Page

6.3 Layout characteristics ... 19
6.3.1 Overview of the layout characteristics ... 19
6.3.2 DocumentLayoutRoot ... 20
6.3.3 PageSet ... 20
6.3.4 Page characteristics ... 20
6.3.4.1 Page constituents ... 20
6.3.4.2 Page dimensions ... 21
6.3.4.3 Nominal page sizes ... 21
6.3.4.4 Page offset ... 21
6.3.4.5 Page layout characteristics ... 21
6.3.5 Body area characteristics ... 25
6.3.5.1 General characteristics ... 25
6.3.5.2 BasicBody ... 25
6.3.5.3 VariableCompositeBody ... 26
6.3.5.4 BasicFloat ... 26
6.3.5.5 SnakingColumns ... 29
6.3.5.6 SynchronizedColumns ... 31
6.3.5.7 ColumnVariable ... 32
6.3.5.8 ColumnFixed ... 32
6.3.5.9 FootnoteArea ... 33
6.3.5.10 FixedCompositeBody ... 33
6.3.5.11 CompositeFloat ... 34
6.3.5.12 CompositeFixtureVariable ... 35
6.3.5.13 CompositeFixtureFixed ... 36
6.3.5.14 CompositeArtwork ... 36
6.3.5.15 BasicFixture ... 36
6.3.5.16 FormArea ... 36
6.3.5.17 EntryGroupArea ... 37
6.3.5.18 FormEntryArea ... 37
6.3.5.19 BasicColumn ... 38
6.3.5.20 CompositeColumnVariable ... 38
6.3.5.21 CompositeColumnFixed ... 38
6.3.5.22 CompositeCommon ... 38
6.3.5.23 Constituents used for laying out tables ... 39
6.3.6 Header and footer area characteristics ... 44
6.3.6.1 General characteristics ... 44
6.3.6.2 BasicHeader and BasicFooter ... 44
6.3.6.3 CompositeHeader and CompositeFooter ... 44
6.3.6.4 SourcedContentVariable ... 45
6.3.6.5 ArrangedContentVariable ... 46
6.3.6.6 SourcedContentFixed ... 46
6.3.6.7 ArrangedContentFixed ... 46
6.3.7 GenericBlock and SpecificBlock ... 46

6.4 Document layout characteristics ... 47
6.4.1 Flow controls ... 47
6.4.1.1 Allocation of content to page sets ... 47
6.4.1.2 Page breaks ... 47
6.4.1.3 Allocation of content to body areas ... 48
6.4.1.4 Allocation of content to header and footer areas ... 52
6.4.2 Layout of document content ... 53
6.4.2.1 Margins.... ... 53
6.4.2.2 Separation ... 54
6.4.2.3 Indivisibility ... 54
6.4.2.4 Same layout object ... 54
6.4.2.5 Concatenation ... 54
6.4.2.6 Block alignment ... 54
6.4.3 Layout controls applicable in the absence of a generic layout structure ... 54

Page

6.5 Content layout and imaging characteristics 55
6.5.1 Character content 55
6.5.1.1 Introduction 55
6.5.1.2 Character content architecture class 56
6.5.1.3 Character repertoires 56
6.5.1.4 Code extension techniques 57
6.5.1.5 Line spacing 58
6.5.1.6 Character spacing 58
6.5.1.7 Character path and line progression 59
6.5.1.8 Character positioning controls 59
6.5.1.9 Character orientation 59
6.5.1.10 Emphasis 60
6.5.1.11 Tabulation 60
6.5.1.12 Indentation 61
6.5.1.13 Alignment 61
6.5.1.14 First line format 61
6.5.1.15 Widow and orphan sizes 61
6.5.1.16 Fonts 62
6.5.1.17 Reverse character strings 62
6.5.1.18 Kerning offset 62
6.5.1.19 Proportional line spacing 62
6.5.1.20 Superscripts and subscripts 62
6.5.1.21 Line breaks 62
6.5.1.22 Substitution of characters 63
6.5.1.23 Initial point 63
6.5.1.24 Use of control functions 63
6.5.1.25 Formatting the content 64
6.5.1.26 Graphic character composition 64
6.5.1.27 Parallel text 64
6.5.1.28 Pairwise kerning 64
6.5.2 Raster graphics content 64
6.5.2.1 Introduction 64
6.5.2.2 Raster graphics content architecture 65
6.5.2.3 Raster graphics encoding methods 65
6.5.2.4 Pel path and line progression 65
6.5.2.5 Clipping... 65
6.5.2.6 Pel spacing 66
6.5.2.7 Spacing ratio 66
6.5.2.8 Image dimensions 66
6.5.3 Geometric graphics content 66

6.6 Miscellaneous features 66
6.6.1 Resource documents 66
6.6.2 External documents 67
6.6.3 Border 67
6.6.4 Application comments 67
6.6.5 Alternative representation 67
6.6.6 Automatic numbering and referencing mechanisms 67
6.6.6.1 Introduction 67
6.6.6.2 Bindings 69
6.6.6.3 Numbering of segments 71
6.6.6.4 Numbering of figures 72
6.6.6.5 Numbering of lists 72
6.6.6.6 Numbering of tables 72
6.6.6.7 Footnote numbering 73
6.6.6.8 Page numbering 73
6.6.6.9 Referenced content 74
6.6.6.10 Common references 74
6.6.6.11 Current instance references 75
6.6.6.12 Common number references 75

6.6.7 User readable comments ... 75
6.6.8 User visible name ... 75
6.6.9 Unit scaling ... 75
6.7 Document management features ... 76
6.7.1 Document constituent information ... 76
6.7.1.1 Presence of document constituents ... 76
6.7.1.2 Resource document information ... 76
6.7.1.3 External document information ... 76
6.7.2 Document identification information ... 76
6.7.2.1 Document application profile information ... 76
6.7.2.2 Document architecture class information ... 76
6.7.2.3 Content architecture classes information ... 76
6.7.2.4 Interchange format class information ... 76
6.7.2.5 ODA version information ... 77
6.7.2.6 Document reference ... 77
6.7.3 Document default information ... 77
6.7.4 Non-basic characteristics information ... 77
6.7.4.1 Profile character sets ... 77
6.7.4.2 Comments character sets ... 78
6.7.4.3 Alternative representation character sets ... 78
6.7.5 Fonts list ... 78
6.7.6 Document management attributes ... 78
7 Specification of constituent constraints ... 79
7.1 Introduction ... 79
7.1.1 Diagrams of relationships of logical constituents ... 79
7.1.2 Diagrams of relationships of layout constituents ... 86
7.1.3 Notation ... 92
7.2 Document profile constituent constraints ... 93
7.2.1 Macro definitions ... 93
7.2.2 Constituent constraints ... 98
7.2.2.1 DocumentProfile ... 98
7.3 Logical constituent constraints ... 101
7.3.1 Macro definitions ... 101
7.3.2 Factor constraints ... 115
7.3.2.1 FACTOR ANY-LOGICAL ... 115
7.3.2.2 FACTOR COMP-LOGICAL ... 115
7.3.2.3 FACTOR BASIC-LOGICAL ... 115
7.3.2.4 FACTOR ANY-COMMON ... 116
7.3.3 Constituent constraints ... 116
7.3.3.1 DocumentLogicalRoot ... 116
7.3.3.2 Passage ... 116
7.3.3.3 NumberedSegment ... 117
7.3.3.4 Number ... 117
7.3.3.5 Title ... 117
7.3.3.6 Caption ... 118
7.3.3.7 Paragraph ... 118
7.3.3.8 Phrase ... 118
7.3.3.9 Footnote ... 118
7.3.3.10 FootnoteNumber ... 119
7.3.3.11 FootnoteReference ... 119
7.3.3.12 FootnoteBody ... 119
7.3.3.13 FootnoteText ... 120
7.3.3.14 Figure ... 120
7.3.3.15 BodyText ... 120
7.3.3.16 Reference ... 120
7.3.3.17 ReferencedContent ... 121
7.3.3.18 BodyRaster ... 121
7.3.3.19 BodyGeometric ... 121

Page
7.3.3.20 CommonContent ... 122
7.3.3.21 CommonText ... 122
7.3.3.22 CommonReference ... 122
7.3.3.23 CommonNumber ... 122
7.3.3.24 CurrentInstance ... 122
7.3.3.25 CommonRaster ... 123
7.3.3.26 CommonGeometric ... 123
7.3.3.27 PageNumber ... 123
7.3.3.28 TableNumber ... 123
7.3.3.29 Description ... 123
7.3.3.30 Artwork ... 124
7.3.3.31 NumberedList ... 124
7.3.3.32 UnNumberedList ... 124
7.3.3.33 DefinitionList ... 125
7.3.3.34 ListItem ... 125
7.3.3.35 ListTerm ... 125
7.3.3.36 Table ... 125
7.3.3.37 Row ... 126
7.3.3.38 TableComponent ... 126
7.3.3.39 RowComponent ... 126
7.3.3.40 Form ... 126
7.3.3.41 EntryElement ... 127
7.3.3.42 EntryGroup ... 127
7.3.3.43 EntryText ... 127
7.3.3.44 EntryRaster ... 127
7.3.3.45 EntryGeometric ... 128
7.4 Layout constituent constraints ... 128
7.4.1 Macro definitions ... 128
7.4.2 Factor constraints ... 132
7.4.2.1 FACTOR ANY-LAYOUT ... 132
7.4.2.2 FACTOR ANY-PAGE ... 132
7.4.2.3 FACTOR ANY-FRAME-FIXED ... 132
7.4.2.4 FACTOR ANY-FRAME-VARIABLE ... 133
7.4.3 Constituent constraints ... 133
7.4.3.1 DocumentLayoutRoot ... 133
7.4.3.2 PageSet ... 133
7.4.3.3 Page ... 134
7.4.3.4 RectoPage ... 134
7.4.3.5 VersoPage ... 135
7.4.3.6 CompositeHeader ... 135
7.4.3.7 CompositeFooter ... 135
7.4.3.8 FixedCompositeBody ... 136
7.4.3.9 VariableCompositeBody ... 136
7.4.3.10 ColumnFixed ... 137
7.4.3.11 ColumnVariable ... 138
7.4.3.12 SnakingColumns ... 139
7.4.3.13 SynchronizedColumns ... 139
7.4.3.14 BasicFloat ... 140
7.4.3.15 CompositeFloat ... 141
7.4.3.16 BasicColumn ... 142
7.4.3.17 FootnoteArea ... 143
7.4.3.18 ArrangedContentFixed ... 144
7.4.3.19 ArrangedContentVariable ... 144
7.4.3.20 SourcedContentFixed ... 145
7.4.3.21 SourcedContentVariable ... 145
7.4.3.22 CompositeFixtureVariable ... 147
7.4.3.23 CompositeFixtureFixed ... 148

Page
7.4.3.24 BasicFixture 149
7.4.3.25 CompositeColumnFixed 149
7.4.3.26 CompositeColumnVariable 150
7.4.3.27 CompositeCommon 151
7.4.3.28 CompositeArtwork 152
7.4.3.29 BasicHeader 152
7.4.3.30 BasicFooter 153
7.4.3.31 BasicBody 153
7.4.3.32 GenericBlock 153
7.4.3.33 SpecificBlock 154
7.4.3.34 FormArea 155
7.4.3.35 EntryGroupArea 156
7.4.3.36 TableArea 156
7.4.3.37 TableHeader 157
7.4.3.38 TableLabel 158
7.4.3.39 CompositeTableLabel 158
7.4.3.40 LabelComponent 159
7.4.3.41 RowArea 160
7.4.3.42 Cell 160
7.4.3.43 SubRowGroup 161
7.4.3.44 SubRow 162
7.4.3.45 TableLabelContent 162
7.4.3.46 FormEntryArea 163
7.5 Layout style constituent constraints 163
7.5.1 Macro definitions 163
7.5.2 Factor constraints 163
7.5.3 Constituent constraints 164
7.5.3.1 LStyle1 164
7.5.3.2 LStyle2 164
7.5.3.3 LStyle3 165
7.5.3.4 LStyle4 165
7.5.3.5 LStyle5 165
7.5.3.6 LStyle6 165
7.5.3.7 LStyle7 166
7.5.3.8 LStyle8 166
7.5.3.9 LStyle9 166
7.5.3.10 LStyle10 166
7.5.3.11 LStyle11 166
7.5.3.12 LStyle12 167
7.5.3.13 LStyleT1 167
7.5.3.14 LStyleT2 167
7.5.3.15 LStyleT3 167
7.5.3.16 LStyleT4 168
7.5.3.17 LStyleT5 168
7.5.3.18 LStyleT6 168
7.5.3.19 LStyleT7 168
7.5.3.20 LStyleT8 168
7.5.3.21 LStyleT9 168
7.6 Presentation style constituent constraints 169
7.6.1 Macro definitions 169
7.6.2 Factor constraints 169
7.6.3 Constituent constraints 169
7.6.3.1 PStyle1 169
7.6.3.2 PStyle2 169
7.6.3.3 PStyle3 170
7.6.3.4 PStyle4 170

7.7 Content portion constituent constraints 170

7.7.1 Macro definitions 170

7.7.2 Factor constraints 170

7.7.3 Constituent constraints 171

7.7.3.1 Character-content-portion 171

7.7.3.2 Raster-graphics-content-portion 172

7.7.3.3 Geometric-graphics-content-portion 172

8 Interchange format 172

8.1. Interchange format class A 172

8.1.1 Interchange format 172

8.1.2 Document application profile object identifier 172

8.1.3 Encoding of application comments 172

8.1.4 Data lengths 173

Annex A – Amendments and technical corrigenda 173

A.1 Amendments 173

A.1.1 Amendments to the base standard 173

A.1.2 Proposed changes to standards due to defects 173

A.2 Technical corrigenda 173

A.2.1 Technical corrigenda to this Recommendation 173

A.3 Versions of the ODA Recommendation 174

Annex B – Recommended practices 174

B.1 Transfer methods for ODA 174

B.1.1 Conveyance of ODA over CCITT X.400 (1984) 174

B.1.2 Conveyance of ODA over FTAM 174

B.1.3 Conveyance of ODA over DTAM 174

B.1.4 Conveyance of ODA over flexible disks 175

B.2 Font reference 175

B.3 ISO 8632 (CGM) constraints for this document application profile 176

B.3.1 Delimiter elements 176

B.3.2 Metafile descriptor elements 176

B.3.3 Picture descriptor elements 177

B.3.4 Control elements 177

B.3.5 Graphical primitive elements 177

B.3.6 Attribute elements 178

B.3.7 External elements 180

Annex C – Bibliography 180

FOREWORD

Development of this document application profile has been done in liaison with several organizations. These include ODA expert groups within the:

- Asia-Oceania Workshop (AOW);
- Study Group 8;
- European Workshop for Open Systems (EWOS);
- OSE implementors’ Workshop (OIW).

The liaison between these organizations has occurred within the meetings of the Profile alignment Group for ODA (PAGODA). These meetings have focused on the development of a single set of internationally aligned ODA document application profiles.

This Recommendation is specific to the profile identifier PM-36.

At present, this Recommendation consists of one part:

- Document application profile.

Further parts may be added to this Recommendation.

This part contains three annexes:

- Annex A (normative): Amendments and technical corrigenda;
- Annex B (informative): Recommended practices;
- Annex C (informative): Bibliography.

INTRODUCTION

The purpose of this Recommendation is to facilitate the interworking of applications interchanging documents based on CCITT Recs. T.410-Series | ISO 8613 ODA. This Recommendation is suitable for interchanging documents in formatted form, processable form or formatted processable form and has been defined in accordance with CCITT Rec. T.411 | ISO 8613-1. The format of this Recommendation is in accordance with the standardized proforma and notation defined in Annex F/T.411.

Recommendation T.506

DOCUMENT APPLICATION PROFILE PM-36 FOR THE INTERCHANGE OF EXTENDED DOCUMENT STRUCTURES AND MIXED CONTENT DOCUMENTS IN PROCESSABLE AND FORMATTED FORMS[1)]

1 Scope

This Recommendation specifies interchange formats for the transfer of structured documents between equipment designed for word or document processing. Such documents may contain character, raster graphics and geometric graphics content.

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

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

a) formatted form;

b) processable form;

c) 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 Recommendation 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 Normative references

The following Recommendations | International Standards contain provisions which, through reference in this text, consititute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation are warned against automatically applying any more recent editions of the Recommendations and Standards listed below, since the nature of references made by ISPs to such Recommendations and Standards is that they may be specific to a particular edition. Members of IEC and ISO maintain registers of currently valid International Standards and International Standardized Profiles. The TSB maintains a list of the currently valid ITU-T Recommendations.

2.1 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation T.411 (1988), *Open Document Architecture (ODA) and Interchange Format: Introduction and General Principles.*

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

- CCITT Recommendation T.411 (1991), *Open Document Architecture (ODA) and Interchange Format: Introduction and General Principles – Annex F: Document Application Profile Proforma and Notation.*

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

[1)] ITU-T Recommendation and ISO/IEC ISP 11182-1 were developed in close collaboration and are technically aligned, except for the use of SGML notation and SDIF interchange format.

- CCITT Recommendation T.412 (1988), *Open Document Architecture (ODA) and Interchange Format: Document Structures.*

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

- CCITT Recommendation T.414 (1988), *Open Document Architecture (ODA) and Interchange Format: Document Profile.*

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

- CCITT Recommendation T.415 (1988), *Open Document Architecture (ODA) and Interchange Format: Open Document Interchange Format (ODIF).*

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

- CCITT Recommendation T.416 (1988), *Open Document Architecture (ODA) and Interchange Format: Character Content Architecture.*

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

- CCITT Recommendation T.417 (1988), *Open Document Architecture (ODA) and Interchange Format: Raster Graphics Content Architectures.*

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

- CCITT Recommendation T.418 (1988), *Open Document Architecture (ODA) and Interchange Format: Geometric Graphics Content Architectures.*

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

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

  ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1).*

- CCITT Recommendation X.209 (1988), *Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1).*

  ISO/IEC 8825:1990, *Information technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*

- CCITT Recommendation T.502 (1991), *Document Application Profile PM-11 for the Interchange of Character Content Documents in Processable and Formatted Forms.*

  ISO/IEC ISP 10610-1:1992, *Information technology – International Standardized Profile FOD 011 – Open Document Format: Simple document structure – Character content architecture only – Part 1: Document Application Profile (DAP).*

- CCITT Recommendation T.505 ((1991), *Document Application Profile PM-26 for the Interchange of Mixed Content Documents in Processable and Formatted Forms.*

  ISO/IEC ISP 11181-1:1992, *Information technology – International Standardized Profile FOD 026 – Open Document Format: Enhanced document structure – Character, raster graphics and geometric graphics content architectures – Part 1: Document Application Profile (DAP) .*

2.2 Additional references

- CCITT Recommendation T.4 (1988), *Standardization of group 3 facsimile apparatus for document transmission.*
- CCITT Recommendation T.6 (1988), *Facsimile coding schemes and coding control functions for group 4 facsimile apparatus.*
- CCITT Recommendation T.400 (1988), *Introduction to Document Architecture, Transfer and Manipulation.*
- ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange.*

- ISO 2022:1986, *Information processing – ISO 7-bit and 8-bit coded character sets – Code extension techniques.*
- ISO 2375:1985, *Data processing – Procedure for registration of escape sequences.*
- ISO 6937-2:1983, *Information processing – Coded character sets for text communication – Part 2: Latin alphabetic and non-alphabetic graphic characters.*
- ISO 6937-2:1983/Add. 1:1989, *Information processing – Coded character sets for text communication – Part 2: Latin alphabetic and non-alphabetic graphic characters – Addendum 1.*
- ISO/IEC 7350:1990, *Information technology – Registration of repertoires of graphic characters from ISO 10367.*
- ISO/IEC 8632:1992, *Information technology – Computer graphics – Metafile for the storage and transfer of picture description information*
- ISO 8859-1:1987, *Information processing – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1.*
- ISO 8879:1986, *Information processing – Text and office systems – Standard Generalized Markup Language (SGML).*
- ISO 9069:1988, *Information processing – SGML support facilities – SGML Document Interchange Format (SDIF).*
- ISO 9293:1987, *Information processing – Volume and file structure of flexible disk cartridges for information interchange.*
- ISO/IEC 9541:1991, *Information technology – Font information internchange.*
- ISO/IEC TR 10000-1:1990, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 1: Framework.*
- ISO/IEC TR 10000-2:1990, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 2: Taxonomy of Profiles.*
- ISO/IEC 11182-1:1993, *Information technology – International standardized Profile FOD36 – Open Document Format: Extended document structure – Character, raster graphics and geometric graphics content architectures – Part 1: Document Application Profile (DAP).*

3 Definitions

For the purposes of this Recommendation the following definitions apply.

The definitions given in CCITT Rec. T.411 | ISO 8613-1 are applicable to this Recommendation.

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

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

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

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

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

4 Relationship with other International Standardized Profiles

This Recommendation belongs to a series of hierarchically related profiles FOD 011 | PM-11 and FOD 026 | PM-26.

The features supported by this Recommendation are a superset of the features supported by the profiles FOD 011 | PM-11 and FOD 026 | PM-26 and thus all data streams that are conformant to FOD 011 | PM-11 and FOD 026 | PM-26 are also conformant to this Recommendation apart from the DAP identifier.

5 Conformance

In order to conform to this Recommendation, a data stream representing a document shall 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 Recommendation.

5.1 Data stream conformance

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

a) the data stream shall be encoded in accordance with the ASN.1 encoding rules defined in CCITT Rec. X.209 | ISO 8825;

b) the data stream shall be structured in accordance with the interchange formats defined in clause 8;

c) the document, as represented by the data stream after resolution of any external references, shall be structured in accordance with one of the documents architecture classes as defined in 6.1 and shall contain all mandatory constituents specified for that class; other constituents may be included, provided that they are permitted for that class as specified in clause 7;

d) each constituent shall contain all those attributes specified as required for that constituent in this Recommendation; 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 Recommendation;

f) the encoded document shall be constructed in accordance with the abstract document architecture defined in CCITT Rec. T.412 | ISO 8613-2;

g) the document shall be structured in accordance with the characteristics and constraints specified in clause 6.

5.2 Implementation conformance

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

The implementation requirements associated with this Specification are defined in Recommendation T.516 to be developed.

6 Characteristics supported by this document application profile

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

6.1 Overview

6.1.1 General

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

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

In addition this Recommendation supports the interchange of

- generic-document;
- document profile.

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

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

6.1.2 Formatted form documents

Required constituents:

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

Optional constituents:

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

6.1.3 Processable form documents

Required constituents:

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

Optional constituents:

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

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

6.1.4 Formatted processable form documents

Required constituents:

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

Optional constituents:

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

6.1.5 Generic documents

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

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

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

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

6.1.6 Document profile

This form of document contains a document profile only.

6.2 Logical characteristics

6.2.1 Introduction

This subclause defines the logical constituent constraints provided by this Recommendation to represent the characteristics of documents containing logical component descriptions.

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

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

6.2.2 Overview of the logical structure

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

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

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

6.2.3 Body part of the logical structure

6.2.3.1 DocumentLogicalRoot

DocumentLogicalRoot is a constituent constraint representing the top level in the document logical structure. Its immediate subordinates consist of an arbitrary ordered sequence of one or more constituent constraints of the types Passage and NumberedSegment.

The automatic numbering schemes that apply to constituent constraints of the types NumberedSegment, Figure, Table, NumberedList and Footnote may be initialized on the DocumentLogicalRoot.

6.2.3.2 Passage

Passage is a constituent constraint that represents a subdivision of a document that corresponds to a logical grouping of subordinate parts of a document. This grouping may be regarded as a logical entity for reading, or it may represent parts of a document that have common layout and presentation characteristics.

Passages are typically used to represent:

- the contents to be placed on the title page of a report;
- the front matter, consisting of the table of contents or foreword;
- the main matter of the document;
- the back matter, consisting of appendices, glossary or index;
- an illustration with associated text which is inserted as a distinct entity within a section of a document.

The automatic numbering schemes that apply to subordinate constituent constraints of the types NumberedSegment, Figure, Table, NumberedList and Footnote may be initialized on a Passage.

The immediate subordinates of a Passage consist of an optional constituent constraint of the type Title which is followed by an arbitrary ordered sequence of one or more of the following constituent constraints:

- Paragraph;
- BodyGeometric;
- BodyRaster;
- BodyText;
- Figure;
- Table;
- NumberedList;
- UnNumberedList;
- DefinitionList;
- NumberedSegment;
- Passage.

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

Therefore, a Passage may itself contain one or more subordinates of the type Passage, and thus Passages may be nested to any number of levels. This allows logical entities within a document to be described in terms of their component logical entities. Also, a NumberedSegment may contain one or more subordinate Passages which allows different logical entities within a NumberedSegment to be distinguished.

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

6.2.3.3 NumberedSegment

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

The automatic numbering schemes that apply to the subordinate constituent constraints NumberedSegment, Figure, Table, NumberedList and Footnote may be initialized on any logical object or logical object class, typically on Passage or a NumberedSegment.

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

- Paragraph;
- BodyGeometric;
- BodyRaster;
- BodyText;
- Figure;
- Table;
- NumberedList;
- UnNumberedList;
- DefinitionList;
- Passage;
- NumberedSegment.

The above indicates that a document may contain any number of nested levels of the constituent constraint NumberedSegment.

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

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

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

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

6.2.3.4 Number

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

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

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

NOTE – The separator may be an empty string.

Further details of the structure and generation of the identifiers are given in 6.6.6.

6.2.3.5 Paragraph

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

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

- BodyText;
- BodyRaster;
- BodyGeometric;
- Footnote;
- Phrase;
- Reference.

The character content associated with a Paragraph may be concatenated to form a continuous stream of character content which may contain single or multiple references to footnotes or other parts of the document, and may be laid out as single unit. Entities within this content which have particular logical significance or presentation features may be distinguished using the constituent constraint Phrase.

Content from subordinates of a paragraph may be separated one from another to give white space between them using Separation (see 6.4.2.2). This may be used to give an effect similar to that achieved with empty lines of text. Use of empty text lines to achieve white space between areas of text or other content may lead to unintended blank areas adjacent to the leading edge of layout objects (e.g. at page breaks) whereas the use of Separation avoids this.

6.2.3.6 BodyText, BodyRaster and BodyGeometric

BodyText, *BodyRaster* and *BodyGeometric* are constituent constraints which represent the lowest level of logical subdivision of a document. These constituent constraints act as carriers for the document content and may be specified as subordinates of constituent constraints:

- Passage;
- NumberedSegment;
- Paragraph;
- Title;
- ListTerm;
- Artwork;
- Phrase;
- Reference;
- UnNumberedList.

In addition, BodyText may be specified as a subordinate to Phrase, Caption and Description. These constituent constraints allow the layout and presentation requirements of different parts of the content of a document to be specified.

These are basic logical constituent constraints that directly refer to content portions that contain character, raster graphics and geometric graphics content respectively. BodyText shall refer to one or more content portions which may contain either processable, formatted or formatted processable character content. BodyRaster and BodyGeometric shall only refer to a single content portion containing formatted processable raster graphics content or formatted processable geometric graphics content respectively.

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

Where the superior constituent constraint referenced is subordinate to a FootnoteBody, it is required to specify one of the layout category names for this constituent constraint, 'Footnote' or 'Footnote-<n>'. This along with a “permitted categories” attribute of the same name on the footnote frame will ensure that a logical object from this constituent constraint is laid out in a FootnoteArea frame when generic layout structure is specified within the document.

6.2.3.7 Footnote

Footnote is a constituent constraint that is used to represent footnotes within a document. This constituent constraint may be specified as a subordinate to:

– Paragraph;

– Title;

– ListTerm;

– Phrase;

– Caption;

– Description.

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

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

6.2.3.8 FootnoteReference

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

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

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

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

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

The format of the content generator referred to above is described in 6.6.6.7.

6.2.3.9 FootnoteBody

FootnoteBody is a constituent constraint which represents the content of a footnote. The content consists of a stream of character content which may contain embedded references to other parts of the document.

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

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

6.2.3.10 FootnoteNumber

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

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

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

It is required to specify one of the layout category names for this constituent constraint, ’Footnote’ or ’Footnote-<n>’. This along with a “permitted categories” attribute of the same name on the footnote frame will ensure that a logical object from this constituent constraint is laid out in a FootnoteArea frame when generic layout structure is specified within the document.

6.2.3.11 FootnoteText

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

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

It is required to specify one of the layout category names for this constituent constraint, 'Footnote' or 'Footnote-<n>'. This along with a “permitted categories” attribute of the same name on the footnote frame will ensure that a logical object from this constituent constraint is laid out in a FootnoteArea frame when generic layout structure is specified within the document.

6.2.3.12 Title

Title is a constituent constraint that is used to represent the title, heading or name of the Passage or NumberedSegment to which it is an immediate subordinate. This constituent constraint may consist of character, raster graphics and geometric graphics content.

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

- BodyText;
- BodyRaster;
- BodyGeometric;
- Phrase;
- Reference;
- Footnote.

The character content associated with a Title may be concatenated to form a continuous stream of character content which may contain single or multiple references to footnotes or other parts of the document, and may be laid out as single unit. Entities within this content which have particular logical significance or presentation features may be distinguished using the constituent constraint Phrase. Content from any subordinate basic text objects within a paragraph may be run-on one from another (that is, to continue on the same line) by use of Concatentation (see 6.4.2.5). Alternatively, content from subordinates of a paragraph may be separated one from another to give white space between them, using Separation (see 6.4.2.2). This may be used to give an effect similar to that achieved with empty lines of text.

Use of empty text lines to achieve white space between areas of text or other content may lead to unintended blank areas adjacent to the leading edge of layout objects, whereas the use of Separation avoids this.

6.2.3.13 Phrase

Phrase is a constituent constraint that is used to group together an amount of character content that represents a single logical entity that needs to be distinguished for some purpose. That is, the content represented by a Phrase may have a particular logical significance, or it may have certain layout or presentation requirements. The character content may contain embedded footnotes and references to other parts of the document content. A typical example is a quotation that is to be reproduced in italics.

A Phrase may be used as subdivision of constituent constraints of the types Paragraph, Title, Caption, Description, Artwork and ListItem. Also, a Phrase may be subordinate to another Phrase and, therefore, constituent constraints of the type Phrase may be nested.

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

- BodyText;
- Phrase;
- Reference;
- Footnote.

The character content associated with a Phrase may be concatenated to form a continuous stream of character content which may contain single or multiple references to footnotes or other parts of the document, and may be laid out as single unit. Alternatively, the character content may contain hard line terminators, which will cause parts of the content to be separated when laid out.

6.2.3.14 Constituents that provide for a general referencing mechanism

This subclause defines the constituent constraints that are provided to support a general referencing mechanism in a document.

6.2.3.14.1 Reference

A *Reference* is a constituent constraint which represents a reference consisting of character content that is derived either fully or partially from another part of the document. This constituent constraint provides a general cross-referencing mechanism in a document.

This constituent constraint may be specified as a subordinate to constituent constraints of the types:

- Paragraph;
- Title;
- ListTerm;
- Phrase;
- Caption;
- Description;
- FootnoteBody.

It is a composite constituent constraint whose immediate subordinates may consist of an ordered sequence of constituent constraints of the types BodyText, ReferencedContent and BodyText.

The general format of the content associated with a constituent constraint of the type Reference is:

[<prefix-string>]<reference-string>[<suffix-string>]

The prefix and suffix strings are optional and, if required, are represented by constituent constraints of the type BodyText. The reference string is represented by the constituent constraint, ReferencedContent.

A reference string may, for example, contain references to identifiers such as a number that distinguishes a chapter or section, a table, a figure, a footnote, an item in a numbered list or a page number. Each reference may contain multiple, concatenated references to different parts of a document; a typical example is the reference "see Table 3 in chapter 2 on page 4” where the values “3”, “2” and “4” are derived automatically from the appropriate table, chapter and page in the document.

6.2.3.14.2 ReferencedContent

ReferencedContent is a constituent constraint that represents a character string that contains a single reference to content in other parts of the document (see 6.2.3.14.1).

It is a basic logical constituent constraint that is an immediate subordinate to the constituent constraint Reference. It contains a content generator which, when evaluated, produces the character string containing the referenced content.

A sequence of two or more constituent constraints of this type may be used to represent a composite reference string such as “see Table 2 in section 3.1 beginning on page 6”, where the strings “2”, “3.1” and “6” are automatically generated by referring to number strings attached to particular parts of the document.

The format of this content generator and its evaluation is described in 6.6.6.9.

Where the superior constituent constraint referenced is subordinate to a FootnoteBody, it is required to specify one of the layout category names for this constituent constraint, 'Footnote' or 'footnote-<n>'. This along with a “permitted categories” attribute of the same name on the footnote frame will ensure that a logical object from this constituent constraint is laid out in a FootnoteArea frame when generic layout structure is specified within the document.

6.2.3.15 Constituents representing illustrations

6.2.3.15.1 Introduction

This Recommendation supports the representation of illustrations or figures consisting of artwork or simple forms.

Artwork typically consists of a diagram or figure which is an image composed of a single content type or which is formed by overlaying two or more separate images consisting of character, raster graphics or geometric graphics content.

A form typically consists of a collection of logical entities, each of which has a certain logical significance. Each logical entity may be further subdivided into subordinate logical entities. A typical example is an order form consisting of a reference number, information about the originator, including the name, address and telephone number, a list of items required and their expected delivery dates.

The entities which make up a form are intended to be laid out in a designated area which is subdivided into areas that are specially reserved for each particular type of entity. This designated area is specified by the frame FormArea when the generic layout structure is present in the document.

Optionally, an illustration may also contain an identifier which may be used to distinguish the illustration from other parts of the document, a caption which may be used to identify the purpose of the illustration, and an amount of associated descriptive text.

Further information concerning the layout of illustrations consisting of artwork or forms is given in 6.4.1.3.7.

The constituent constraints used to represent illustrations are defined below.

6.2.3.15.2 Figure

Figure is a constituent constraint that is typically used to represent an illustration. Such an illustration may consist of artwork or a form as described in 6.2.3.15.1.

A Figure is a composite logical constituent constraint which may be specified as a subordinate to a Passage or NumberedSegment.

The subordinates of a Figure may consist of a sequence, in any order, of the following constituent constraints:

- either Artwork or Form;
- Number;
- Caption;
- Description.

A constituent constraint of the type Artwork or Form shall always be present as a subordinate to the constituent constraint Figure. Constituent constraints of the types Number, Caption and Description are independently optional. The constituent constraints may occur in any order, except that a Number and a Caption shall be in this order.

6.2.3.15.3 Artwork

Artwork is a constituent constraint that is typically used to represent a graphical image within an illustration or figure. This may be a simple image that is represented by character, raster graphics or geometric graphics content, or it may be a composite image consisting of a combination of these content types.

This constituent constraint shall only occur as a subordinate to a constituent constraint of the type Figure.

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

– Phrase;

– BodyRaster;

– BodyGeometric.

The constituent constraint Phrase is used to enable the image to contain character content which contains references to footnotes or other parts of the document.

6.2.3.15.4 Form

Form is a constituent constraint that is used to represent a simple form within an illustration or figure. It is a composite logical constituent constraint which shall only occur as a subordinate to a constituent constraint of the type Figure.

A Form consists of an arbitrary order sequence of basic entries and composite entries. Basic entries, which may consist of character, raster graphics or geometric graphics content, are represented by subordinate constituent constraints of the type EntryElement. Composite entries are represented by subordinate constituent constraints of the type EntryGroup.

Constituent constraints of the type EntryGroup may be further subdivided into a set of one or more basic or composite entries. Thus a composite entry may be nested to any number of levels such that each level may consist of a set of subordinate basic and composite entries.

This Recommendation does not define a mechanism for defining the semantics associated with each basic or composite entry. This can be achieved by individual applications by making use of the parameter “external data” in the attribute “application comments” (see 6.6.4).

6.2.3.15.5 Caption

Caption is a constituent constraint that typically represents a title or header that is associated with an illustration or figure. This constituent constraint represents character content that may contain embedded references to footnotes and to other content within the document.

A Caption is a composite logical constituent constraint which shall only occur as a subordinate to a constituent constraint of the type Figure.

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

– BodyText;

– Reference;

– Phrase;

– Footnote.

The above constituent constraints may be concatenated to form a continuous stream of character content which is to be laid out as a single unit.

6.2.3.15.6 Description

Description is a constituent constraint that typically represents some general supplementary information that forms part of an illustration that contains a figure or form.

This constituent constraint is a composite logical constituent constraint which shall only occur as a subordinate to a constituent constraint of the type Figure.

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

- BodyText;
- Reference;
- Phrase;
- Footnote.

The above constituent constraints may be concatenated to form a continuous stream of character content which is to be laid out as a single unit.

6.2.3.15.7 EntryGroup

EntryGroup is a constituent constraint that represents a composite logical entry within a form.

It is a composite logical constituent constraint which shall only occur as a subordinate to a constituent constraint of the types Form or EntryGroup.

The immediate subordinates of a EntryGroup consist of an arbitrary ordered sequence of one or more constituent constraints of the types EntryElement and EntryGroup. Therefore, the subordinates to EntryGroup may be nested to any number of levels.

NOTE – Constituent constraint EntryElement is defined in 6.2.3.16.6.

6.2.3.16 Constituents used to represent tables

6.2.3.16.1 Introduction

For the purpose of this Recommendation, a table is a logical entity that consists of an ordered sequence of elements, called cells, that are arranged into a two dimensional array of rows and columns.

Each row may consist of a simple row which contains a sequence of one or more cells. Alternatively, a row may consist of a composite row which contains a single cell followed by a sequence of one or more subrows, each of which contains a sequence of cells.

The subclauses below define the logical constituents used to represent tables. Figure 28 illustrates the structural relationships between the constituents used to represent a table. Subclause 6.4.1.3.8 describes how tables are intended to be laid out.

6.2.3.16.2 Table

Table is a logical constituent constraint that represents a table as a whole. This constituent constraint may be specified as a subordinate to constituent constraints Passage and NumberedSegment.

The immediate subordinates of this constituent constraint consist of a sequence of constituent constraints Row. Each row may or may not have the same characteristics with regard to its sub-structure.

6.2.3.16.3 Row

Row is a constituent constraint that is a subordinate of the constituent constraint Table and represents a row of entries in a table. This may consist of a sequence of entries, or it may consist of a single entry followed by a sequence of subrows of entries.

In order to represent these two cases, the immediate subordinates of a Row may consist of either:

- a sequence of constituent constraints EntryElement; or
- a single constituent constraint EntryElement, followed by a single constituent constraint of the type TableComponent.

6.2.3.16.4 TableComponent

TableComponent is a constituent constraint that is a subordinate of the constituent constraint, Row, and represents a sequence of one or more subrows of entries within a row of a table.

The immediate subordinates of this constituent constraint consist of a sequence of one or more constituent constraints of the type RowComponent. Each RowComponent shall have the same characteristics with regard to its sub-structure.

6.2.3.16.5 RowComponent

RowComponent is a constituent constraint that is a subordinate of the constituent constraint TableComponent and represents a sub-row of entries in a row of entries in a table.

The immediate subordinates of this constituent constraint consist of a sequence of one or more components of the type EntryElement. Each EntryElement may or may not have the same characteristics with regard to its sub-structure.

6.2.3.16.6 EntryElement

EntryElement is a constituent constraint that represents a single entry in a form or table. It is a sub-division of constituent constraints of the types Table and Form. In the case of Table, it is specified as a subordinate to a Row or a RowComponent and represents a single entry in a table. In the case of Form, it is specified as a subordinate of a Form itself or of an EntryGroup.

Each entry in a table or form may consist of character, raster graphics or geometric graphics content and, hence, EntryElement has a single immediate subordinate constituent constraint of the type EntryText, EntryRaster or EntryGeometric.

6.2.3.16.7 EntryText, EntryRaster and EntryGeometric

EntryText, *EntryRaster* and *EntryGeometric* are constituent constraints which represent content that is to be entered into tables and forms. These constituent constraints may be specified as subordinates of the constituent constraint EntryElement and allow the layout and presentation requirements for the content allocated to tables and forms to be specified.

These are basic logical constituent constraints that directly refer to content portions that contain character, raster graphics and geometric graphics content respectively. EntryText shall refer to one or more content portions which may contain either processable, formatted or formatted processable character content. EntryRaster and EntryGeometric shall only refer to a single content portion containing formatted processable raster graphics content or formatted processable geometric graphics content respectively.

Constituent constraints of these types in the generic logical structure may refer to generic content. This provides the means of defining common content within tables and forms.

6.2.3.17 Constituents representing lists

6.2.3.17.1 Introduction

This Recommendation supports the representation of three types of lists, as follows:

- numbered lists consisting of ordered lists of items, each of which is preceded by an identifier such as an alphabetic character or numeral;
- unnumbered lists consisting of unordered lists of items, each of which may optionally be preceded by a separator such as a hyphen, bullet or small circle;
- definition lists consisting of lists of ordered pairs of items such as a term and its corresponding definition.

Each type of list may be nested without restriction, and one particular type of list may be composed of lists of other types. For example, an item in a numbered list can consist of a subordinate numbered list, unnumbered list or definition list.

The constituent constraints that may be used to represent these list types are defined below.

6.2.3.17.2 NumberedList

A *NumberedList* is a constituent constraint that is used to represent a list of items, each of which is preceded by an identifier that serves to distinguish that item.

This constituent constraint may be specified as an immediate subordinate to a Passage, NumberedSegment, or ListItem.

The immediate subordinates of a NumberedList consist of the constituent constraint Number which contains a content generator that generates the identifier corresponding to each item in the list, followed by the constituent constraint, ListItem. This pair of constituent constraints may be repeated without limitation.

Further information concerning the numbering of items in a list is contained in 6.6.6.5.

6.2.3.17.3 UnNumberedList

UnNumberedList is a constituent constraint that is used to represent a list of items, each of which may be preceded by an optional separator consisting of character, raster graphics or geometric graphics content.

This constituent constraint may be specified as an immediate subordinate to a Passage, NumberedSegment, or ListItem.

The immediate subordinates of an UnNumberedList consist of a separator, which is represented by a constituent constraint of the type BodyText, BodyRaster or BodyGeometric, followed by the constituent constraint ListItem. This pair of constituent constraints may be repeated without limitation.

6.2.3.17.4 DefinitionList

DefinitionList is a constituent constraint that represents a sequence of ordered pairs of items.

This constituent constraint may be specified as an immediate subordinate to a Passage, NumberedSegment, or ListItem.

The immediate subordinates of this item consist of the constituent constraint ListTerm, followed by the constituent constraint ListItem. This pair of constituent constraints may be repeated without limitation.

6.2.3.17.5 ListItem

ListItem is a constituent constraint that represents an item within a NumberedList, UnNumberedList or DefinitionList. That is, this constituent constraint represents the second element of each pair of elements within a numbered, unnumbered or definition list.

The immediate subordinates of this constituent constraint may consist of a sequence of one or more constituent constraints Phrase, or one of the constituent constraints NumberedList, UnNumberedList or DefinitionList.

Thus this constituent constraint represents an amount of character content that may contain embedded references to footnote and other parts of the document, or it represents a subordinate list of items.

6.2.3.17.6 ListTerm

ListTerm is a constituent constraint that represents a term element within a DefinitionList. A term element is the first item of each pair of items that constitutes a definition list.

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

- BodyText;
- BodyRaster;
- BodyGeometric;
- Phrase;
- Reference;
- Footnote.

Thus, this constituent constraint represents an amount of character, raster graphics and geometric graphics content. Character content may contain embedded phrases, references and footnotes.

6.2.4 Common content part of the logical structure

6.2.4.1 CommonContent

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

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

- CommonText;
- PageNumber;
- CommonRaster;
- CommonGeometric;
- CommonReference;
- CurrentInstance;
- TableNumber;
- CommonNumber.

When the generic layout structure is present, constituent constraints of the type CommonContent and their associated subordinate constituent constraints are constrained to be laid out in a specified frame within a body, header or footer area using the logical source mechanism (see 6.3.6).

6.2.4.2 CommonText

CommonText is a constituent constraint that represents the common character content that is to be laid out in a specified area within a page.

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

6.2.4.3 PageNumber

PageNumber is a constituent constraint that represents the common character content that is to be laid out in a specified area in a page. This constituent constraint is specifically used when it is required to represent 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. For example, this provides the means of representing the page numbers that are displayed on the consecutive pages of a document.

Each page number consists of a single number which may be represented in the form of Arabic or Roman numerals or in its alphabetic equivalent. Page numbering schemes may start at 0 or any value greater than 0. The page number that is generated may have a prefix or suffix character string.

The format of the content generators is defined in 6.6.6.8.

6.2.4.4 CommonRaster

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

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

6.2.4.5 CommonGeometric

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

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

6.2.4.6 CommonReference

CommonReference is a constituent constraint that represents the common character content that is to be laid out in a specified area in a page and which represents a character string that contains references to other parts of the document. Such a reference may consist of a reference to a number that relates to a segment, table, figure, footnote or page number.

CommonReference is a constituent constraint for a basic logical object class that contains a content generator which, when evaluated, produces a character string containing the references. The format of this reference string is defined in 6.6.6.10.

6.2.4.7 CurrentInstance

CurrentInstance is a constituent constraint that represents the common character content that is to be laid out in a specified area in a page. This constituent constraint is specifically used when it is required to refer to a character string that is attached to any logical constituent constraint in the document or any of the layout constituent constraints DocumentLayoutRoot, PageSet, RectoPage, VersoPage or Page. The number string may represent, for example, the title of a sub-section or table that is contained elsewhere in the document.

CurrentInstance is a constituent constraint for a basic logical object class that contains a content generator which, when evaluated, produces a copy of the character string associated with a specified constituent. The format of this reference is defined in 6.6.6.11.

6.2.4.8 TableNumber

TableNumber is a constituent constraint that represents the common character content that is to be laid out in a specified area in a page. This constituent constraint is specifically used when it is required to represent a table number which is to be placed within the header area of a table.

TableNumber is a constituent constraint for a basic logical object class that contains a content generator which, when evaluated, generates the required table number. The format of the content generator is defined in 6.6.6.6.

6.2.4.9 CommonNumber

CommonNumber is a constituent constraint that represents the common character content that is to be laid out in a specified area in a page. This constituent constraint is specifically used when it is required to refer to character content consisting of a number string which can, for example, represent the number of a sub-section or a current page.

CommonNumber is a constituent constraint for a basic logical object class that contains a content generator which, when evaluated, generates the reference to the required number. The format of this reference is defined in 6.6.6.12.

6.3 Layout characteristics

This subclause defines the layout constituent provided by this Recommendation to represent the characteristics of documents.

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

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

6.3.1 Overview of the layout characteristics

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

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

It should be noted that, in the case of FOD036 | PM-36, common content may also be laid out in the body area, as well as the header/footer area.

Three body layout types are provided by this Recommendation. Each body layout type specifies how the body is positioned within each page, and how the content may be presented within the body. These are referred to as body layouts A, B, and C, and are defined in 6.3.4.5.

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

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

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

Body layout C is used when the character content is to be laid out vertically and from right to left within the body area. This layout may be typically used for contents written in languages which use ideograms, such as Japanese and Chinese characters.

The body, header and footer areas may be further subdivided into areas to support a wide range of different layout requirements. These features are described further in 6.3.5 and 6.3.6.

6.3.2 DocumentLayoutRoot

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

6.3.3 PageSet

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

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

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

6.3.4 Page characteristics

6.3.4.1 Page constituents

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

The pages that make up a page set consist of an arbitrary sequence of one or more of the constituent constraints Page, RectoPage and VersoPage.

The only difference in the characteristics of these types of constituent constraints concerns the values that may be specified for the parameter “side of sheet” in the attribute “medium type”. In the case of Page, 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 values 'recto' and 'verso' of the "side of sheet" parameter of the "medium type" attribute are non-basic.

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

a) all the pages shall have the same dimensions, but may differ in orientation (see 6.3.4.2);

b) all the pages are to be laid out on the same size of presentation medium (see 6.3.4.3);

c) all the pages instantiated from a given page class shall have the same layout characteristics. That is, for a given page class, there is not a choice of layout characteristics. However, the layout characteristics of pages in a page set may or may not be the same.

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

6.3.4.2 Page dimensions

The dimensions of the pages may be specified as any value (in BMUs) that is equivalent to or less than ISO A0 or ANSI-E paper sizes. The dimensions may be specified in portrait or landscape orientation. Japanese page sizes B4 and B5 are also supported, but the dimension of these pages lie within the range of dimensions given above.

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

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

NOTE – The size termed "North American Letter (NAL)" in CCITT Recs. T.410-Series | ISO 8613 (e.g. in CCITT Rec. T.412 | ISO 8613-2, clause 7) is called "ANSI-A" in this Recommendation to be consistent with the other reference to ANSI standard paper sizes.

6.3.4.3 Nominal page sizes

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

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

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

6.3.4.4 Page offset

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

6.3.4.5 Page layout characteristics

6.3.4.5.1 General characteristics

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

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

TABLE 1/T.506

Nominal page sizes

Page type	Size in inches or millimeters	Size in BMUs	ARA in BMUs
ISO A5	148 mm × 210 mm	7015 × 9920	not defined
ISO A4	210 mm × 297 mm	9920 × 14 030	9240 × 13 200
ISO A3	297 mm × 420 mm	14 030 × 19 840	13 200 × 18 480
ISO A2	420 mm × 594 mm	19 840 × 28 060	18 898 × 27 118
ISO A1	594 mm × 841 mm	28 060 × 39 680	26 173 × 37 843
ISO A0	841 mm × 1189 mm	39 680 × 56 120	37 843 × 54 283
ANSI legal	8.5 in. × 14 in.	10 200 × 16 800	9240 × 15 480
ANSI A	8.5 in. × 11 in.	10 200 × 13 200	9240 × 12 400
ANSI B	11 in. × 17 in.	13 200 × 20 400	12 744 × 19 656
ANSI C	17 in. × 22 in.	20 400 × 26 400	19 500 × 25 800
ANSI D	22 in. × 34 in.	26 400 × 40 800	25 800 × 39 600
ANSI E	34 in. × 44 in.	40 800 × 52 800	39 600 × 52 200
ANSI F	28 in. × 40 in.	33 600 × 48 000	32 400 × 47 400
Japan-legal	257 mm × 364 mm	12 141 × 17 196	11 200 × 15 300
Japan-letter	182 mm × 257 mm	8598 × 12 141	7600 × 10 200

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

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

6.3.4.5.2 Body area layout

Three types of layout of body area are defined:

- *Body layout type A* – In this case, the layout path in the body area is specified as 270 degrees.
- *Body layout type B* – In this case, the layout path in the body area is specified as 0 degrees.
- *Body layout type C* – In this case, the layout path in the body area is specified as 180 degrees.

6.3.4.5.3 Header/footer area layout

Four types of layout of header/footer area are defined:

- *H/F layout A1* – In this case, the layout path in the header and footer area is 270 degrees. If the header or footer area is composite, the layout paths in the lowest frames are 270 degrees.
- *H/F layout A2* – In this case, the layout path in the header and footer area is 0 degrees. If the header or footer area is composite, the layout paths in the lowest frames are 270 degrees.

- *H/F layout B1* – In this case, the layout path in the header and footer area is 180 degrees. If the header or footer area is composite, the layout paths in the lowest frames are 180 degrees.

- *H/F layout B2* – In this case, the layout path in the header and footer area is 270 degrees. If the header or footer area is composite, the layout paths in the lowest frames are 180 degrees.

6.3.4.5.4 Page layout

Page layout type is determined by a combination of the body layout type and the H/F layout type. Any combination is permitted. However, it is intended that all applications which use this Recommendation shall support the combinations of body layout A and H/F layout A1 and A2, whereas support for other combinations may be specified as optional.

NOTE – The combinations of body layout A and H/F layout A1, B and A1, C and A1, and C and B1 are equivalent to page layouts A, B, C and D respectively in FOD026 | PM-26.

The header and footer of H/F layout A1 or A2 are laid out above and below the body area. Figure 1 illustrates this case, and Figure 2 illustrates H/F layout type A1 and A2 corresponding to this case.

The header and footer of H/F layout B1 or B2 are laid out to the right and left of the body area. Figure 3 illustrates this case, and Figure 4 illustrates H/F layout type B1 and B2 corresponding to this case.

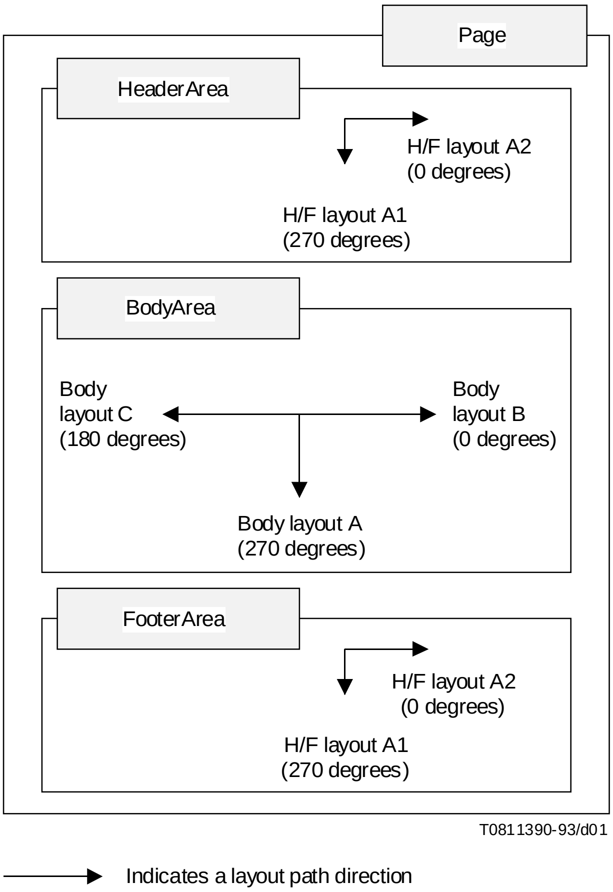
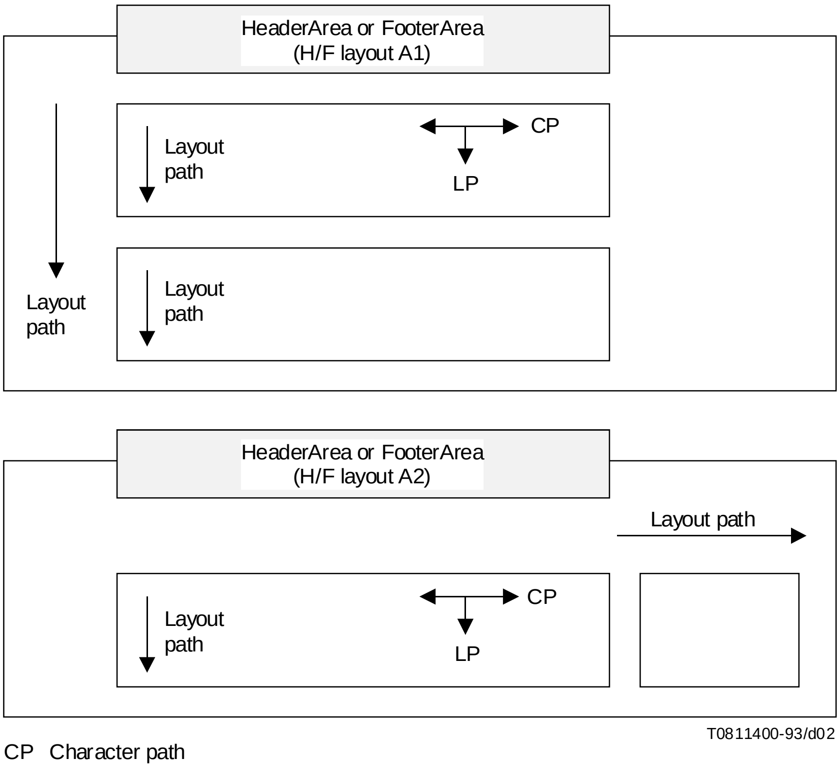


FIGURE 1/T.506

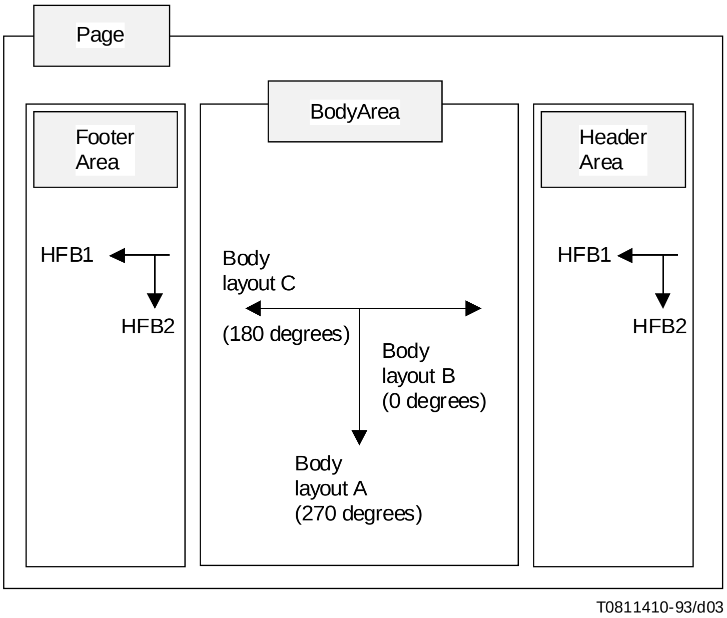
Body layout types A, B and C with header and footer above and below the body area



CP Character path
LP Line progression

FIGURE 2/T.506

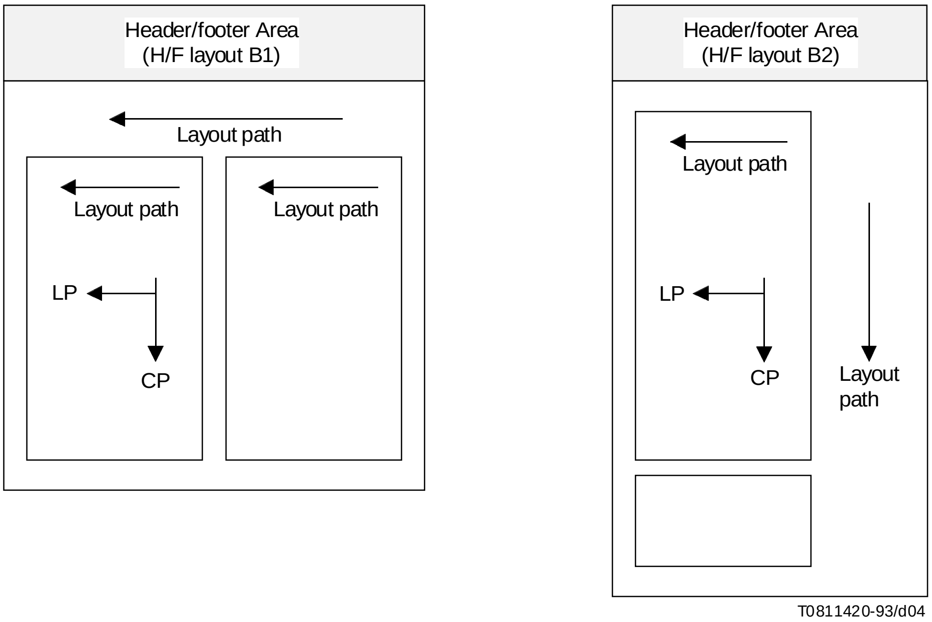
Header and footer frame layouts corresponding to Figure 1



HFB1 H/F layout B1 (180 degrees)
HFB2 H/F layout B2 (270 degrees)

FIGURE 3/T.506

Body layout types A, B and C with header and footer to the right and left of the body



CP Character path
LP Line progression

FIGURE 4/T.506

Header and footer frame layouts corresponding to Figure 3

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 layout path specified in the body area determines the body layout type being used.

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

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

The subordinate areas within a VariableCompositeBody frame are represented by variably positioned frames. Thus the subordinate areas are not pre-determined and are automatically adjusted during the layout process to accommodate the content that is allocated to them.

When a FixedCompositeBody frame is used, the subordinate areas are represented by fixed positioned frames, and hence the body area layout can be precisely specified.

However, in order to provide both areas whose layout is fixed and areas whose layout is variable within a single body area, it is possible to specify one or more VariableCompositeBody frames as subordinates to a FixedCompositeBody frame. In this case, the layout paths for each of the VariableCompositeBody frames may or may not be the same. This allows, for example, text belonging to different languages to be laid out on the same page.

6.3.5.2 BasicBody

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

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

6.3.5.3 VariableCompositeBody

VariableCompositeBody is a constituent constraint that defines a composite frame which represents either the entire body area or a part of it, and which contains one or more subordinate variably positioned frames. A VariableCompositeBody frame has a fixed position and fixed dimensions. The layout path specified for this frame depends upon the layout type used (see 6.3.1).

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

- BasicFloat;
- SnakingColumns;
- SynchronizedColumns;
- FootnoteArea;
- CompositeFloat;
- CompositeFixtureVariable;
- TableArea;
- ArrangedContentVariable;
- SourcedContentVariable.

The subordinate frames are all variably positioned and have variable dimensions except for ArrangedContentVariable frames. Thus, the relative positions of these frames in the body area may vary and depend upon the positions of other frames (if any) that are placed in the same VariableCompositeBody frame.

The layout path for VariableCompositeBody frames may be specified as 270, 0 or 180 degrees. This determines the body layout type used in the case where VariableCompositeBody represents the entire body area (see 6.3.4.5).

All immediate subordinate frames are laid out along the layout path specified (in normal positioning fill order). FootnoteArea frames are laid out in the same direction as the body area layout path, but reverse fill order is used.

Also frames are constrained to have the same layout path as the VariableCompositeBody frame to which they are subordinate. However, exceptions to this rule are frames of the types CompositeFixtureVariable, CompositeFloat, SnakingColumns and SourcedContentVariable (see appropriate subclause below).

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

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

6.3.5.4 BasicFloat

BasicFloat is a constituent constraint that defines a lowest level frame that is used to represent a single column area within a body area.

This is a variably positioned frame which may be specified as a subordinate to frames of the types VariableCompositeBody, CompositeColumnVariable, CompositeColumnFixed, CompositeFixtureVariable and CompositeFixtureFixed.

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

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

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

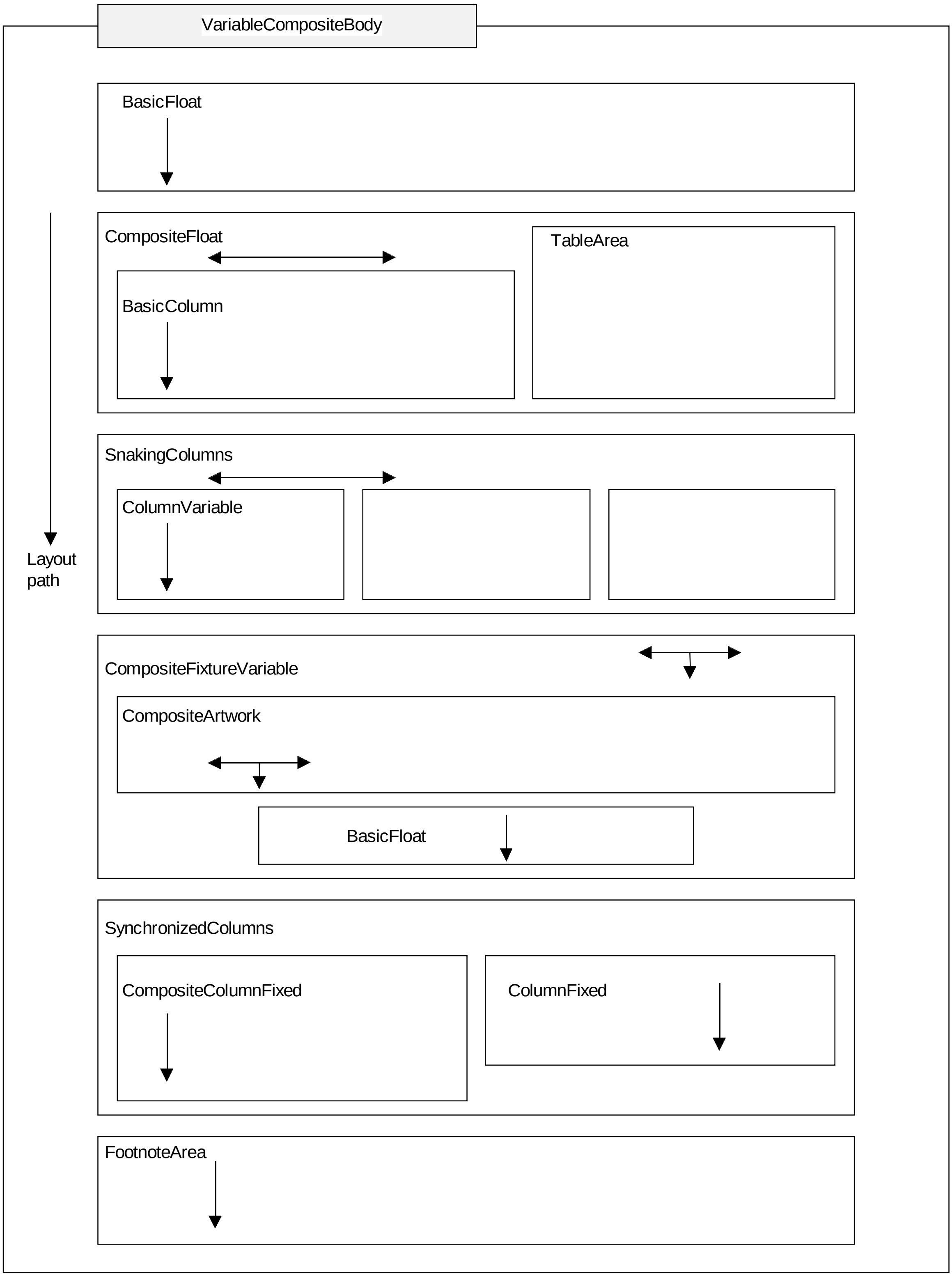


FIGURE 5/T.506

Example of body area layout for body layout type A

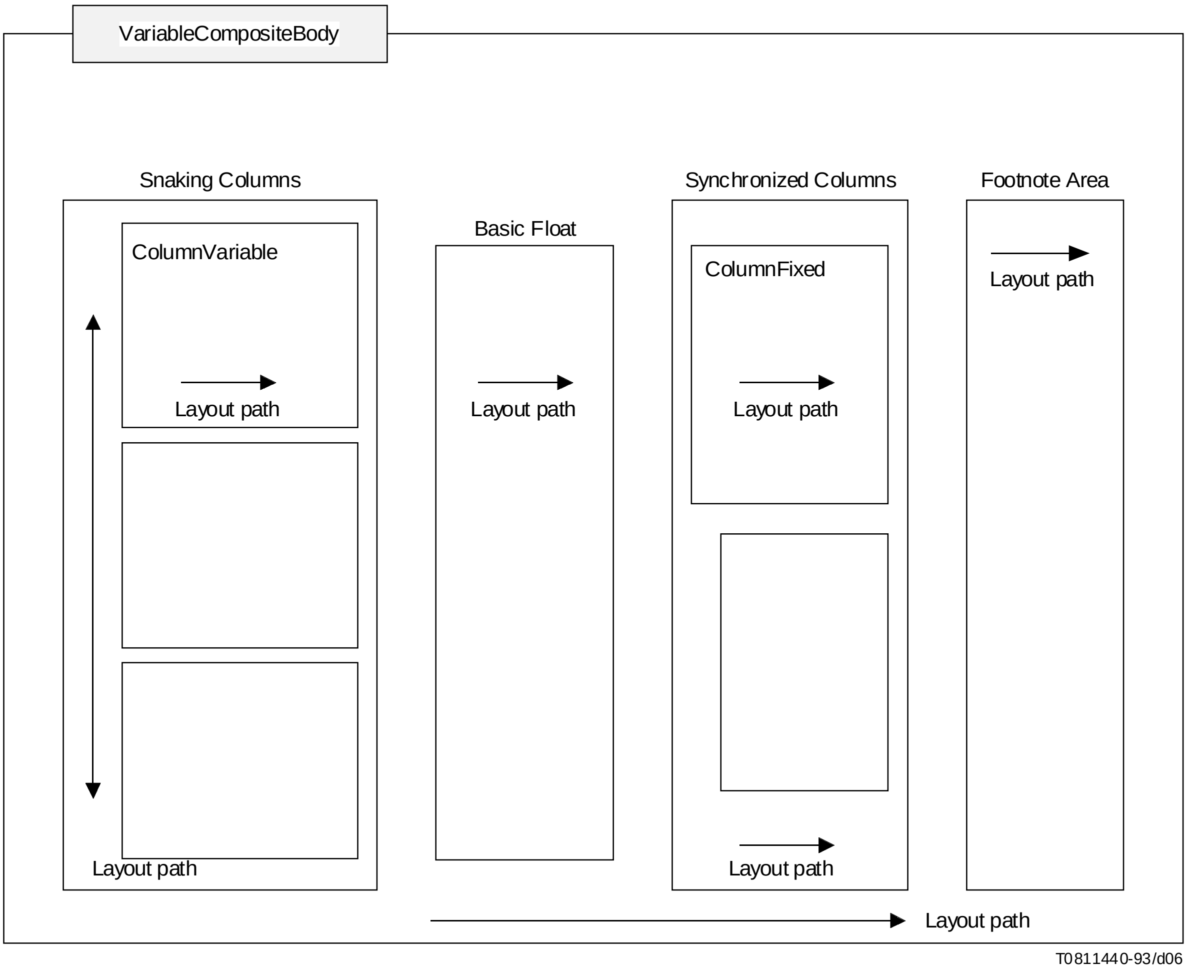


FIGURE 6/T.506

Example of body area layout for body layout type B

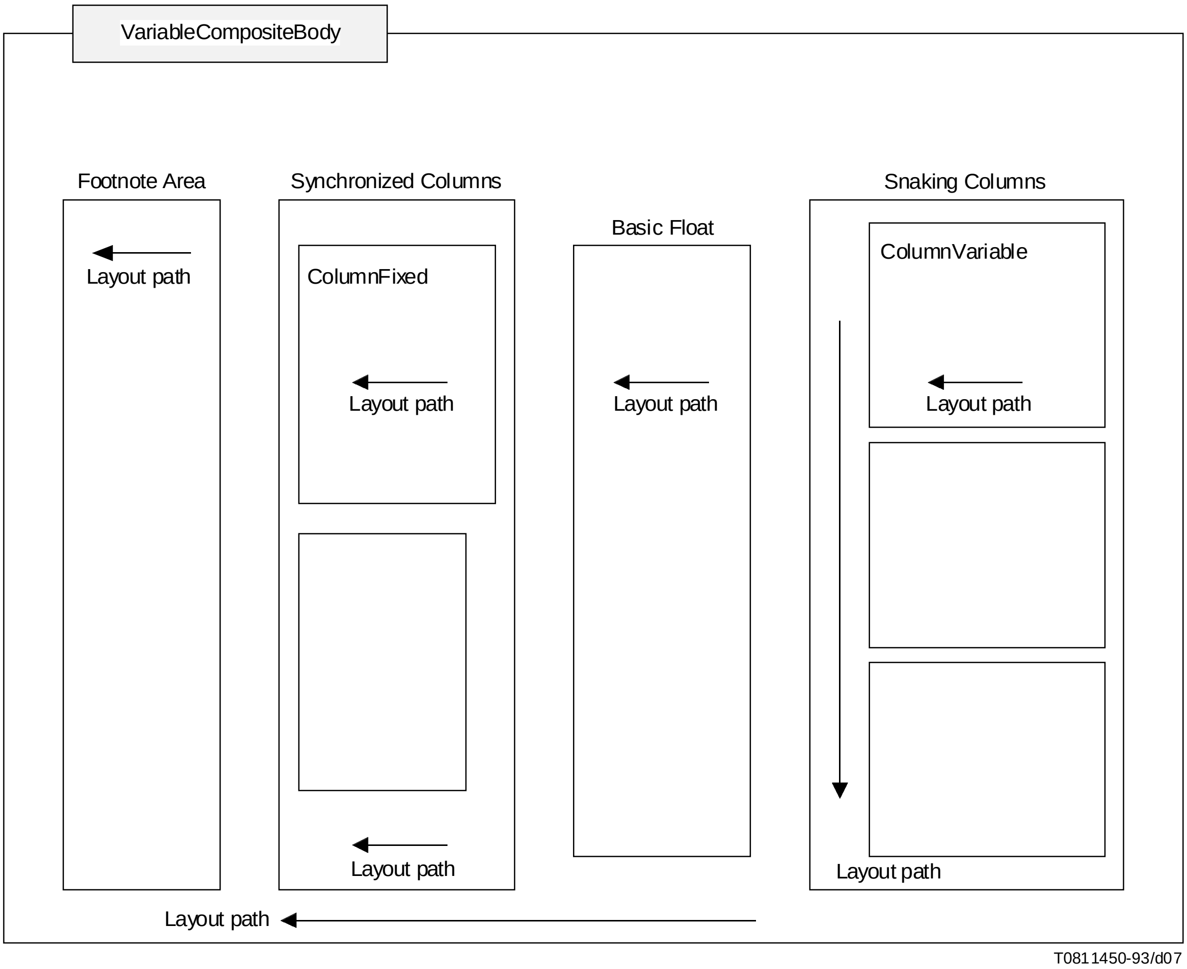


FIGURE 7/T.506

Example of body area layout for body layout type C

6.3.5.5 SnakingColumns

SnakingColumns is a constituent constraint that defines a composite frame that represents a snaking columns area within a body area. A snaking columns area is typically used for the layout of one or more columns of content in which the content is allowed to flow freely from one column to the next. Examples are shown in Figure 8 and Figure 9.

This is a variably positioned frame which may only be specified as a subordinate to a VariableCompositeBody frame.

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

– ColumnVariable;

– CompositeColumnVariable;

– ArrangedContentVariable;

– SourcedContentVariable.

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

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

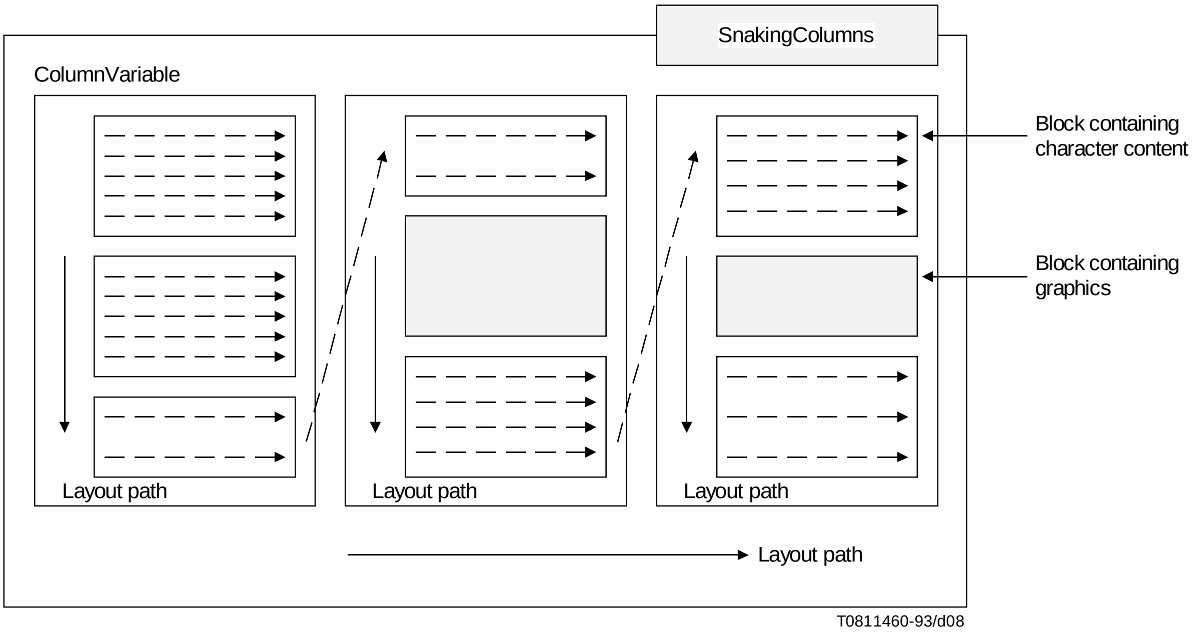


FIGURE 8/T.506

Example of the layout of a snaking columns frame

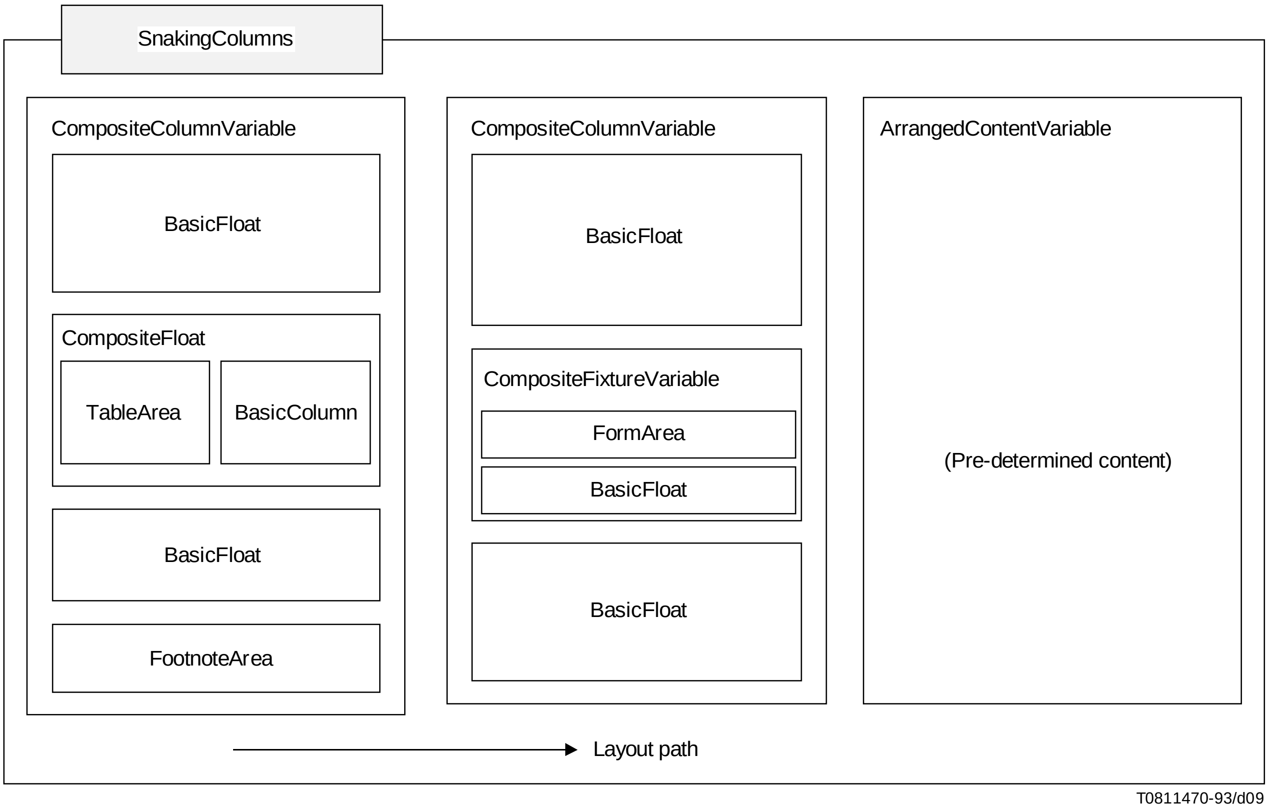


FIGURE 9/T.506

Another example of the layout of a snaking columns frame

The layout path for a SnakingColumns frame may be specified as 0 or 180 degrees in the case of body layout A, 90 or 270 degrees in the case of body layout B, and 270 degrees in the case of body layout C.

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

6.3.5.6 SynchronizedColumns

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

This type of column layout is typically used when it is required to layout separate amounts of content in parallel with one another such that they are aligned. Examples are the synchronized layout of content belonging to different languages and the layout of a figure in parallel with some text. An example is shown in Figure 10.

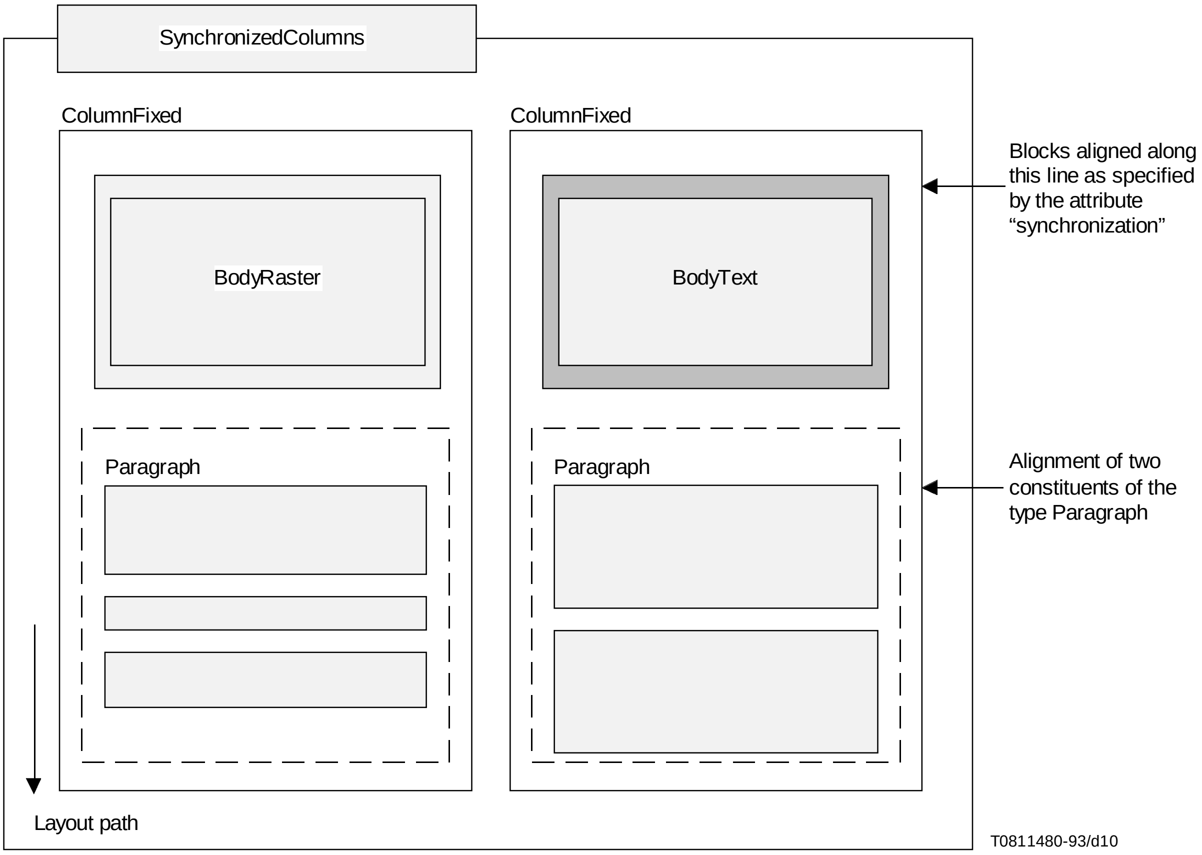

SynchronizedColumns
ColumnFixed
ColumnFixed
BodyRaster
BodyText
Blocks aligned along this line as specified by the attribute “synchronization”
Paragraph
Paragraph
Alignment of two constituents of the type Paragraph
Layout path
T0811480-93/d10

FIGURE 10/T.506

Example of the layout of a synchronized column

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

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

- ColumnFixed;
- CompositeColumnFixed;
- ArrangedContentFixed;
- SourcedContentFixed.

The layout path for a SynchronizedColumns frame is 270 degrees for body layout A, 0 degrees for body layout B and 180 degrees for body layout C.

6.3.5.7 ColumnVariable

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

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

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

The layout path for ColumnVariable frames is 270 degrees in the case of body layout A, 0 degrees in body layout B and 180 degrees in body layout C.

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

6.3.5.8 ColumnFixed

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

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

The dimension of this frame in the direction parallel to the layout path of the superior frame (i.e. the column length) may be specified by the sub-parameters "Rule-B" or "maximum-size" in the cases of body layouts A and B. In the case of body layout C, this dimension shall only be specified by the sub-parameter "maximum-size".

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

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

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

The values for the sub-parameter "maximum-size" shall only be specified for the last ColumnFixed frame laid out in a SynchronizedColumns frame to prevent overlapping of the frames. That is, for a page coordinate system with its reference point in the upper left corner, only the right most ColumnFixed frame shall specify values for the sub-parameter "maximum-size" without the risk of overlapping frames.

6.3.5.9 FootnoteArea

FootnoteArea is a constituent constraint that defines a lowest level frame that is used to represent an area reserved for the layout of footnotes. Footnotes may be placed in body areas and also in columns and areas reserved for illustrations within body areas.

This frame may be specified as a subordinate to frames of the constituent constraints:

- VariableCompositeBody;
- CompositeColumnVariable;
- CompositeColumnFixed;
- CompositeFixtureVariable;
- CompositeFixtureFixed.

Frames of this type are variably positioned with a positioning fill order specified as reverse. Therefore, this frame is positioned adjacent to the leading edge of its superior frame.

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

The layout path for FootnoteArea frames is the same as that specified for the superior frame.

The content that may be laid out in this frame is limited to the content that is associated with basic logical objects which are directly or indirectly subordinate to the composite logical object FootnoteBody. To achieve this, the “permitted categories” attribute of this frame shall specify the same category name required on the basic logical objects for footnotes (see 6.2.3.10 and 6.2.3.11).

6.3.5.10 FixedCompositeBody

FixedCompositeBody is a constituent constraint that defines a composite frame which represents the body area and which contains one or more subordinate frames that are fixed in position. The position and dimensions of this frame are fixed.

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

- BasicFixture;
- ColumnFixed;
- CompositeCommon;
- CompositeFixtureFixed;
- ArrangedContentFixed;
- SourcedContentFixed;
- VariableCompositeBody.

The subordinate frames may overlap without restriction.

The layout path for FixedCompositeBody frames may be specified as 270, 0 or 180 degrees. This determines the body layout type used (see 6.3.4.5). The layout path specified does not affect the positioning of frames, but it may affect their dimensions since some of the frames listed above have variable dimensions in a particular direction.

Also frames are constrained to have the same layout path as the FixedCompositeBody frame to which they are subordinate. However, exceptions to this rule are frames of the types VariableCompositeBody, CompositeFixtureFixed and SourcedContentFixed (see appropriate subclause below).

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

6.3.5.11 CompositeFloat

CompositeFloat is a constituent constraint that defines a composite frame that specifies an area which is used for the side by side layout of complex objects such as figures, forms or tables and simple columns of text.

This is a variably positioned frame which may be specified as a subordinate to a VariableCompositeBody, CompositeColumnVariable or CompositeColumnFixed frame.

The dimension of a CompositeFloat frame in the direction orthogonal to the layout path of its superior frame is fixed or defaults to the maximum value allowed.

The dimension of the frame in the direction parallel to the layout path is specified by the sub-parameter “Rule-A”. Thus the dimension in this direction is determined by the dimension in the same direction of the first frame that is laid out in the CompositeFloat frame.

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

- BasicColumn;
- CompositeFixtureVariable;
- TableArea;
- ArrangedContentVariable;
- SourcedContentVariable.

The layout path for a CompositeFloat may be specified as 0 or 180 degrees in the case of body layout A and as 90 or 270 degrees in the case of body layouts B and C.

A typical example of the use of this constituent is a form or an illustration with character content flowing to its side as shown in Figure 11.

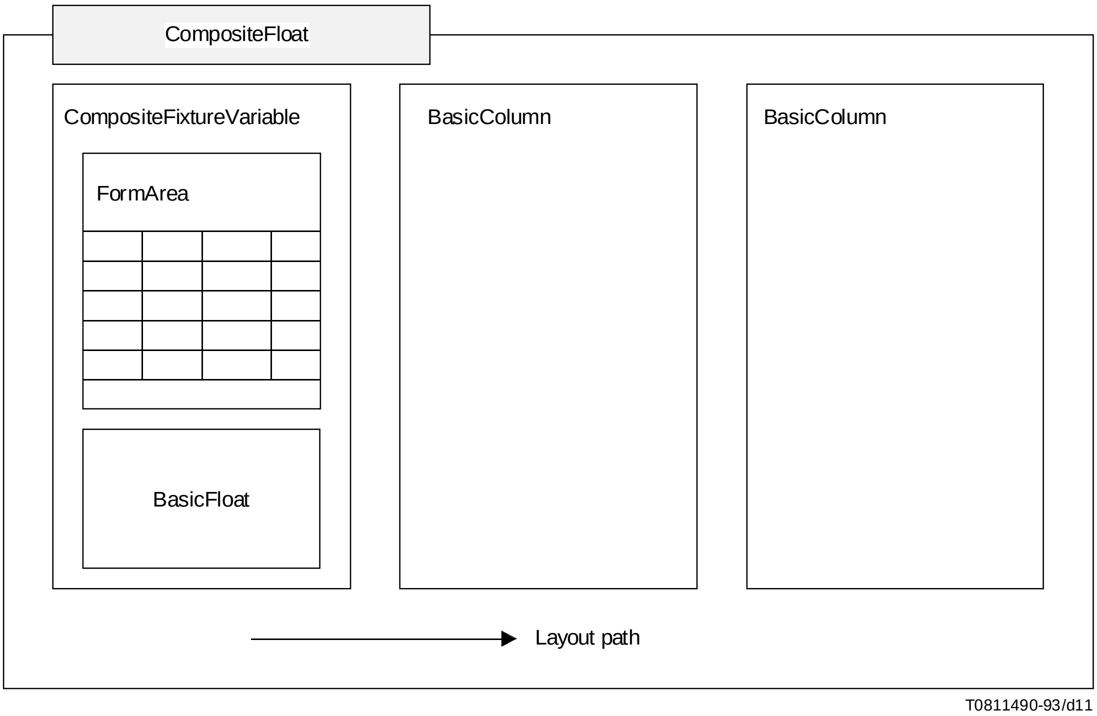


FIGURE 11/T.506

Example of a CompositeFloat frame

6.3.5.12 CompositeFixtureVariable

CompositeFixtureVariable is a constituent constraint that defines a composite frame used to specify an area which is used to layout an illustration, such as a figure or form, with which is associated some descriptive text and a caption. Hence, the prime purpose of this frame is to layout logical constituent constraints of the type Figure.

This is a variably positioned frame which may be specified as a subordinate to a VariableCompositeBody, CompositeColumnVariable or CompositeColumnFixed frame.

The dimension of a CompositeFixtureVariable frame in the direction orthogonal to the layout path of the superior frame is fixed, specified by the sub-parameter "Rule-B" or defaults to the maximum allowed. In the direction parallel to the layout path of the superior frame, the dimension may be fixed or specified by the sub-parameter "Rule-B".

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

– BasicFloat;

– CompositeArtwork;

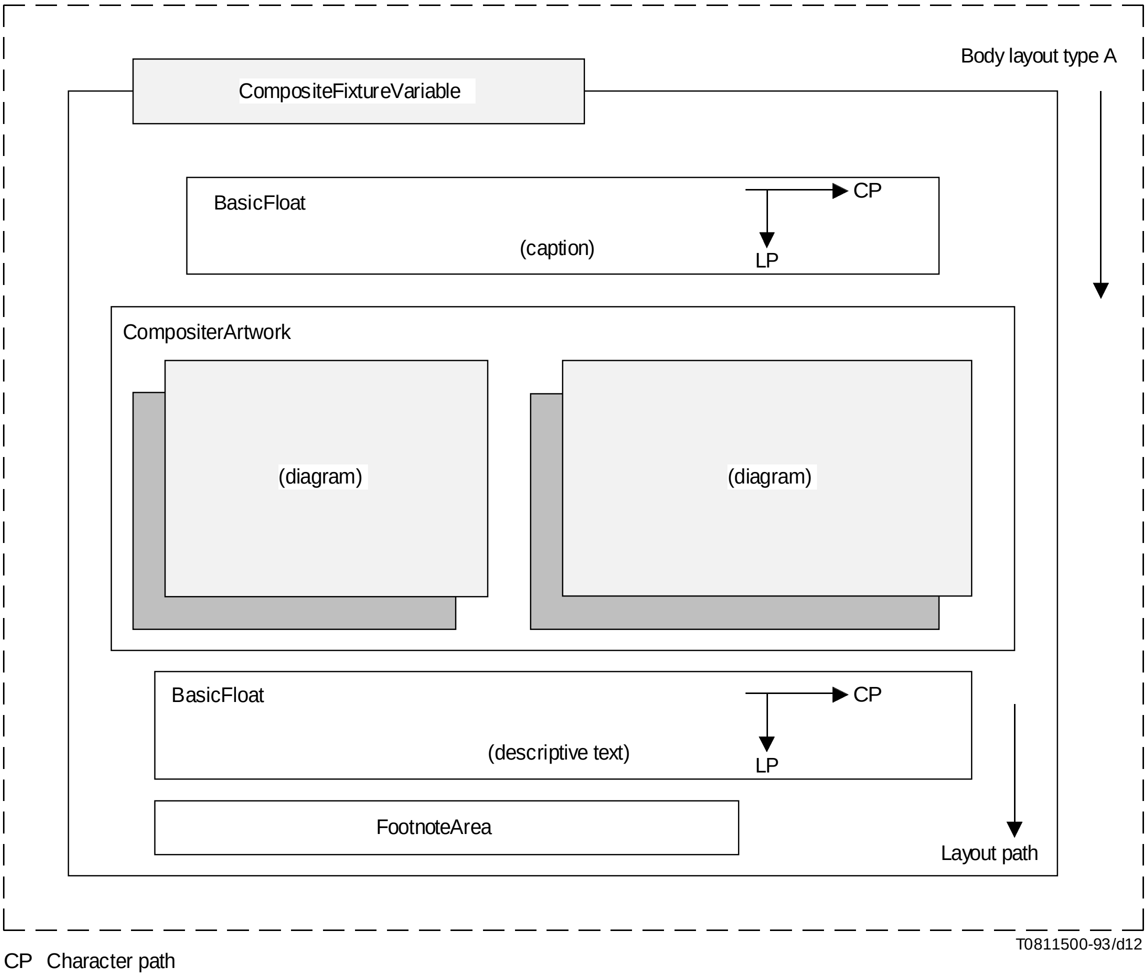
– FormArea;

– FootnoteArea.

The layout path of a CompositeFixtureVariable frame may be set to 0, 180 or 270 degrees in the case of body layout A, to 0, 90 or 270 degrees in the case of body layout B and 180 or 270 degrees in the case of body layout C.

All subordinate frames are laid out in normal positioning fill order with the exception of FootnoteArea frames which are laid out in reverse fill order.

Examples of the layout of CompositeFixtureVariable frames are shown in Figure 12.



CP Character path
LP Line progression

FIGURE 12/T.506

Example of a CompositeFixtureVariable frame

6.3.5.13 CompositeFixtureFixed

CompositeFixtureFixed is a constituent constraint that defines a composite frame which has the same characteristics as that of a CompositeFixtureVariable frame, except for the positioning and dimensions.

This frame may be specified as a subordinate to a FixedCompositeBody frame.

6.3.5.14 CompositeArtwork

CompositeArtwork is a constituent constraint that defines a composite frame that represents an area that contains an illustration such as a figure or diagram. It is used for laying out logical constituents of the type Artwork which are subordinate to logical constituent constraints of the type Figure.

This is a variably positioned frame that may be specified as a subordinate to a CompositeFixtureVariable or CompositeFixtureFixed frame.

The dimensions of this frame in the directions orthogonal and parallel to the layout path of the superior frame may be independently either fixed or specified by the sub-parameter "Rule-B".

The immediate subordinates of a CompositeArtwork frame consist of a sequence of one or more lowest level frames of the type BasicFixture which contain character, raster graphics or geometric graphics content. BasicFixture frames may overlap to allow composite images to be formed.

The layout path of a CompositeArtwork frame is set to be the same as that of its superior frame. Because of this restriction, the layout path of a CompositeArtwork frame may be set to 0, 180 or 270 degrees in the case of body layout A, to 0, 90 or 270 degrees in the case of body layout B and 180 or 270 degrees in the case of body layout C.

6.3.5.15 BasicFixture

BasicFixture is a constituent constraint that defines a lowest level frame which specifies an area for laying out content within FixedCompositeBody and CompositeArtwork.

This frame has a fixed position within its superior frame. The dimensions of this frame in the directions orthogonal and parallel to the layout path of the superior frame may be independently either fixed or specified by the sub-parameter "Rule-B".

If the superior frame of a BasicFixture frame is FixedCompositeBody, the layout path of the BasicFixture frame is set to be the same as that of its superior frame.

If the superior frame of a BasicFixture frame is CompositeArtwork, the layout path of the BasicFixture frame is restricted as follows:

- if the layout path of the superior frame is set to 270 degrees, the layout path of BasicFixture may be set to 270 or 180 degrees;
- if the layout path of the superior frame is set to 0 degrees, the layout path of BasicFixture may be set to 0 or 270 degrees;
- if the layout path of the superior frame is set to 180 degrees, the layout path of BasicFixture may be set to 180 or 270 degrees.

6.3.5.16 FormArea

FormArea is a constituent constraint that defines a composite frame which represents a fixed dimensioned area reserved for laying out an illustration consisting of a form.

This area is divided into one or more simple and composite areas. A simple area is one in which content is directly laid out. A composite area is an area which is further divided into one or more simple and composite areas.

This is a variably positioned frame that may be specified as a subordinate to a CompositeFixtureVariable or CompositeFixtureFixed frame.

The dimensions of this frame are fixed in both directions.

The immediate subordinates of a FormArea frame consists of an arbitrary ordered sequence of one or more fixed positioned frames of the following constituent constraints:

- FormEntryArea;
- EntryGroupArea;
- ArrangedContentFixed.

Frames of the type FormEntryArea and ArrangedContentFixed represent simple areas in a form as described above. EntryGroupArea frames represent composite areas.

The layout path of a FormArea frame defaults to 270 degrees and does not affect the layout of subordinate frames.

An example of the layout of a form is shown in Figure 13.

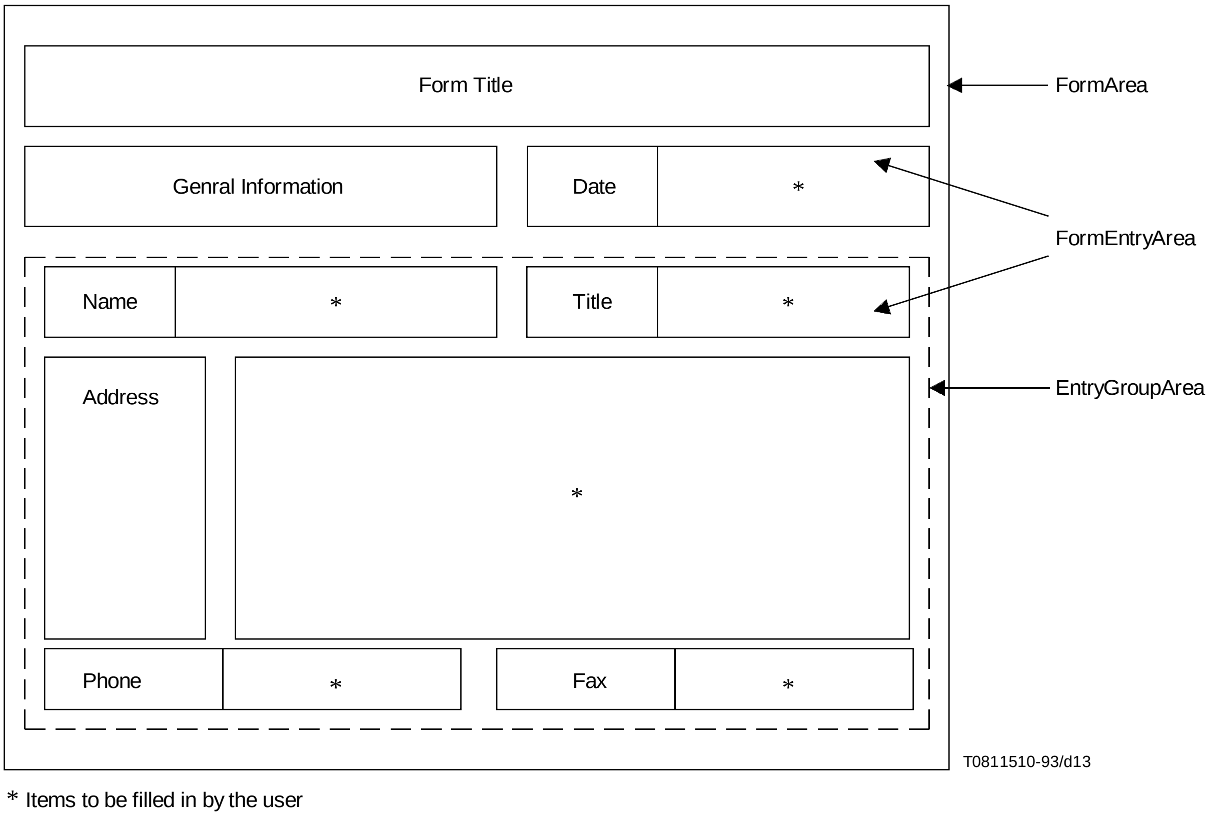


FIGURE 13/T.506

Example of the layout of a form

6.3.5.17 EntryGroupArea

EntryGroupArea is a constituent constraint that represents a composite area within a FormArea frame. The positions and dimensions of this frame are fixed in both directions. The layout path of this frame defaults to 270 degrees.

The immediate subordinates of this frame consist of an arbitrary ordered sequence of one or more fixed positioned frames of the type FormEntryArea.

6.3.5.18 FormEntryArea

FormEntryArea is a constituent constraint that defines a lowest level frame that specifies an area for laying out content within a FormArea frame. This frame may contain character, raster graphics or geometric graphics content.

The position and dimensions of this frame are fixed. Its layout path is 270 degrees regardless of the page layout type.

6.3.5.19 BasicColumn

BasicColumn is a constituent constraint that defines a lowest level frame that specifies an area for laying out content in the form of a column within a CompositeFloat frame. This frame has a variable position within its superior frame.

The dimension of this frame in the direction orthogonal to the layout path of its superior frame may be fixed or specified by the sub-parameter "Rule-B". Its dimension in the direction parallel to the layout path of the superior frame may be fixed, specified by the sub-parameter "Rule-B" or defaulted to its maximum size. Thus, the dimensions of this frame may be allowed to be automatically adjusted so that it contains all the content allocated to it.

The layout path specified for this frame is specified as 270 degrees in the case of body layout A, 0 degrees in the case of body layout B and 180 degrees in the case of body layout C.

6.3.5.20 CompositeColumnVariable

CompositeColumnVariable is a constituent constraint that defines a composite frame which specifies an area representing a column within a SnakingColumns frame. The column is subdivided into different areas to allow different layout requirements to be achieved. For example, this frame can be used to represent a column containing a table and a complex diagram which is embedded in a stream of character content. This is a frame which is variably positioned.

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

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

The immediate subordinates of this frame consist of an arbitrary ordered sequence of frames of the following constituent constraints:

- – BasicFloat;
- – CompositeFloat;
- – CompositeFixtureVariable;
- – TableArea;
- – FootnoteArea;
- – ArrangedContentVariable;
- – SourcedContentVariable.

The layout path for CompositeColumnVariable frames is 270 degrees in the case of body layout A, 0 degrees in body layout B and 180 degrees in body layout C.

All subordinate frames are laid out in normal positioning fill order except FootnoteArea which is laid out in reverse fill order.

6.3.5.21 CompositeColumnFixed

CompositeColumnFixed is a constituent constraint that defines a composite frame which specifies an area representing a column within a SynchronizedColumns frame. The column is subdivided into different areas to allow different layout requirements to be achieved.

The characteristics of this frame are the same as those of CompositeColumnVariable frames, except that this frame has a fixed position, and its dimensions are the same as for ColumnFixed frames.

6.3.5.22 CompositeCommon

CompositeCommon is a constituent constraint that defines a composite frame which specifies an area within the body area that is to contain common content. This area may be subdivided so that different types of common content may be laid out.

This is a frame which has fixed positions and dimensions. It may be specified as a subordinate to a FixedCompositeBody frame.

The subordinates of a CompositeCommon frame may consist of either:

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

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

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

The layout path of a CompositeCommon frame is 270, 0 and 180 degrees in the cases of body layouts A, B and C respectively.

6.3.5.23 Constituents used for laying out tables

This subclause defines the constituents used to support the layout of tables. An overview of the layout features pertaining to tables is given in 6.3.5.23.1 and the subsequent subclauses define the individual constituents provided.

6.3.5.23.1 Overview

A table consists of three main areas:

- a single optional header area which is placed at the top of the table;
- a single optional table label area which is placed immediately below the header area;
- one or more row areas, which are placed in sequence below the header area and table label area.

The table header area is typically used to contain a title or caption that describes the purpose of the table. It is an area which is subdivided into one or more areas, each of which contains common content derived from the logical structure using the logical source mechanism.

The table label area is typically used to contain the labels which relate to the columns in the table. The row areas form the main body of the table and are used to layout the information that belongs to the table. These areas may be simple or composite as described below.

An example of a simple table is shown in Figure 14. In this example, the table label and each row consist of a sequence of areas called cells which are laid out horizontally across the table area. These are examples of simple table label and row areas.

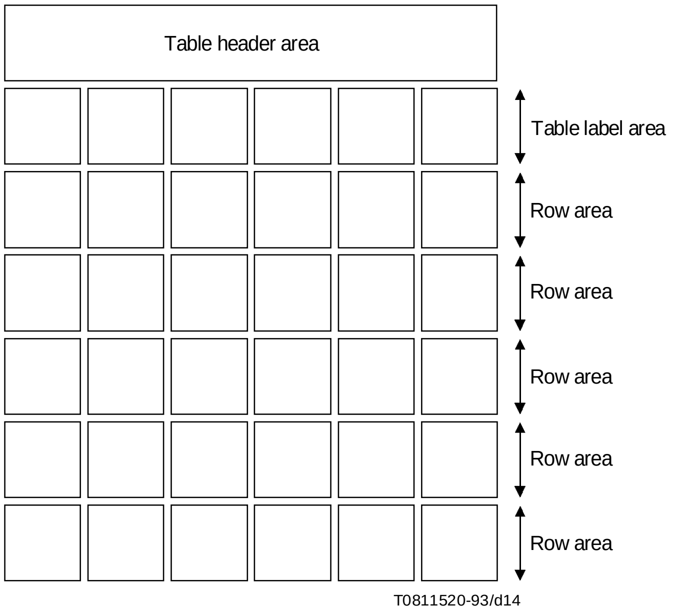


FIGURE 14/T.506

Example of a *simple* table

An example of a more complex table is shown in Figure 15 which illustrates the use of composite table labels and row areas. A composite table label area consists of a sequence of two or more sub-rows, each of which is divided into separate cells. The cells in each row may or may not be the same size to allow a label to refer to a single column or several columns. The sequence of sub-rows may be preceded on the left by a single cell which, for example, can be used to contain a title which refers to the group of sub-rows.

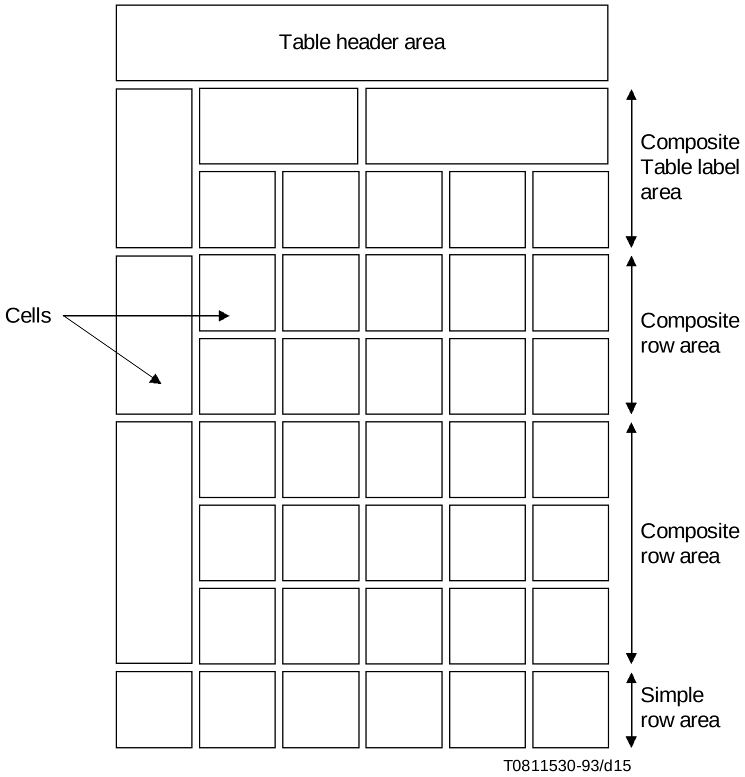


FIGURE 15/T.506

Example of a *complex* table

Composite row areas have the same general structure as composite table label areas. By mixing together different combination of simple and composite table label and row areas, it is possible to obtain a wide range of different table types.

The frames that are used to represent table label and row areas are shown in Figures 16 and 17 respectively. The areas labelled as cells are intended to accommodate content belonging to a single content type. The frames which represent the cells are fixed in position and have a fixed horizontal dimension. However, the vertical dimension of a cell may be specified as variable, so that this dimension is automatically adjusted during the layout process to accommodate all the content allocated to it.

In the case of a composite table label or row, the containing frame (i.e. the LabelComponent or SubRow frame respectively) also has a dimension which is variable in the vertical direction. Thus, the dimension of this frame may be automatically adjusted during the layout process so that it is large enough to accommodate the largest cell in that row.

Also, the containing frames (i.e. the CompositeTableLabel, TableLabel, SubRowGroup and SubRow frames) may all be specified as having a variable vertical dimension, and, therefore, the vertical dimension of each table label and row area may be automatically adjusted to take into account all the information that is to be laid out in these areas.

The frames which specifies the complete table area (i.e. the TableArea frame), which is not shown in the figures, also may have a variable vertical dimension.

The mechanism by which content is allocated to the cells in a table is described in 6.4.1.3.8.

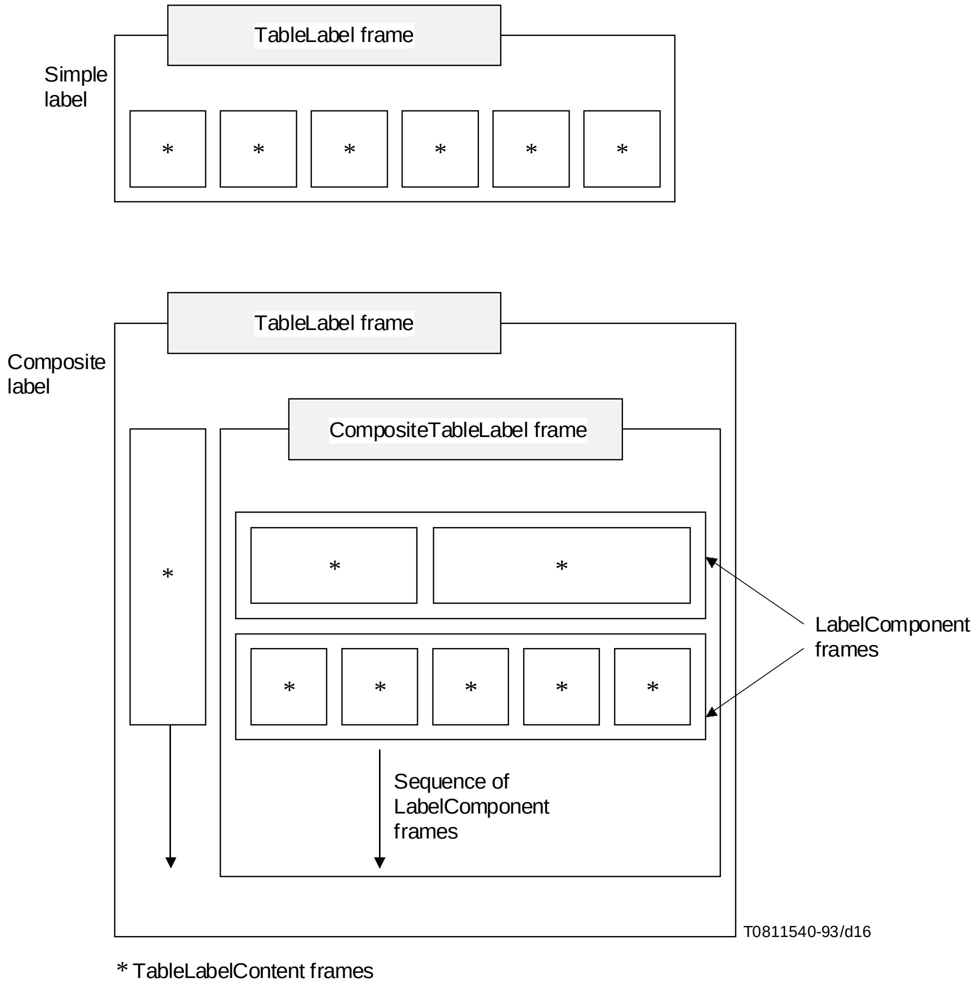


FIGURE 16/T.506

Frames used to represent table labels

6.3.5.23.2 TableArea

TableArea is a constituent constraint that defines a composite frame that is used to specify an area reserved for the layout of a table. This constituent constraint may be specified as a subordinate to the following constituent constraints:

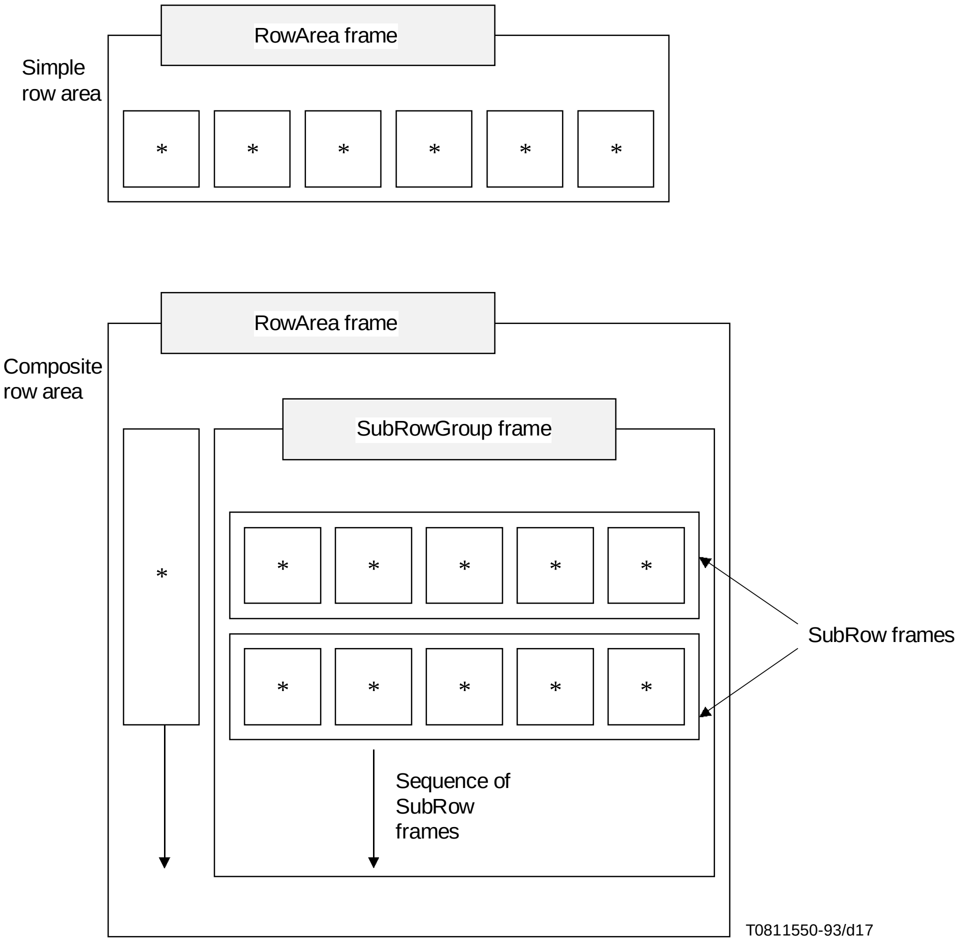
- VariableCompositeBody;
- CompositeFloat;
- CompositeColumnVariable;
- CompositeColumnFixed.

This is a frame that has a variable position. Its dimension in the direction orthogonal to the layout path of the superior frame is fixed. Its dimension in the direction parallel to the layout path is fixed or specified by the sub-parameter "Rule-B". Its layout path is specified as 270 degrees.

The immediate subordinates of this constituent constraint consist of an optional TableHeader, followed by an optional TableLabel, which is followed by a sequence of one or more constituent constraints of the types RowArea and an optional TableLabel.

6.3.5.23.3 TableHeader

TableHeader is a constituent constraint that specifies a composite frame that specifies an area within a TableArea frame that is typically used to present the header information associated with a table.



* Frames of the type Cell

FIGURE 17/T.506

Frames used to represent table rows

This is a frame whose position is variable. Its dimension in the direction orthogonal to the layout path of the superior frame is fixed. Its dimension in the direction parallel to the layout path is fixed or specified by the sub-parameter "Rule-B".

The immediate subordinates of this constituent constraint consist of a sequence of constituent constraints of the type SourcedContentFixed. Hence, the content laid out in a TableHeader frame is derived from logical constituent constraints of the type CommonContent.

6.3.5.23.4 TableLabel

TableLabel is a constituent constraint that defines a composite frame that specifies an area within a TableArea frame that is used for laying out labelling information relating to the columns of information in the table.

This is a frame whose position is variable. Its dimension in the direction orthogonal to the layout path of the superior frame is fixed. Its dimension in the direction parallel to the layout path is fixed or specified by the sub-parameter "Rule-B".

The immediate subordinates of this constituent constraint consist of either:

a) a sequence of one or more constituent constraints of the type TableLabelContent; or

b) a sequence of a constituent constraint of the type TableLabelContent, and a constituent constraint of the type CompositeTableLabel.

6.3.5.23.5 CompositeTableLabel

CompositeTableLabel is a constituent constraint that defines a composite frame that specifies an area with a TableLabel frame for laying out a composite table label.

This is a frame whose position is fixed. Its dimension in the direction orthogonal to the layout path of its superior frame is fixed. Its dimension in the direction parallel to the layout path is fixed, specified by the sub-parameter "Rule-B" or defaults to the maximum size allowed.

The immediate subordinates of this constituent constraint consist of a sequence of one or more constituent constraints of the type LabelComponent.

6.3.5.23.6 LabelComponent

LabelComponent is a constituent constraint that defines a composite frame that specifies an area within a CompositeTableLabel frame for laying out a row of labels within a table header.

This is a frame whose position is variable. Its dimension in the direction orthogonal to the layout path of the superior frame is fixed. Its dimension in the direction parallel to the layout path is fixed, specified by the sub-parameter "Rule-B" or defaults to the maximum size allowed.

The immediate subordinates of this frame consist of a sequence of constituent constraints of the type TableLabelContent.

6.3.5.23.7 TableLabelContent

TableLabelContent is a constituent constraint that defines a lowest level frame that defines an area within a TableLabel or LabelComponent frame that is used for laying out header information that relates to one or more columns in a table. Character, raster graphics or geometric graphics content may be allocated to this frame.

This is a frame whose position is fixed. Its dimension in the direction orthogonal to the layout path of the superior frame is fixed. Its dimension in the direction parallel to the layout path is fixed or defaults to the maximum size allowed.

The content of a frame of this type is derived from a logical constituent constraint of the type CommonContent, using the logical source mechanism.

6.3.5.23.8 RowArea

RowArea is a constituent constraint that defines a composite frame that specifies an area within a TableArea frame used for laying out a row of entries in a table.

This is a frame whose position is variable. Its dimension in the direction orthogonal to the layout path of the superior frame is fixed. Its dimension in the direction parallel to the layout path is fixed or specified by the sub-parameter "Rule-B".

The immediate subordinates of this constituent constraint consist of either:

a) a sequence of one or more constituent constraints of the type Cell; or

b) a sequence of a single constituent constraint of the type Cell, and a constituent constraint of the type SubRowGroup.

6.3.5.23.9 SubRowGroup

SubRowGroup is a constituent constraint that defines a composite frame that specifies an area within a RowArea frame for laying out a composite row of entries in a table.

This is a frame whose position is fixed. Its dimension in the direction orthogonal to the layout path of its superior frame is fixed. Its dimension in the direction parallel to the direction of the layout path is fixed, specified by the sub-parameter "Rule-B" or defaults to the maximum size allowed.

The immediate subordinates of this constituent constraint consist of a sequence of one or more constituent constraints of the type SubRow.

6.3.5.23.10 SubRow

SubRow is a constituent constraint that defines a composite frame that specifies an area within a SubRowGroup frame for laying out a sub-row of entries within a composite row in a table.

This is a frame whose position is variable. Its dimension in the direction orthogonal to the layout path of the superior frame is fixed. Its dimension in the direction parallel to the layout path is fixed, specified by the sub-parameter “Rule-B” or defaults to the maximum size allowed.

The immediate subordinates of this frame consist of a sequence of constituent constraints of the type Cell.

6.3.5.23.11 Cell

Cell is a constituent constraint that defines a lowest level frame that specifies an area within a RowArea or SubRow frame for laying out an entry in a table.

This is a frame whose position is fixed. Its dimension in the direction orthogonal to the layout path of the superior frame is fixed. Its dimension in the direction parallel to the layout path is fixed, specified by the sub-parameter “Rule-B” or defaults to the maximum size allowed.

The content of frames of this type is derived from logical constituent constraints of the type EntryElement.

6.3.6 Header and footer area characteristics

6.3.6.1 General characteristics

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

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

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

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

6.3.6.2 BasicHeader and BasicFooter

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

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

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

6.3.6.3 CompositeHeader and CompositeFooter

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

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

The subordinates of these frames may consist of either:

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

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

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

6.3.6.4 SourcedContentVariable

A *SourcedContentVariable* frame is a constituent constraint that defines a lowest level frame that contains common content derived from the generic logical structure. This frame is variably positioned and is used for the positioning content which is generated during the layout process, such as a character sequence containing a page number, a chapter title, etc.

This frame may be placed in the body area as well as the header or footer area:

- When this frame is in the header or footer area, it is the immediate subordinate of the frame of the constituent constraint type CompositeHeader or CompositeFooter.
- When this frame is in the body area, it is the immediate subordinate of the frame of the constituent constraint type VariableCompositeBody, CompositeColumnVariable, CompositeColumnFixed, CompositeCommon, SnakingColumns or CompositeFloat.

  SnakingColumns are used to place common contents in the multi-column format. CompositeFloat is used to place common contents along side a figure or a form.

The attribute "logical source" shall be specified for this frame to indicate the particular instance of the constituent constraint CommonContent which contains the content to be laid out.

When this frame is a subordinate of CompositeHeader or CompositeFooter:

- The layout path of the frame is:
  - 270 degrees for H/F layouts A1 and A2;
  - 180 degrees for H/F layouts B1 and B2 (see 6.3.4.5 and the comment in 7.4.3.21).
- The horizontal dimension of the frame is:
  - either fixed or specified by sub-parameter "maximum-size" for H/F layouts A1, A2 and B2;
  - either fixed or specified by sub-parameter "Rule-B" for H/F layout B1.
- The vertical dimension of the frame is:
  - either fixed or specified by sub-parameter "Rule-B" for H/F layouts A1 and A2;
  - either fixed or specified by sub-parameter "maximum-size" for H/F layout B1;
  - only fixed in the case of H/F layout B2.

When this frame is a subordinate of VariableCompositeBody, CompositeColumnVariable, CompositeColumnFixed or CompositeCommon:

- the layout path of the frame is 270, 0 or 180 degrees for body layouts A, B or C respectively (see 6.3.4.5 and the comment in 7.4.3.21);
- the dimension of the frame in the direction orthogonal to the layout path of the superior frame is either fixed or specified by sub-parameter "maximum-size";
- the dimension of the frame in the direction parallel to the layout path of the superior frame is either fixed or specified by sub-parameter "Rule-B".

When this frame is a subordinate of SnakingColumns:

- the layout path of the frame is 270, 0 or 180 degrees for body layouts A, B or C respectively (see 6.3.4.5 and the comment in 7.4.3.21);
- the dimension of the frame in the direction orthogonal to the layout path of the superior frame is specified by either sub-parameter "Rule-B" or sub-parameter "maximum-size", except that only use of sub-parameter "maximum-size" is permitted for body layout C;
- the dimension of the frame in the direction parallel to the layout path of the superior frame is fixed.

When this frame is a subordinate of CompositeFloat:

- the layout path of the frame is 270, 0 or 180 degrees for body layouts A, B or C respectively (see 6.3.4.5 and the comment in 7.4.3.21);

- the dimension of the frame in the direction orthogonal to the layout path of the superior frame is either fixed, specified by sub-parameter “Rule-B” or specified by sub-parameter “maximum-size”;
- the dimension of the frame in the direction parallel to the layout path of the superior frame is either fixed or specified by sub-parameter “Rule-B”.

6.3.6.5 ArrangedContentVariable

An *ArrangedContentVariable* frame is a constituent constraint that defines a lowest level frame that contains pre-defined common content contained in the generic layout structure. This frame is variably positioned, and its dimensions are fixed.

This frame references one or more blocks of type GenericBlock (see 6.3.8) which contain the content to be laid out in this frame. Thus, this frame is typically used when it is required to layout pre-determined common content.

6.3.6.6 SourcedContentFixed

A *SourcedContentFixed* frame is a constituent constraint that defines a lowest level frame that contains common content derived from the generic logical structure. This frame has a fixed position and dimensions.

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

This frame may be placed in the body area as well as the header or footer area:

- when this frame is in the header or footer area, the frame is the immediate subordinate of the frame of the constituent constraint type CompositeHeader or CompositeFooter;
- when this frame is in the body area, the frame is the immediate subordinate of the frame of the constituent constraint type FixedCompositeBody, CompositeCommon or SynchronizedColumns.

When this frame is a subordinate of CompositeHeader or CompositeFooter:

- the layout path of this frame is 270 degrees for H/F layouts A1 and A2;
- the layout path of this frame is 180 degrees for H/F layouts B1 and B2 (see 6.3.4.5).

When this frame is a subordinate of FixedCompositeBody, CompositeCommon or SyncronizedColumns:

- the layout path of this frame is 270, 0 or 180 degrees for body layouts A, B or C respectively (see 6.3.4.5).

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

6.3.6.7 ArrangedContentFixed

An *ArrangedContentFixed* frame is a constituent constraint that defines a lowest level frame that contains pre-defined common content derived from the generic layout structure. The position and dimensions of this frame are fixed.

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

6.3.7 GenericBlock and SpecificBlock

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

Object classes of the type GenericBlock may occur in the generic layout structure referenced by the attribute “generator for subordinates” of object classes of the types ArrangedContentVariable and ArrangedContentFixed. When the layout process is performed to produce a document in formatted processable form, equivalent blocks may occur in the specific layout structure.

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

6.4 Document layout characteristics

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

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

6.4.1 Flow controls

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

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

a) layout in a nominated page set;

b) starting a new page set.

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

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

- Passage;
- NumberedSegment;
- Paragraph;
- Number;
- BodyText;
- BodyRaster;
- BodyGeometric;
- Title;
- Figure;
- NumberedList;
- UnNumberedList;
- DefinitionList.

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

6.4.1.2 Page breaks

This provides the ability to specify that the logical objects derived from a particular logical constituent constraint in a document and all subsequent parts of a document are to be laid out starting at the beginning of a new page. The page specified shall belong to the page set in which the logical objects from the immediate preceding logical constituent constraint is laid out. The specification of a page breaks shall not be used to layout part of a document in a new page set. If a new page set is required, then this shall be explicitly specified as described in 6.4.1.1.

This may be specified for logical objects from the following logical constituent constraints:

- Passage;
- NumberedSegment;

- Paragraph;
- Number;
- BodyText;
- BodyRaster;
- BodyGeometric;
- Title;
- Figure;
- NumberedList;
- UnNumberedList;
- DefinitionList.

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

6.4.1.3 Allocation of content to body areas

6.4.1.3.1 Introduction

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

If the page contains a composite body area, i.e. a VariableCompositeBody or FixedCompositeBody frame, then the content is allocated to subordinate areas in that body area as described below.

The general layout mechanism is described in 6.4.1.3.2. However, particular layout facilities are provided for the layout of logical constituent constraints of the type Table (see 6.4.1.3.8) and the type Figure, which may contain either Artwork or Forms (see 6.4.1.3.7). Also, the layout of footnotes is described in 6.4.1.3.10.

6.4.1.3.2 General mechanism for laying out content in a composite body area

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

Basic logical objects other than those which are within a footnote structure may be specified to be laid out in instances of one or more lowest level frame class. This is done by giving each such basic logical component a value of the attribute "layout category" which corresponds to the value of the attribute "permitted categories" that applies to the lowest level frame in which the content is to be laid out.

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

The use of the attribute "layout category" ensures that if there is insufficient area on one page to lay out all of the content allocated to a particular type of area, the laying out of the content will automatically continue in the same type of area in a succeeding page when possible. Thus content is allowed to flow freely from one page to another when the type of layout used at the end of one page is the same as that at the beginning of the succeeding page. When continuation to the same type of area in a succeeding page is not possible because of conflict with other layout directives or because the attribute "generator for subordinates" of the page class does not allow such choice, backtracking may occur or other type of area may be selected.

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

The attribute "new layout object" may be applied to logical components whenever a change in layout is required. The attribute "new layout object" may specify the identifier or the category name corresponding to the frame class that is required.

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

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

The following subclauses describe the layout mechanism applicable to subordinate areas for each of the frame types listed above.

6.4.1.3.3 Layout into BasicFloat and BasicFixture frames

These are lowest level frames and hence content continues to be directly laid out in these frame types until an occurrence of the attribute “new layout object” causes the layout to proceed starting with an alternative frame class.

6.4.1.3.4 Layout in SnakingColumns frames

A SnakingColumns frame is a composite frame which contains columns represented by lowest level or composite frames.

In the case of lowest frames, all the frames may have the same category name so that content can flow from one frame to the next. That is, a column break will occur naturally when the size of one column reaches the limit imposed by the superior frame and the layout process will continue automatically in the next column.

In the case of composite columns, the subordinate areas are represented by subordinate frames of the types BasicFloat, CompositeFloat, CompositeFixtureVariable, TableArea and FootnoteArea. The frame type into which the constituents are to be laid out is selected using the attribute “new layout object” which indicates the identifier or category name of the required subordinate frame, or is automatically selected according to “layout categories”. Logical constituents will continue to be laid out in the selected frame type until a different frame type is selected. Also, the layout process continues automatically from one column to the next, but column breaks can again be forced as described in 6.4.1.3.2.

6.4.1.3.5 Layout in SynchronizedColumns frames

A SynchronizedColumns frame is a composite frame which contains columns represented by subordinate lowest level or composite frames.

In the case of lowest frames, all the frames are required to have different categories. Hence, the layout of logical objects from the constituents into different columns is controlled by the category name specified for each constituent. The attribute “new layout object” may also be used for this purpose, but this is not necessary.

In the case of composite columns, the subordinate areas are represented by subordinate frames of the types BasicFloat, CompositeFloat, CompositeFixtureVariable, TableArea and FootnoteArea.

The selection of a particular composite column or a particular sub-area within a composite column can be achieved using the attribute “new layout object” which specifies the identifier or category name of the particular frame class required.

6.4.1.3.6 Layout in CompositeFloat frames

This is a composite frame which contains two or more subordinate frames that are laid out side-by-side. The appropriate subordinate frame is chosen according to the category names or chosen using the attribute “new layout object” which specifies the appropriate identifier or category name of the required subordinate frame class.

6.4.1.3.7 Layout of figures

Frames of the types CompositeFixtureVariable and CompositeFixtureFixed are provided specifically for the layout of logical constituent constraints of the type Figure. Similarly, frames of the type CompositeArtwork and FormArea are provided for the layout of logical constituent constraints of the types Artwork and Form respectively.

The schematic diagram in Table 2 shows how the logical constituent constraint Figure and its subordinates are allocated to the frame CompositeFixtureVariable or CompositeFixtureFixed and their subordinates.

Table 2 indicates a mapping between logical constituent constraints and frames, and their respective subordinates. Also, the diagram indicates that this mapping is hierarchical.

TABLE 2/T.506

Layout of figure

Logical constituent constraint	Frame class
Figure.................................... >	CompositeFixtureVariable ou CompositeFixtureFixed
Artwork........................... >	CompositeArtwork
Phrase .... >	BasicFixture
BodyRaster .... >	BasicFixture
BodyGeometric .... >	BasicFixture
Form .... >	FormArea
EntryGroup............... >	EntryGroupArea
EntryElement............ >	FormEntryArea
Number........................... >	BasicFloat
Caption .... >	BasicFloat
Description .... >	BasicFloat
Footnote.................... >	FootnoteArea

For example, Figure is to be laid out into a single instance of the frame CompositeFixtureVariable or CompositeFixtureFixed. A subordinate constituent constraint of the type Artwork is to be laid out in a single instance of a frame of type CompositeArtwork within the specified CompositeFixtureVariable in a CompositeFixtureFixed frame.

Also, each instance of a subordinate Phrase, BodyRaster or BodyGeometric is to be laid out in a single instance of a subordinate frame of the type BasicFixture. BasicFixture frames may overlap to form a composite image.

Similarly, a Form is to be laid out in a single instance of a FormArea frame. Frames subordinate to this FormArea, that is, EntryGroupArea and FormEntryArea frames, will each receive single instances of logical constituent constraints of the type EntryGroup and EntryElement respectively.

This layout mechanism is achieved by specifying the attribute “new layout object” for Figure with a value indicating the identifier of the appropriate frame classes in which that constituent is to be laid out.

The constituent constraints Number, Caption and Description (and their subordinates in the case of Caption and Description) are laid out in frames of the type BasicFloat. This is achieved automatically according to category names or explicitly using the attribute “new layout object” which indicates the identifier or category name of the frame class required. More than one instance of the constituent constraints Caption, Description and Number may be laid out in a particular BasicFloat frame.

Frames of the type FootnoteArea may be generated within a CompositeFixtureVariable or CompositeFixtureFixed to accommodate instances of the logical constituent constraint Footnote which occurs as a subordinate of Phrase, Caption or Description.

6.4.1.3.8 Layout of tables

Frames of the type TableArea are provided specifically for the layout of logical constituent constraints of the type Table.

Table 3 illustrates the relationships between the logical constituent constraint Table and its subordinates, and the frames used to lay out these constituent constraints.

TABLE 3/T.506

Layout of tables

Logical constituent constraint	Frame class
Table ....>	TableArea
Row ....>	RowArea
EntryElement...................>	Cell
TableComponent ....>	SubRowGroup
RowComponent.........>	SubRow
EntryElement ......>	Cell
CommonContent <	TableHeader
CommonContent <	TableLabel

Table 3 indicates that there is a hierarchical mapping between logical constituent constraints and their corresponding frames. For example, each Row is to be laid out in a separate frame of the type RowArea. Each TableComponent that is subordinate to that Row shall be laid out in a specific SubRowGroup that is subordinate to the RowArea indicated.

The layout mechanism is achieved for the logical constituent constraints Table, Row and RowComponent by specifying the attribute “new layout object” with a value indicating the identifier of the required frame class of the type TableArea, RowArea and SubRow respectively.

For the logical constituent constraint EntryElement, the layout mechanism is achieved by one of the following:

– If the attribute “generator for subordinates” of the superior (RowArea or SubRow) of the affected EntryElement is constructed using SEQuence, the attribute “new layout object” is used to specify a Cell into which the contents is laid out. The attribute value specifies the identifier of the required frame class in the EntryElement.

– If the attribute “generator for subordinates” of the superior (RowArea or SubRow) of the affected EntryElement is constructed by REPetition, the value of the attribute “new layout object” indicates a category name to be used for the EntryElement. In this case, the category name shall be specified in the attribute “layout category” for EntryText, EntryRaster or EntryGeometric, and in the attribute “permitted categories” for the Cell into which the contents is laid out.

In the case of TableComponent, the attribute “layout object class” is used to specify that this logical constituent constraint is to be laid out in a SubRowGroup frame.

This mechanism allows a table to be laid out such that it is split over two or more successive frames or pages. A split may occur at the boundary of a RowArea frame, or such that a RowArea frame is split over two successive frames or pages. A split cannot occur within a SubRowGroup frame.

When such a split does occur, then the TableHeader and TableLabel frames are repeated at the top of each frame of page in which the table is continued.

The content allocated to the frames TableHeader and TableLabel is derived from logical constituent constraints of the type CommonContent in the generic logical structure using the logical source mechanism.

6.4.1.3.9 Layout of forms

Frames of the type FormArea are provided specifically for the layout of logical constituent constraints of the type Form.

Table 4 illustrates the relationships between the logical constituent constraint Form and its subordinates, and the frames used to lay out these constituent constraints.

TABLE 4/T.506

Layout of forms

Logical constituent constraint	Frame class
Form..........................>	FormArea
EntryGroup................>	EntryGroupArea
EntryElement..................>	FormEntryArea

Table 4 indicates there is a hierarchical mapping between logical constituent constraints and the corresponding frames, and their respective subordinates.

The layout mechanism is achieved by specifying for logical constituent constraints the attribute “layout object class” which indicates the object class identifier of an appropriate frame in accordance with the above diagram. This mechanism does not allow frames of the type FormArea to be split over two or more superior frames.

The content associated with the logical constituent constraint EntryElement is specified by one of the constituent constraints EntryText, EntryRaster or EntryGeometric. The layout of this content is controlled by the layout directives “offset” and “block alignment”.

6.4.1.3.10 Layout of footnotes

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

This constraint is specified by means of category names. That is, the logical constituent constraints of the types FootnoteNumber and FootnoteText, and layout constituent constraints of the type FootnoteArea are all required to have the category name 'Footnote' or 'Footnote <n>'.

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

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

6.4.1.4 Allocation of content to header and footer areas

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

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

6.4.2 Layout of document content

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

6.4.2.1 Margins

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

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

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

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

These margins are defined in relationship to the layout path specified for the frame into which the content is to be laid out (see Figure 18).

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

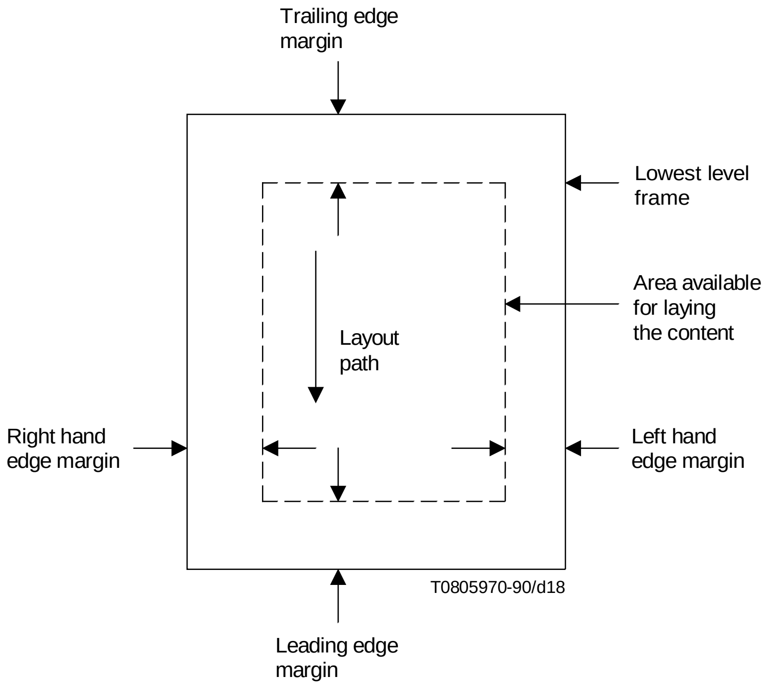


FIGURE 18/T.506

Specification of margins

6.4.2.2 Separation

Leading separation is the minimum distance between one basic logical object and the next one, if any, when they are laid out. Trailing separation is the minimum distance between one basic logical object and the previous one, if any, when they are laid out. Both may be specified for basic logical components of any constituent constraint types. These distances are specified in SMUs by the attribute “separation”. If no value is specified, then the minimum distance is assumed to be 0 SMUs.

6.4.2.3 Indivisibility

Indivisibility provides the means to specify whether or not a basic or composite logical object derived from a constituent constraint is allowed to be split over more than one page or over more than one area within a page. It may be specified for constituent constraints of the types Passage, NumberedSegment, Number, Title, Paragraph, Caption, Phrase, Reference, Description, ListItem, ListTerm, FootnoteText, ReferencedContent, FootnoteBody, Artwork, EntryElement, Row, RowComponent, Footnote, Figure, Table, UnNumberedList, NumberedList, DefinitionList, FootnoteReference and BodyText. The attribute “indivisibility” is used to specify this feature.

6.4.2.4 Same layout object

Same layout object provides the means to specify that the start of the content associated with a logical object and the end of the content associated with the previous logical object are to be laid out within a single layout object. This may be specified for logical objects of the types Passage, NumberedSegment, Title, Caption, Number, Paragraph, Phrase, Footnote, FootnoteBody, Figure, FootnoteReference, ReferencedContent, Reference, Description, Table, NumberedList, UnNumberedList, DefinitionList, ListItem, ListTerm, BodyText, BodyRaster and BodyGeometric. The attribute “same layout object” is used to specify this feature.

6.4.2.5 Concatenation

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

6.4.2.6 Block alignment

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

This feature may only be specified using the attribute “block alignment” for constituent constraints of the types BodyText, EntryText, Number, CommonNumber, PageNumber, TableNumber, FootnoteNumber, FootnoteText, FootnoteReference, CommonReference, ReferencedContent, CurrentInstance, and CommonText, and when they contain formatted character content, BodyRaster, and BodyGeometric, EntryRaster, EntryGeometric, CommonRaster and CommonGeometric.

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

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

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

a) It is not possible to specify that certain logical parts of a document are to be allocated to a given page set or that a part of a document is to be laid out starting in a new page set, as defined in 6.4.1.1.

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

c) The logical parts of the document that are intended to be laid out in the body area and in the header/footer areas of each page may be distinguished from each other by means of application comments specified for them (see 6.6.4). An exception is that it is not possible to distinguish whether a particular portion of common content is to be placed in a header or a footer area (or both).

d) It is not possible to indicate the type of layout area to be used to layout each logical constituent in the body part of a document. That is, it is not possible to indicate whether single column or multiple column areas are to be used (see 6.4.1.3). This shall be decided by the receiver.

e) Footnotes within the body part of a document may be distinguished by use of the attribute “application comments”. Footnotes are intended to be read and laid out separately from the other logical constituents of the body part (see 6.4.1.3). However, it is the responsibility of the receiver to decide how footnotes are laid out.

f) Margins, separation, indivisibility, same layout object, concatenation and block alignment, as defined in 6.4.2, may all be specified. Only one restriction applies. Indivisibility (see 6.4.2.3) may be assumed to specify that a logical constituent constraint is not to split over more than one page, but indivisibility shall not be specified for other types of layout areas, such as single or multiple column areas.

6.5 Content layout and imaging characteristics

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

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

6.5.1 Character content

6.5.1.1 Introduction

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

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

- graphic character sets;
- graphic character subrepertoire;
- code extension announcers;
- line spacing;
- character spacing;
- character path;
- line progression;
- character orientation;
- graphic rendition, including the parameter values: default rendition, increased intensity (bold), italicized, underlined, crossed-out, slowly blinking, rapidly blinking, negative image, positive image, primary font, 1st alternative font, 2nd alternative font, 3rd alternative font, 4th alternative font, 5th alternative font, 6th alternative font, 7th alternative font, 8th alternative font, 9th alternative font, doubly underlined, normal intensity, not italicized, decreased intensity, not underlined, not blinking, not crossed-out;
- line layout table;

- indentation;
- alignment;
- first line offset;
- itemization;
- widow size;
- orphan size;
- character fonts;
- kerning offset;
- pairwise kerning;
- proportional line spacing;
- formatting indicator;
- initial offset.

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

6.5.1.2 Character content architecture class

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

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

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

6.5.1.3 Character repertoires

The basic character repertoire supported by this Recommendation is composed of the 94 characters of ISO-IR6 (the IRV of ISO 646 revised 1991) plus the character space.

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

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

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

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

If the character set defined in ISO 6937-2 with or without Addendum 1 is designated and invoked, then the use of any of its subrepertoires registered according to ISO 7350 may be specified using the presentation attribute “graphic character subrepertoire”. All subrepertoires are non-basic and their use shall be indicated in the document profile. The subrepertoire shall not be changed within a content portion.

NOTES

1 The basic character repertoire supported by this Recommendation is not the standard default value specified in CCITT Rec. 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.

2 Revised CCITT Recs. T.50 and T.51 and new CCITT Rec. T.52 are under preparation. CCITT Recs. T.50 and T.51 are intended to be completely compatible with ISO 646 revised 1991 (ISO-IR6) and ISO 6937 (under revision) respectively.

6.5.1.4 Code extension techniques

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

a) *G0 set* – Only ISO-IR6 (the IRV of ISO 646 revised 1991), ISO-IR2 (the primary set of ISO 6937-2), or any other version of ISO 646 (revised 1991) may be designated for this set; these graphic character sets may only be invoked in GL.

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

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

   – LS0, to invoke the G0 set into GL;

   – LS1R, to invoke the G1 set into GR;

   – LS2R, to invoke the G2 set into GR;

   – LS3R, to invoke the G3 set into GR;

   – SS2, to invoke one character from the G2 set into GL;

   – SS3, to invoke one character from the G3 set into GL;

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

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

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

The code extension techniques allowed are illustrated in Figures 19 and 20.

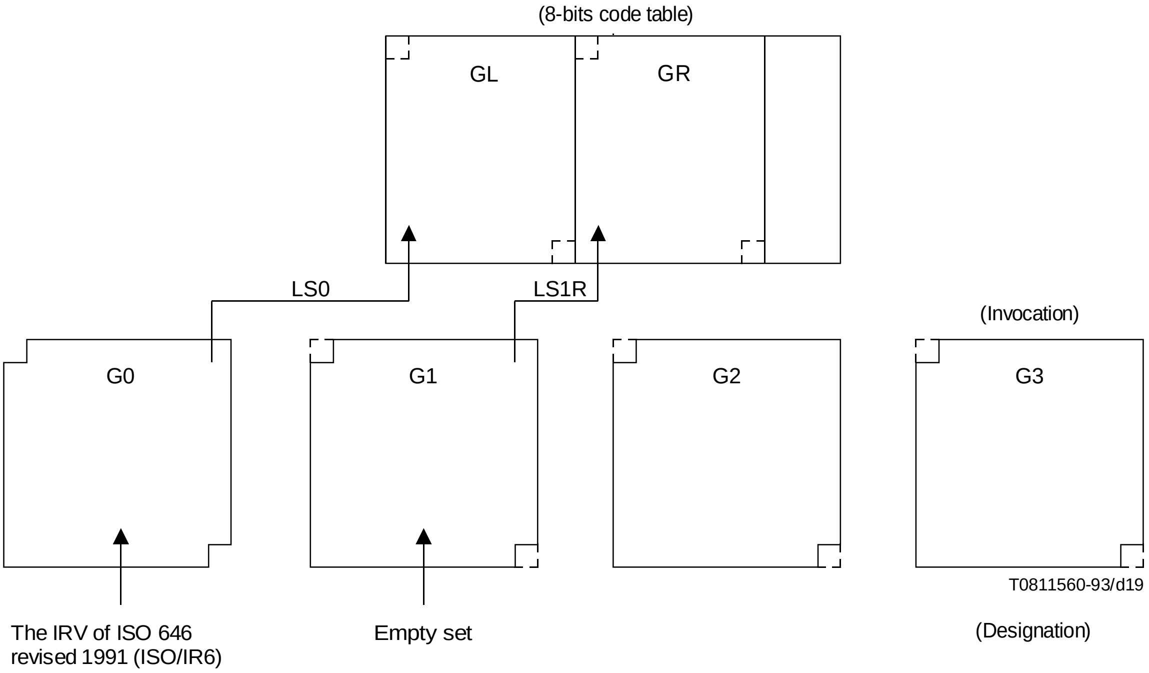


FIGURE 19/T.506

Code extension features (basic case)

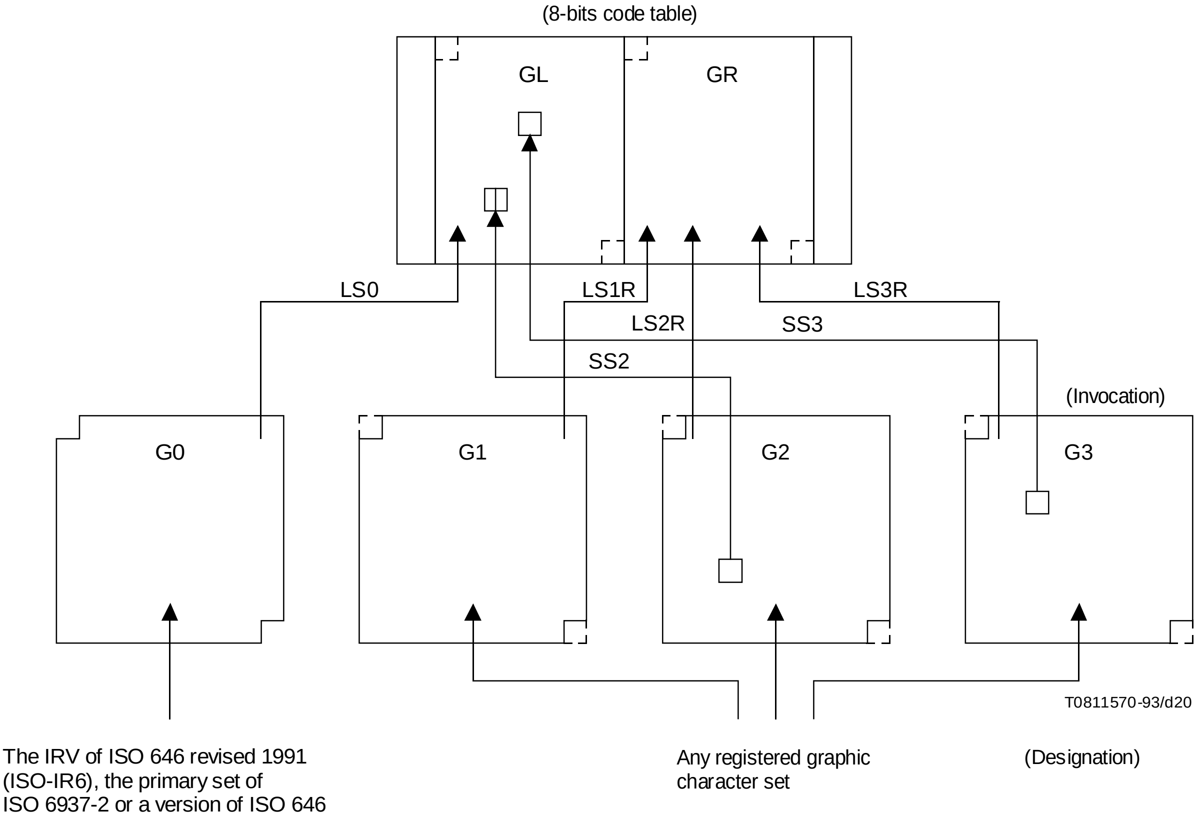

(8-bits code table)
GL
GR
LS0
LS1R
LS3R
LS2R
SS3
SS2
(Invocation)
G0
G1
G2
G3
T0811570-93/d20
The IRV of ISO 646 revised 1991 (ISO-IR6), the primary set of ISO 6937-2 or a version of ISO 646
Any registered graphic character set
(Designation)

FIGURE 20/T.506

Code extension features (all possible cases)

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

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

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

6.5.1.5 Line spacing

Any value of line spacing may be specified. Values of 100, 150, 200, 300, 400 and 600 BMUs are basic; the use of any other value in a document is non-basic and shall be indicated in the document profile.

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

6.5.1.6 Character spacing

Any value of character spacing may be specified as basic values.

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

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

6.5.1.7 Character path and line progression

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

For body layout A, only horizontal writing may be used in the body area. Thus, the character path and line progression are specified either as 0 and 270 degrees respectively or 180 and 90 degrees respectively.

For body layout B, again only horizontal writing may be used in the body area. However, in this case the content in the body area is presented for viewing with the page in landscape orientation and the content in the header and footer areas is presented for viewing with the page in portrait orientation. Thus, for body layout B, in the body area the character path and line progression are specified either as 90 and 270 degrees respectively or 270 and 90 degrees respectively.

For body layout C, only vertical writing may be used in the body area. Thus, the character path and line progression are specified as 270 and 270 degrees respectively.

With regard to the header and footer areas, if these areas are placed above and below the body area, as shown in Figure 1, then only horizontal writing is allowed in these areas. Thus, in this case, the character path and line progression are specified either as 0 and 270 degrees respectively or 180 and 90 degrees respectively (see Figure 2).

If the header and footer areas are placed to the left and right of the body area, then only vertical writing is allowed in these areas. Thus, in this case, the character path and line progression are specified as 270 and 270 degrees respectively (see Figure 4).

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

6.5.1.8 Character positioning controls

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

The active position of a character can be moved forward or backward along the direction of the character path using the control functions character position backward (HPB) and character position relative (HPR).

The spacing between characters can be increased or decreased using the control functions set additional character spacing (SACS) and set reduced character spacing (SRCS) respectively. Also, the width of the character SPACE can be set using the control function set SPACE width (SSW).

The control functions HPB, HPR, SACS, SRCS and SSW shall only be specified in formatted or formatted processable character content; any parameter value may be specified.

6.5.1.9 Character orientation

The character orientation may be specified as 0, 90, 180 or 270 degrees.

The orientations of 0, 90, and 180 degrees are usually applied depending on whether vertical or horizontal writing is used. When horizontal writing is used, characters may only be orientated at 0 degrees. When vertical writing is used, characters may be orientated at 90 or 180 degrees.

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

6.5.1.10 Emphasis

The following modes of emphasising graphic characters may be distinguished:

- default rendition;
- normal intensity;
- increased intensity (bold);
- italicized;
- not italicized;
- underlined;
- doubly underlined;
- not underlined;
- crossed-out;
- not crossed-out;
- slowly blinking;
- rapidly blinking;
- not blinking;
- negative image;
- positive image.

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

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

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

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

6.5.1.11 Tabulation

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

a) The tabulation position relative to the position of that margin in the direction opposite to the character path.

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

   - start aligned;
   - end aligned;
   - centred;
   - aligned around.

These alignment qualifiers are defined in CCITT Rec. T.416 | ISO 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 may be specified to be applicable to the content associated with a basic component. No limit is placed on the number of tabulation stops that may be specified within a given set.

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

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

6.5.1.12 Indentation

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

Indentation acts as 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 shall not be changed within a content portion.

6.5.1.13 Alignment

This feature is concerned with how the first and last characters on each line of character content 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 Rec. T.461 | ISO 8613-6.

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

6.5.1.14 First line format

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

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

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

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

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

6.5.1.15 Widow and orphan sizes

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

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

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

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

6.5.1.16 Fonts

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

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

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

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

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

6.5.1.17 Reverse character strings

Bi-directional writing is supported by this Recommendation. Hence, a string of characters in a content portion associated with a basic component may be specified to be imaged in the reverse direction of the immediately preceding character string. Such strings may be specified by the control function start reverse string (SRS) as defined in CCITT Rec. 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.

6.5.1.18 Kerning offset

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

6.5.1.19 Proportional line spacing

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

6.5.1.20 Superscripts and subscripts

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

6.5.1.21 Line breaks

The control functions break permitted here (BPH) and no break here (NBH) may be inserted in processable or formatted processable form character content to indicate where line breaks may occur or may not occur respectively when the content is laid out.

6.5.1.22 Substitution of characters

The control function substitute (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 Recommendation.

6.5.1.23 Initial point

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

6.5.1.24 Use of control functions

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

SHS	Select character spacing (allowed parameter values: 0, 1, 2, 3, 4);
SCS	Set character spacing (allowed parameter values: any);
SVS	Select line spacing (allowed parameter values: any);
SLS	Set line spacing (allowed parameter values: any);
SGR	Select graphic rendition (allowed parameter values: any);
STAB	Selective tabulation (allowed parameter values: any);
SRS	Start reverse string (allowed parameter values: any);
VPB	Line position backward (allowed parameter values: any);
VPR	Line position relative (allowed parameter values: any);
PLD	Partial line down;
PLU	Partial line up;
BPH	Break permitted here;
NBH	No break here;
JFY	No justify;
SUB	Substitute;
SP	Space;
CR	Carriage return;
LF	Line feed;
SOS	Start of string;
ST	String terminator;
GCC	Graphic character composition (allowed parameter values: any);
IGS	Identify graphic subrepertoire (allowed parameter values: any);
HPB	Character position backward (allowed parameter values: any);
HPR	Cracter position relative (allowed parameter values: any);
SACS	Set additional character spacing (allowed parameter values: any);

SRCS	Set reduced character spacing (allowed parameter values: any);
SSW	Set SPACE width (allowed parameter values: any);
PTX	Parallel text;
	Code extension control functions (see 6.5.1.4).

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

6.5.1.25 Formatting the content

The attribute "formatting indicator" may be specified for particular basic objects that are conformant with this Recommendation.

The effect is to provide for transmission of formatted (or formatted processable) objects for which the precise placement of individual characters has been fully computed, and all necessary control functions included. The implication is that most operations normally performed by the imaging processing in handling formatted character text will be rendered unnecessary.

To make use of this, the imaging process of the recipient shall operate with a font containing metrics identical to that used by the originator.

6.5.1.26 Graphic character composition

A string of two or more characters may be combined to form a single character using the control function graphic character composition (GCC). Parameter values 0, 1 and 2 may be specified.

6.5.1.27 Parallel text

A string of characters in a content portion associated with a basic component may be specified to be imaged in parallel with another character string. Typical example is "ruby" in the Japanese language.

In processable and formatted processable content, parallel text may be specified by the control function parallel text (PTX).

In formatted content, the control functions character position relative (HPR), character position backward (HPB), line position relative (VPR) and line position backward (VPB) may be used to specify parallel text. These control functions may also be present in formatted processable content provided that they are contained within strings delimited by the control functions start of string (SOS) and string terminator (ST).

6.5.1.28 Pairwise kerning

Pairwise kerning may be specified to take place during the layout process using the attribute "pairwise kerning". This process depends upon the font used and the modification applied to the positions of characters depends on the kerning information in the font attributes. Pairwise kerning shall only be carried out if a variably spaced font is used; the attribute "pairwise kerning" is ignored if a constant spaced font is used.

6.5.2 Raster graphics content

6.5.2.1 Introduction

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

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

- type of coding;
- compression;
- pel spacing;

- spacing ratio;
- image dimensions;
- pel path;
- line progression.

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

6.5.2.2 Raster graphics content architecture

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

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

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

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

6.5.2.3 Raster graphics encoding methods

The content may be encoded in accordance with the encoding schemes defined in Recommendations T.4 and T.6. In the case of T.4, either the one-dimensional or two dimensional encoding scheme may be used. Also the bitmap encoding scheme defined in CCITT Rec. T.417 | ISO 8613-7 may be used. All these forms of encoding may be used in a single document, and all are basic. Uncompressed mode of encoding may also be used, but as a non-basic feature.

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

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

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

6.5.2.4 Pel path and line progression

The pel path direction may be specified as 0, 90, 180 or 270 degrees. The line progression direction may be specified as 90 or 270 degrees.

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

6.5.2.5 Clipping

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

6.5.2.6 Pel spacing

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

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

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

NOTES

1 The basic pel spacing values listed above are equivalent to resolutions of 75, 100, 150, 200, 240, 300, 400, 600 and 1200 pels per 25.4 mm respectively when the SMU is interpreted as 1/1200 inch.

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

6.5.2.7 Spacing ratio

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

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

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

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

6.5.2.8 Image dimensions

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

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

6.5.3 Geometric graphics content

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

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

6.6 Miscellaneous features

6.6.1 Resource documents

Object classes of the types BodyText, BodyRaster, BodyGeometric, CommonText, CommonRaster, CommonGeometric, EntryText, EntryRaster, EntryGeometric and GenericBlock may refer to corresponding constituent constraints in a resource generic-document.

The constituent constraints in the resource document may refer to content portions and to layout and presentation styles that are contained within the resource document. The constituent constraints listed above are the only ones that are allowed to be referenced from another document via the resource attribute: however, generic-documents used as resource documents may contain any combination of generic constituent constraints which is conformant to this document application profile.

6.6.2 External documents

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

6.6.3 Border

Border may be specified for all the frame types defined in 6.3.5, 6.3.6 and 6.3.7 using the attribute "border". All the features of borders specified in CCITT Rec. T.412 | ISO 8613-2 may be specified as basic values. Border may also be specified in presentation styles.

6.6.4 Application comments

Specification of the attribute "application comments" is mandatory for all object classes contained in a document that conforms to this Recommendation. Specification and use of this attribute is optional.

This attribute is structured so that it contains two fields. The first field is mandatory when the attribute is specified and contains a numeric string which uniquely identifies the constituent constraint applicable to the constituent for which the attribute is specified. This facilitates the processing of documents. A list of these identifiers is given in Tables 5 and 6.

NOTES

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

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

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

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

6.6.5 Alternative representation

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

The specification and use of this attribute is optional. The string of characters specified shall belong to the character repertoires indicated in the document profile attribute "alternative representation character sets" (see 6.7.4.3). If the latter attribute is not explicitly specified in the document profile, then the default defined in CCITT Recs. T.410-Series | ISO 8613 is assumed. The control functions space (SP), carriage return (CR) and line feed (LF) may also be used within the character string but no other control function is allowed; hence graphic character sets cannot be changed within the alternative representation.

6.6.6 Automatic numbering and referencing mechanisms

6.6.6.1 Introduction

This Recommendation provides general mechanisms to support the automatic numbering of various types of constituents in a document and for referencing those numbers from other constituents in the document. For example, the numbering of segments (such as chapters, sections and annexes), tables, figures, footnotes, lists of items and pages is supported.

Also character strings may be specified for various constituents and these strings may be referenced from other parts of a document in order to support a general referencing mechanism within a document.

To achieve these features, this Recommendation provides the bindings listed in 6.6.6.2. Subclauses 6.6.6.3 to 6.6.6.11 describe how these bindings are used in the automatic numbering and referencing schemes. These descriptions are not intended to restrict the use of the bindings provided by this Recommendation or the mechanisms used to achieve these numbering and referencing schemes.

TABLE 5/T.506

List of number string identifiers for logical constituent constraints

Logical constituent constraints	Numeric string identifier
DocumentLogicalRoot	0
Passage	1
NumberedSegment	2
Number	3
Title	4
Caption	5
Paragraph	6
Phrase	7
Footnote	8
FootnoteNumber	9
FootnoteReference	10
FootnoteBody	11
FootnoteText	12
Figure	13
BodyText	14
Reference	15
ReferencedContent	16
BodyRaster	17
BodyGeometric	18
CommonContent	19
CommonText	20
CommonRaster	21
CommonGeometric	22
Description	23
Artwork	24
NumberedList	25
UnNumberedList	26
DefinitionList	27
ListItem	28
ListTerm	29
Table	30
Row	31
TableComponent	32
RowComponent	33
Form	34
EntryElement	35
EntryGroup	36
CommonReference	37
CommonNumber	38
CurrentInstance	39
PageNumber	40
EntryText	41
EntryRaster	42
EntryGeometric	43
TableNumber	44

TABLE 6/T.506

List of number string identifiers for layout constituent constraints

Layout constituent constraints	Numeric string identifier
DocumentLayoutRoot	0
PageSet	1
Page	2
RectoPage	3
VersoPage	4
CompositeHeader	5
FixedCompositeBody	6
VariableCompositeBody	7
ColumnFixed	8
ColumnVariable	9
SnakingColumns	10
SynchronizedColumns	11
BasicFloat	12
CompositeFloat	13
BasicColumn	14
FootnoteArea	15
ArrangedContentFixed	16
ArrangedContentVariable	17
SourcedContentFixed	18
SourcedContentVariable	19
CompositeFixtureVariable	20
CompositeFixtureFixed	21
BasicFixture	22
CompositeColumnFixed	23
CompositeColumnVariable	24
CompositeCommon	25
CompositeArtwork	26
BasicHeader	27
BasicBody	28
GenericBlock	29
SpecificBlock	30
FormArea	31
CompositeFooter	32
BasicFooter	33
TableHeader	34
EntryGroupArea	35
TableArea	36
TableLabel	37
CompositeTableLabel	38
LabelComponent	39
RowArea	40
Cell	41
SubRowGroup	42
SubRow	43
TableLabelContent	44
FormEntryArea	45

6.6.6.2 Bindings

The binding listed below may be specified, unless otherwise stated, on any composite logical constituent constraint, on basic logical constituent constraints of the types BodyText, BodyRaster and BodyGeometric and on layout constituent constraints of the types DocumentLayoutRoot, PageSet, Page, RectoPage and VersoPage.

Groups of bindings have names whose general form is ’<name>-<n>’. The character string <name> serves to identify a particular group of bindings and <n> is a string of characters that serves to identify a particular binding. The field <n> is a sequence of characters taken from the set of characters ’0..9’; this sequence may be of any length, but shall consist of a string representing an integer with no leading zeroes.

Binding values may consist of integers or character strings. In the case of integers, any value may be specified. In the case of character strings, the string may consist of any of the 94 characters of the IRV of ISO 646 revised 1991, ISO-IR6, plus the character space. Any other character repertoire may be used provided it is designated and invoked by the appropriate designation and invocation sequences, and indicated in the document profile as a non-basic value. No other control functions may be used.

6.6.6.2.1 Binding ’prefixes-<n>’

This binding specifies a character string that is typically used to specify a prefix string in a character string represented by another binding. Examples are prefixes to segment, footnote and page numbers.

6.6.6.2.2 Binding ’suffixes-<n>’

This binding specifies a character string that is typically used to specify a suffix string in a character string represented by another binding. Examples are suffixes to segment, footnote and page numbers.

6.6.6.2.3 Binding ’numberstring-<n>’

This binding specifies a character string that typically consists of one or more numerals and separators that constitutes, for example, a segment, figure, table, list item or footnote number in a document. An example is the string “3.4.3.6” which might identify a sub-section in a document.

6.6.6.2.4 Binding ’numbers-<n>’

This binding specifies an integer that is associated with a particular constituent. This integer is typically used to generate a numeral or a sequence of numerals, represented by the binding 'numberstring-<n>' that identifies, for example, a particular segment, footnote, table, figure, list item or page within a document.

6.6.6.2.5 Binding ’separator-<n>’

This binding specifies a character string that is typically used to represent the separators between numerals in a string represented by the binding 'numberstring-<n>'. An example is the string “3.4.3.6” where the character ”.” forms the separator.

6.6.6.2.6 Binding ’string-<n>’

This binding specifies a character string that is associated with a constituent and is used to support a general referencing mechanism with a document. Typically this binding is used to support the referencing of character content specified in one part of the document from another part of the document. For example, this binding might be used to carry the title of a chapter or figure that is referenced in some other part of the document.

6.6.6.2.7 Binding ’PGnum’

This binding specifies an integer that typically represents a page number. This binding may only be specified on layout constituent constraints of the types DocumentLayoutRoot, PageSet, RectoPage, VersoPage and Page.

6.6.6.2.8 Binding ’fnotenumber’

This binding specifies an integer that is specifically provided to represent the numbers that identify footnotes. This binding may only be specified for the logical constituent constraints DocumentLogicalRoot, Passage and Footnote, and is provided for compatibility with FOD026 | PM-26.

6.6.6.2.9 Binding ’fnotestring’

This binding specifies a character string which is the equivalent of the number represented by the binding ’fnotenumber’. This binding may only be specified for the logical constituent constraint Footnote, and is provided for compatibility with FOD 26 | PM-26.

6.6.6.3 Numbering of segments

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

The format of this character string is as follows:

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

This format is defined by a string expression which is specified by the macro SEGMENTNUMBER (see 7.3.1). The description below indicates how this string is typically generated.

The fields <pre-str> and <suf-str> are optional prefix and suffix character strings respectively which may be of any length. These may be predefined in the expression or derived from bindings of the type ’prefix-<n>’ and ’suffix-<n>’ respectively that are defined on constituents at higher levels in the document structure.

The field <num-str> is the segment identifier which has the following general form:

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

where [..]... indicates optional repetition. That is, a segment identifier consists of a single numeral or a sequence of two or more numerals, each of which is separated by a character string called a ’separator’. An example is the string“7.3.3.1”.

Segment identifiers are represented by the binding 'numberstring-<n>'. This binding may be explicitly specified or generated automatically by a string expression defined by the macro USENUMBERSTRINGS (see 7.3.1) which has the general form:

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

The field <numberstring-x> is a reference to the segment identifier (i.e. another instance of a binding of the type 'numberstring-<n>') that is specified for the immediately superior NumberedSegment. This allows hierarchically structured numbering schemes to be specified. If this NumberedSegment does not exist, then this field is empty and, in this case, non-hierarchical.

The field <separator-y> is character string derived from a binding of the type 'separator-<n>' specified at some higher level in the document structure. This field may be empty.

The field <number-z> is the number represented in the form of a character string that is applicable to the NumberedSegment whose identifier is being constructed. This number may be represented in the form of an Arabic numeral string, upper or lower case Roman numeral string or by upper or lower case alphabetic characters.

The integer value corresponding to the field <number-z> may be generated by one of two methods. The value may be generated by an ORDINAL function within the expression defined by the macro USENUMBERSTRINGS. Alternatively, it may be derived by a binding of the type 'number-<n>' which is specified for the NumberedSegment whose identifier is being constructed. The binding 'number-<n>' is initialized at some suitable point in the document and is then automatically incremented on each successive NumberedSegment. This is achieved using an expression defined by the macro USENUMBERS (see 7.3.1).

The constructed binding 'numberstring-<n>' generated by the macro USENUMBERSTRINGS may then be referred to by a content generator specified by the constituent Number to generate the identifier of the NumberedSegment as indicated earlier in this subclause.

This binding is also available for constructing the segment identifiers used at lower levels of NumberedSegment. This means that a hierarchical numbering scheme may be specified for the NumberedSegments at different levels of the document structure.

The level number shall be indicated using the smallest possible number of characters, that is, there shall be no leading zeroes.

A document may contain any number of different independent numbering schemes. This is achieved by setting the value of bindings of the type ’number-<n>’, ’numberstring-<n>, ’prefix-<n>’ and ’suffix-<n>’; and the expressions indicated above at appropriate points in the document structure.

The above mechanism may be used for different purposes; subsequent subclauses illustrate how this mechanism is typically used for the numbering of figures, tables, lists of items and footnotes.

6.6.6.4 Numbering of figures

The mechanism used for numbering figures is the same as that used for numbering segments (see 6.6.6.3). That is, the number of a figure is generated by a content generator specified in the constituent Number which is immediately subordinate to the given Figure. This content generator contains an expression whose value is defined by the macro SEGMENTNUMBER. The number associated with each figure may be represented by a binding of the type ’number-<n>’ and a binding ’numberstring-<n>’ is used to specify the character string that represents the figure number. The generation of these bindings is as described in 6.6.6.3.

The figures in a document may be consecutively numbered throughout the document irrespective of the part of the document in which the figures are contained. Alternatively, the numbers may be linked to the part of the document to which they relate; for example, the figures in chapter 3 of a document can be specified as 3.1, 3.2, 3.3 and so on.

6.6.6.5 Numbering of lists

A list of items that are individually numbered is represented by the logical constituent NumberedList. Each item is represented by a constituent constraint of the type ListItem and each number belonging to each item is represented by a preceding constituent constraint of the type Number.

Number contains a content generator which, when evaluated, generates the number belonging to the subsequent item. The format of this content generator is the same as that used for numbering segments (see 6.6.6.3) and is defined by the macro SEGMENTNUMBER.

As described in 6.6.6.3, the binding ’numberstring-<n>’ is used to represent the number string belonging to each item. In this case, this binding is generated on the constituent constraint Number which is referred to by the content generator contained in the same constituent constraint. (Note that in the case of segment numbers, the binding ’numberstring-<n>’ is generated on the superior object.)

6.6.6.6 Numbering of tables

The constituent constraint TableNumber is typically used to represent a character string that constitutes the number that relates to a Table. This string is laid out in a TableHeader frame by means of a SourcedContentFixed frame. The latter frame specifies the attribute “logical source” which indicates the instance of the constituent constraint CommonContent that contains the subordinate TableNumber which specifies the required table number.

The character string represented by TableNumber is generated by a content generator which defines a string expression specified by the macro TABLENUMBER (see 7.3.1). The general format of this character string is as follows:

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

The fields <pre-str> and <suf-str> are optional prefix and suffix character strings which are pre-defined in the expression or are derived from bindings of the types 'prefix-<n>' and 'suffix-<n>' respectively that are specified for constituents at higher levels in the document structure.

The field <num-str> is a character string that represents the identifier of the table being laid out. It is obtained from the binding 'numberstring-<n>' which is specified for the logical object of the type Table which is being laid out. That is, the field <num-str> is derived from the current instance of the constituent constraint of the type Table.

The general format of the field <num-str> and the mechanisms used to specify and generate this field are described in 6.6.6.3.

Tables, like figures, may be independently numbered throughout a document, or their numbers may be linked to the segments in which they are contained. An example of a table number is the string “1.1.5” where “1.1” is the number of the segment to which the table belongs, “.” is a separator and “5” is the number associated with the particular table.

6.6.6.7 Footnote numbering

The constituent constraints FootnoteReference and FootnoteNumber contain content generators which, when evaluated during the layout process, produce character strings that serve to identify the Footnote to which the constituent constraints FootnoteReference and FootnoteNumber are subordinate.

The format of this character string is as follows:

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

This string is defined by a string expression specified by the macro FNNUMBER (see 7.3.1).

The numbering mechanism is the same as that used for numbering segments (see 6.6.6.3). Thus, the fields <pre-str> and <suf-str> are optional prefix and suffix character strings respectively which may be of any length. These are derived from bindings of the type 'prefix-<n>' and 'suffix-<n>' respectively that are defined on constituents at a higher level in the document structure.

The field <num-str> may be derived by one of three methods. It may be represented by bindings of the type 'numberstring-<n>' which are derived using an expression specified by the macro USENUMBERSTRINGS as described in 6.6.6.3. Alternatively, it may be derived from a binding of the type 'fnotestring' which is specified on the constituent Footnote to which the constituent constraints FootnoteReference and FootnoteNumber are subordinate. This field is automatically generated using expressions defined by the macros INCFNOTENUMBER and FNOTENUMBERSTRING. The field <num-str> may also be explicitly specified; this case is defined by the macro FNOTESTRINGLITERAL.

The above mechanisms allow footnotes to be numbered consecutively throughout a document, or any number of independent footnote numbering schemes may be used. For example, the footnotes applicable to segments, figures and tables may all be independently numbered.

6.6.6.8 Page numbering

The constituent constraint PageNumber is specifically provided to represent common content that contains a page number and that is to be placed on each successive page of a document. A mechanism is provided which allows the page number to be automatically incremented on each successive page of a document.

The format of the content generator specified by the constituent PageNumber is as follows:

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

This format is defined by a string expression specified by the macro PGNUMBER (see 7.3.1).

The fields <pre-str> and <suf-str> are optional prefix and suffix character strings respectively which may be of any length. These may be explicitly specified in the expression, or they may be derived from bindings of the type 'prefix-<n>' and 'suffix-<n>' respectively that are defined on constituents at a higher level in the document structure.

The field <num-str> is the page number. This consists of a single number derived from the binding 'number-<n>' or 'PGnum' which is specified for the current instance of the frame or page in which the page number is to be laid out. A page number may be represented in the form of Arabic numeric strings, an upper or lower case Roman numeric string or an equivalent upper or lower case alphabetic string.

The binding 'number-<n>' is initialized at the document layout root, page set level or a particular page class (using the macro INITIALISEBINDINGS defined in 7.3.1). The binding 'PGnum' is initialized at the document layout root or page set level (using the macro INITIALISEPGNUM defined in 7.3.1). This binding is automatically incremented on each successive page using an expression specified by the macro USEPGNUMBERS (see 7.3.1). By placing initialization on the layout root, rather than on the pageset classes, the pagenumbering may be defined to be continued from one pageset to the next.

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

Similar numbering is applicable for page sets.

6.6.6.9 Referenced content

ReferencedContent is a constituent constraint that is provided to support a general referencing mechanism within the content of the body part of a document. This constituent constraint, therefore, is used to represent character content that contains a reference to content specified elsewhere in a document. Examples are references to strings that represent the numbers of segments, figures, tables, footnotes and pages. Particular strings specified by 'string-<n>' may be referenced.

This constituent constraint contains a content generator which, when evaluated, produces a character string of the following form:

<pre-str> a sequence of <ref-str><suf-str>

This content generator is defined by a string expression which is specified by the macro REF (see 7.3.1).

The fields <pre-str> and <suf-str> are optional prefix and suffix character strings which may be explicitly defined in the expression or may be derived from bindings of the type 'prefix-<n>' and 'suffix-<n>' respectively that are defined on constituents at a higher level in the document structure.

The field <ref-str> is a character string that is obtained from content specified on a particular constituent in the document by reference to one of the following bindings: 'numberstring-<n>', 'number-<n>', 'string-<n>', 'fnotestring' or 'PGnum'.

The following referencing mechanisms are permitted in this case:

– a particular logical object is referenced for the required binding;

– a particular logical object is referenced and a search for the required binding is made on the logical objects superior to the referenced object.

The referencing mechanism shall take into account the constituent constraints for which the particular bindings are permitted, as defined in 6.6.6.2.

6.6.6.10 Common references

CommonReference is a constituent constraint within the common part of the logical structure of a document that represents common character content that may be reproduced in more than one place in a document. This constituent constraint is specifically provided to represent content which contains a reference to content specified elsewhere in a document. This content is specified by bindings. Examples are references to character strings that represent the numbers of segments, figures, tables and footnotes in a document.

This constituent constraint contains a content generator, the general format of which is the same as the content generator specified for the constituent constraint ReferencedContent (see 6.6.7.9) and content may be obtained from the following bindings: 'numberstring-<n>', 'numbers-<n>', 'string-<n>', 'fnotestring' or 'PGnum'.

This content generator is defined by an expression which is specified by the macro COMMONREF.

The following referencing mechanisms are permitted in this case:

– a current page or frame, or a logical object which is laid out currently is referenced for the required binding;

– a current page or frame, or a logical object which is laid out currently is referenced and a search for the required binding is made on the constituents superior to a referenced constituent;

– a page or frame into which a particular logical object is laid out is referenced for the required binding;

– a page or frame into which a particular logical object is laid out is referenced and a search for the required binding is made on the layout objects superior to the referenced page or frame.

The referencing mechanism shall take into account the constituents for which the particular bindings are permitted, as defined in 6.6.6.2.

6.6.6.11 Current instance references

The constituent constraint CurrentInstance, like CommonReference, represents common character content that may be reproduced in more than one place in a document when it is laid out.

This constituent constraint is specifically provided to represent the current instance of a character string that is to be laid out in multiple places in a document. A typical example is the reproduction of a title of chapter on each page of a document in which that chapter is reproduced.

The constituent constraint CurrentInstance contains a content generator, the general format of which is the same as the content generator specified for the constituent constraint ReferencedContent. This content generator is restricted to referencing content obtained from bindings of the type ’string-<n>’.

The referencing mechanism allows bindings of the type ’string-<n>’ to be referenced for the current instance of a logical constituent of a specified class or the current instance of a particular frame or page.

The format of this content generator is defined by the macro CURRENTINSTANCE (see 7.3.1).

6.6.6.12 Common number references

The constituent constraint CommonNumber represents common character content that may be reproduced in more than one place in a document when it is laid out.

This constituent constraint is specifically provided to represent identifiers of other parts of a document, that is, character strings that represent the numbers of segments, figures and tables within a document.

This constituent constraint contains a content generator, the general format of which is the same as the content generator specified for the constituent constraint ReferencedContent (see 6.6.6.9). This content generator may reference content specified by the bindings ’numberstring-<n>’ and ’numbers-<n>’.

The format of this content generator is defined by the macro COMMONNUMBER (see 7.3.1).

6.6.7 User readable comments

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

The information consists of a string of characters which shall belong to one of the graphic character sets indicated in the document profile attribute “comments character sets” (see 6.7.4.2). If the latter attribute is not explicitly specified, then the default defined in CCITT Recs. T.410-Series | ISO 8613 is assumed. The control functions space (SP), carriage return (CR) and line feed (LF) and code extension control functions may also be used within the character string, but no other control functions are allowed.

6.6.8 User visible name

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

The information consists of a string of characters which shall belong to one of the graphic character sets indicated in the document profile attribute “comments character sets” (see 6.7.4.2). If the latter attribute is not explicitly specified, then the default defined in CCITT Recs. T.410-Series | ISO 8613 is assumed. The control functions space (SP), carriage return (CR), line feed (LF) and code extension control functions may also be used within the character string, but no other control functions are allowed.

6.6.9 Unit scaling

The document profile attribute “unit scaling” may be specified to indicate a scaling factor that shall be applied to all attributes and control functions values that specify absolute or relative positions and dimensions. This attribute specifies two integers, *m* and *n*, which indicate that these values are to be interpreted as being equal to *m/n* MU.

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 shall be specified in every document.

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

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

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

6.7.1 Document constituent information

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

6.7.1.1 Presence of document constituents

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

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

6.7.1.2 Resource document information

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

6.7.1.3 External document information

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

6.7.2 Document identification information

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

6.7.2.1 Document application profile information

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

6.7.2.2 Document architecture class information

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

6.7.2.3 Content architecture classes information

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

6.7.2.4 Interchange format class information

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

6.7.2.5 ODA version information

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

6.7.2.6 Document reference

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

6.7.3 Document default information

This information specifies various default values for attributes used in the document. The default values that are allowed are specified in the various subclauses of clause 6. The specification of this information is only required when it is required to specify a default value which is other than the standard default value specified in CCITT Recs. T.410-Series | ISO 8613.

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

– document architecture attributes;

– character content attributes;

– raster graphics attributes;

– geometric graphics attributes.

6.7.4 Non-basic characteristics information

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

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

– profile character sets;

– comments character sets;

– alternative representation character sets;

– page dimensions;

– medium types;

– character presentation features;

– raster graphics presentation features;

– raster graphics coding attributes.

NOTE – In addition to the above, layout paths and borders may be specified for upwards compatibility with FOD026 | PM-26.

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

6.7.4.1 Profile character sets

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

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

– The code extension announcer shall be 04/03 when specified. This code extension announcer specifies the use of G0 and G1 sets in an 8-bit environment and also the invocation of G0 and G1 sets into GL and GR respectively. Thus, in each attribute to which this attribute applies, invocation shift functions are not necessary because G0 and G1 sets are implicitly invoked by this code extension announcer.

– *G0 set* – Only ISO-IR6 (the IRV of ISO 646 revised 1991), ISO-IR2 (the primary set of ISO 6937-2), or any other version of ISO IEC 646 may be designated for this set; these graphic character sets are implicitly invoked in GL.

- *G1 set* – No restrictions are placed on the graphic character sets that may be designated for this set; these graphic character sets are implicitly invoked in GR.
- The empty set shall be designated into G1 and invoked into GR if no other specific character set is invoked in GR.

If the attribute “profile character sets” is not specified, then the default defined in CCITT Recs. T.410-Series | ISO 8613 is assumed.

6.7.4.2 Comments character sets

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

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

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

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

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

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

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

If the attribute “comments character sets” is not specified, the default defined in CCITT Recs. T.410-Series | ISO 8613 is assumed.

6.7.4.3 Alternative representation character sets

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

The restriction on profile character sets described in 6.7.4.1 is also applied. If this attribute is not explicitly specified in the document profile, the default defined in CCITT Recs. T.410-Series | ISO 8613 is assumed.

6.7.5 Fonts list

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

6.7.6 Document management attributes

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

- document description (see 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 Rec. T.414 | ISO 8613-4.

The string of characters used in the document management attributes shall belong to the character sets indicated in the document profile attribute “profile character sets” (see 6.7.4.1). If the latter attribute is not explicitly specified in the document profile, then the default character set is the minimum subrepertoire of ISO 6937-2.

The control functions space (SP), carriage return (CR) and line feed (LF) may also be used within the character strings, but no other control functions are allowed. Therefore, the graphic character set cannot be changed in the document management attributes.

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

7 Specification of constituent constraints

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

7.1 Introduction

The structure diagrams illustrating the relationships between the constituents in the logical structures are shown in 7.1.1. The macros indicated on these diagrams are defined in 7.3.1. These macros define the permissible values for the attribute “generator for subordinates” that are applicable to the constituents, and define the allowed structures that are supported by this profile.

The structure diagrams illustrating the layout structures are shown in 7.1.2. The macros indicated on these diagrams are defined in 7.4.1.

7.1.1 Diagrams of relationships of logical constituents

See Figures 21 to 33.

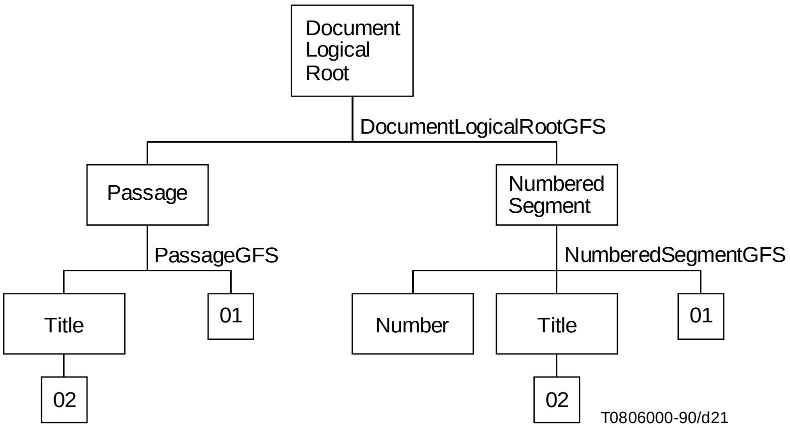


FIGURE 21/T.506

DocumentLogicalRoot, 1st level

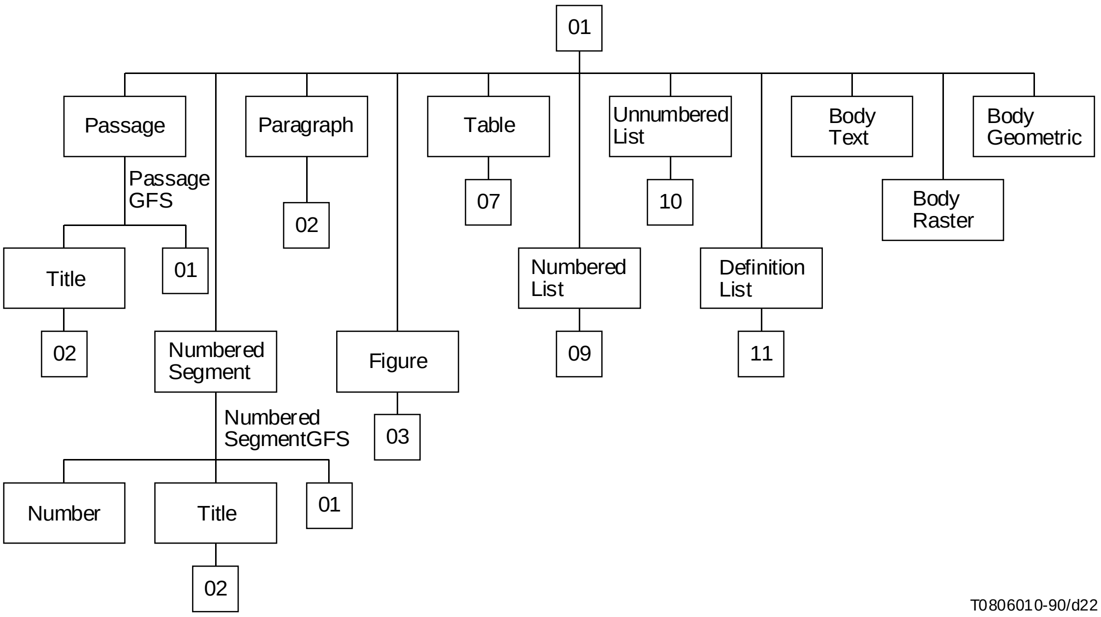


FIGURE 22/T.506

DocumentLogicalRoot, 2nd level

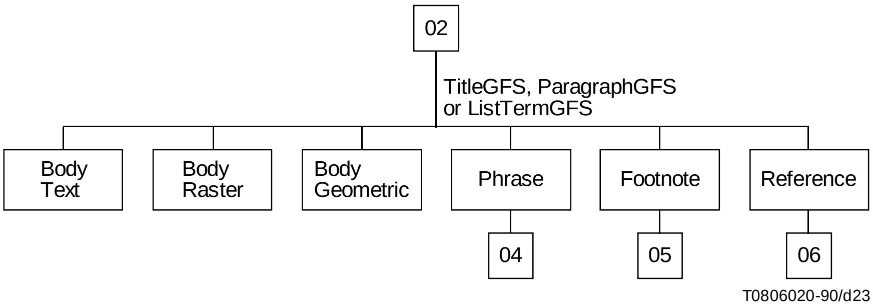


FIGURE 23/T.506

Title, Paragraph and ListTerm

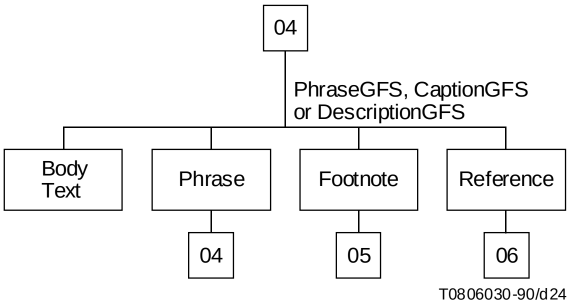


FIGURE 24/T.506

Phrase, Caption and Description

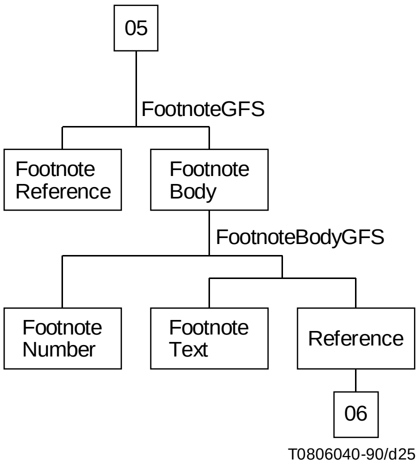


FIGURE 25/T.506

Footnote

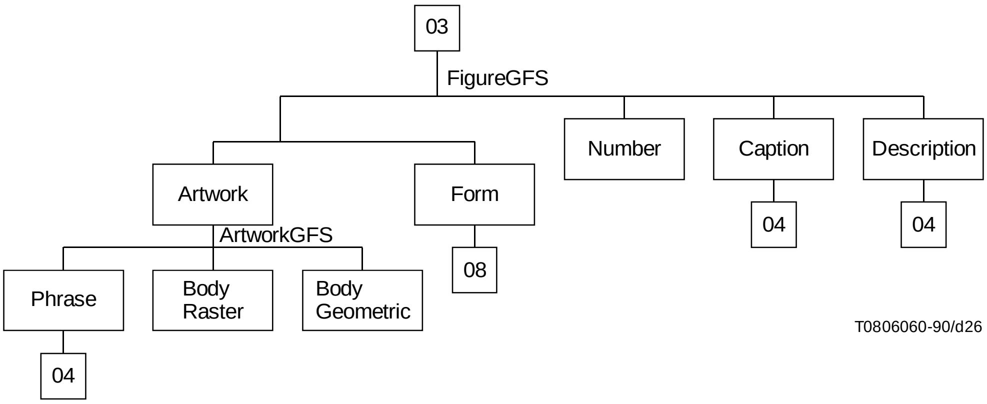


FIGURE 26/T.506

Figure

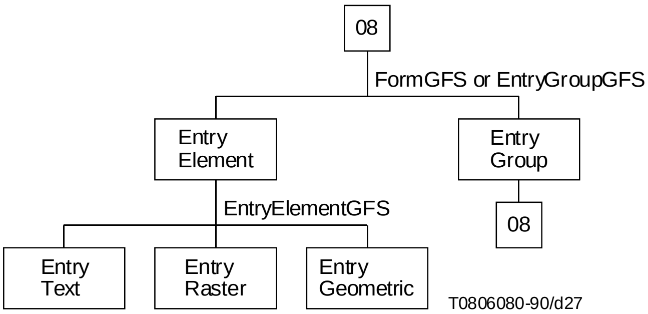


FIGURE 27/T.506

Form

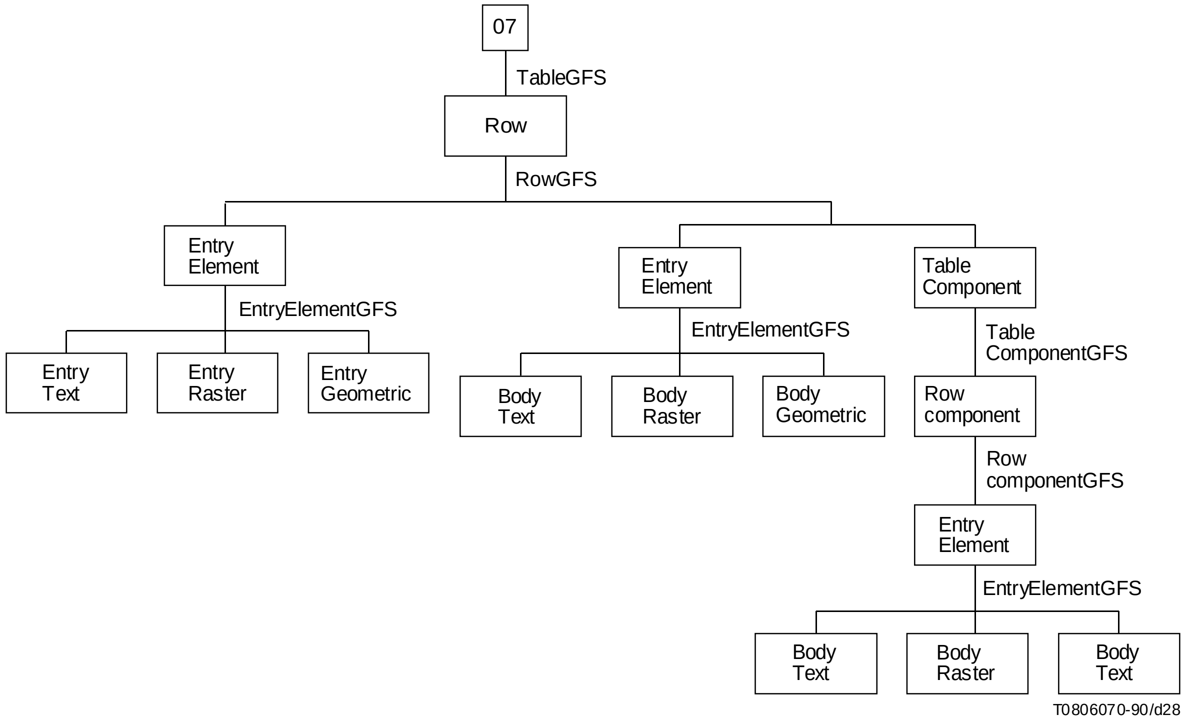


FIGURE 28/T.506

Table

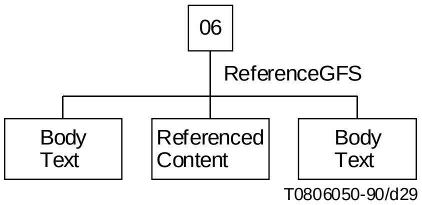


FIGURE 29/T.506

Reference

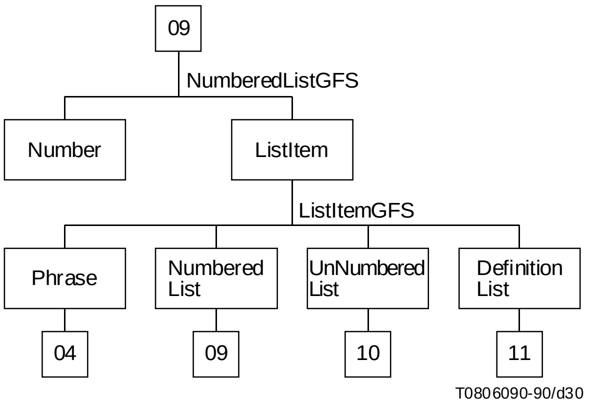


FIGURE 30/T.506

NumberedList

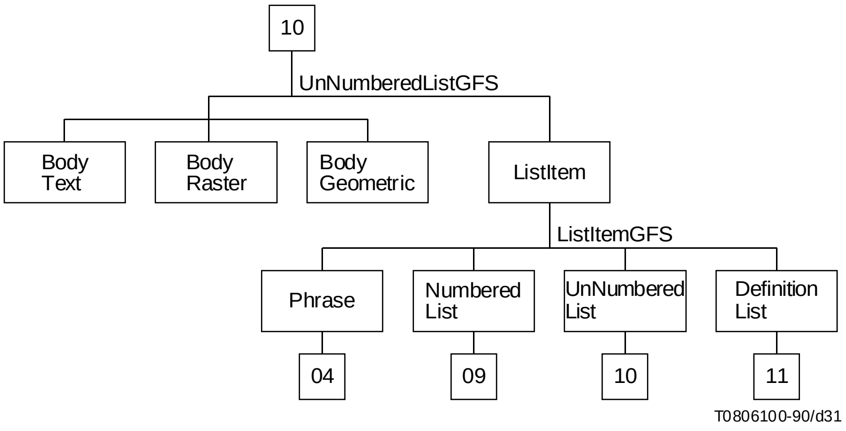


FIGURE 31/T.506

UnNumberedList

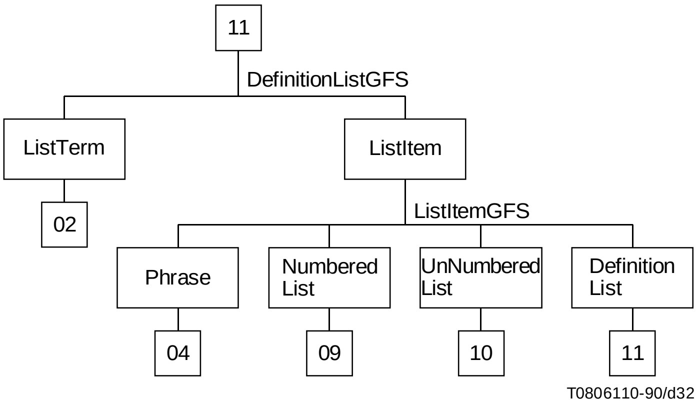


FIGURE 32/T.506

DefinitionList

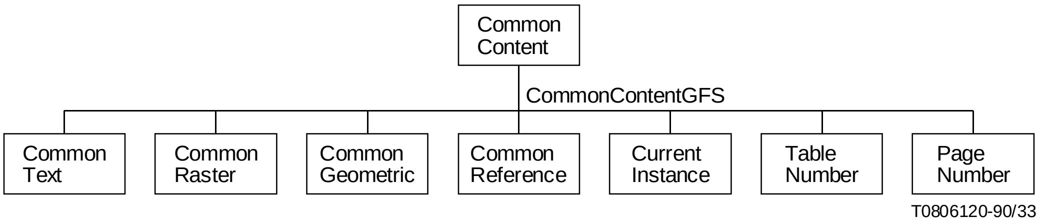


FIGURE 33/T.506

CommonContent

7.1.2 Diagrams of relationships of layout constituents

See Figures 34 to 45.

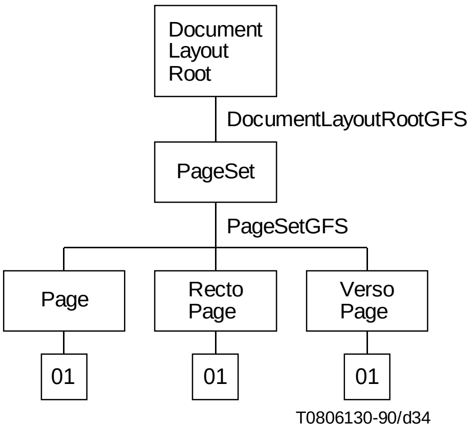


FIGURE 34/T.506

DocumentLayoutRoot

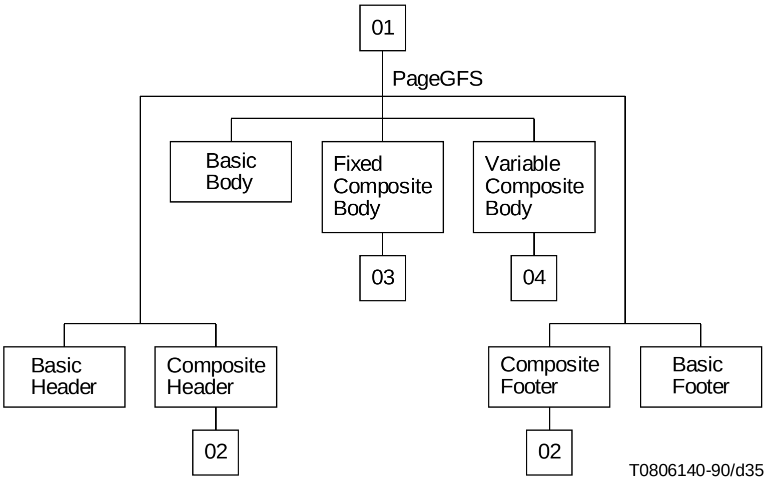


FIGURE 35/T.506

Page, RectoPage and Versopage

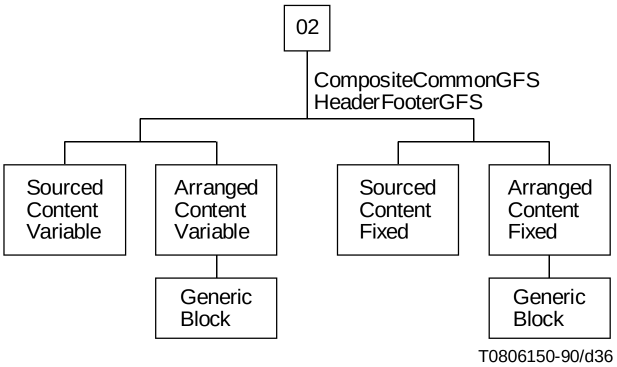


FIGURE 36/T.506

CompositeHeader, CompositeFooter and CompositeCommon

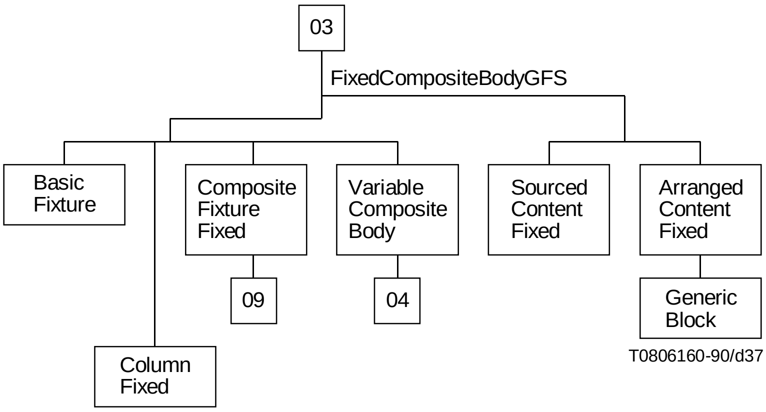


FIGURE 37/T.506

FixedCompositeBody

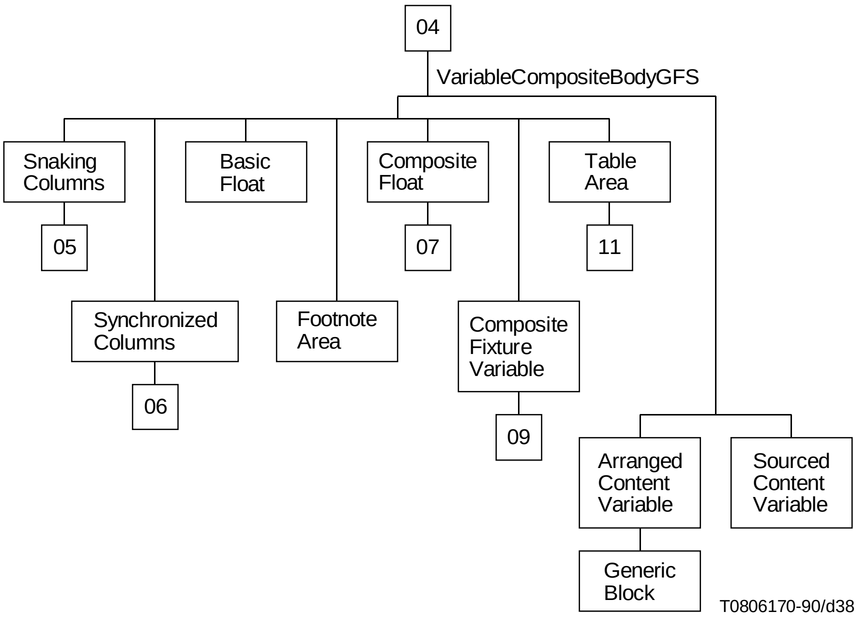


FIGURE 38/T.506

VariableCompositeBody

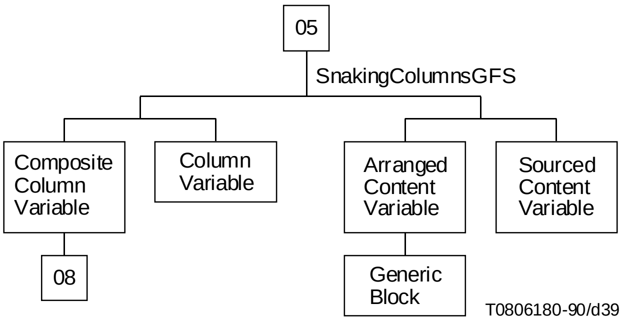


FIGURE 39/T.506

SnakingColumns

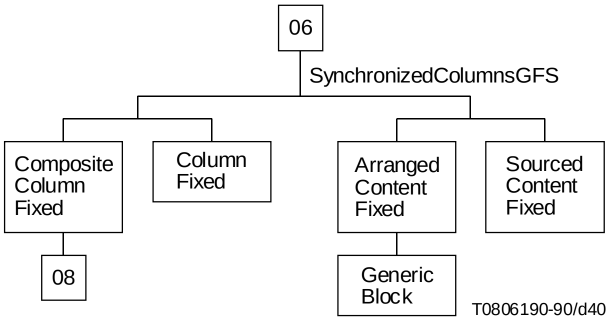


FIGURE 40/T.506

SynchronizedColumns

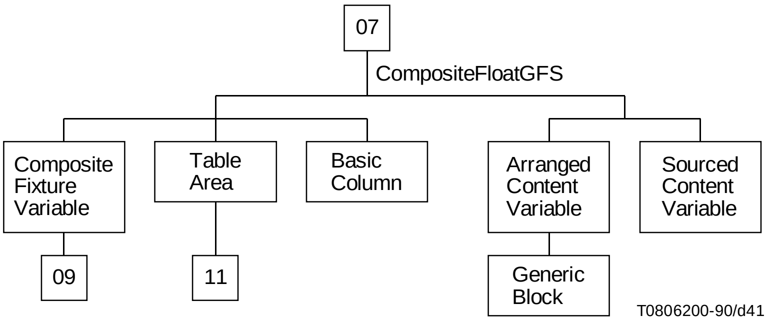


FIGURE 41/T.506

CompositeFloat

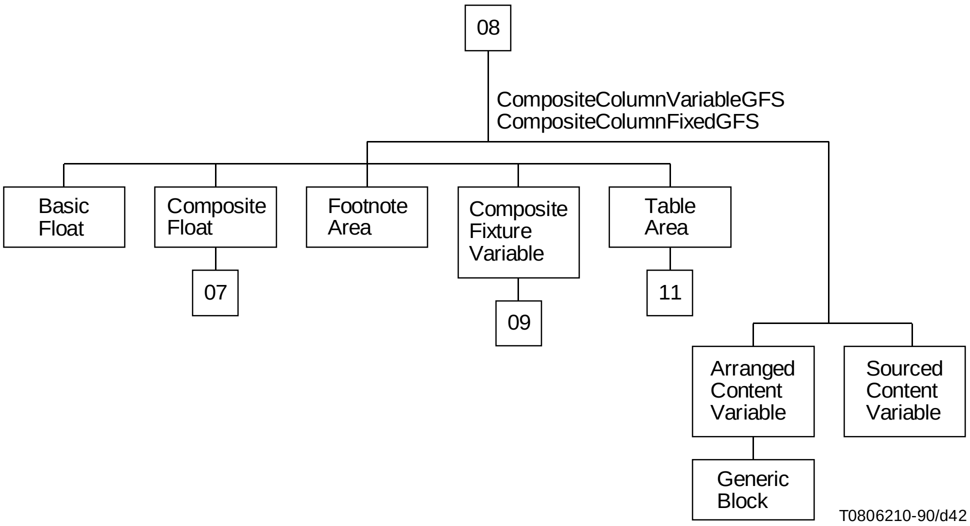


FIGURE 42/T.506

CompositeColumnVariable and CompositeColumnFixed

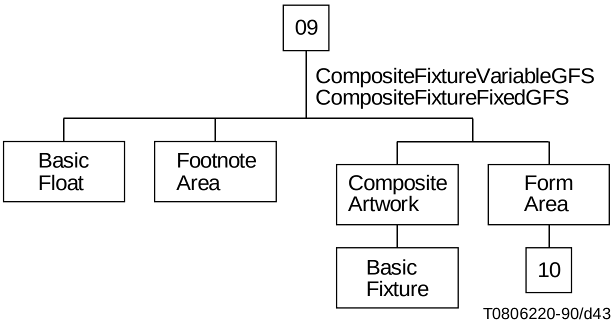


FIGURE 43/T.506

CompositeFixtureVariable and CompositeFixtureFixed

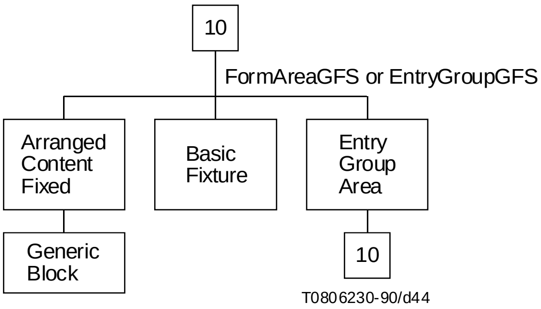


FIGURE 44/T.506

FormArea and EntryGroup

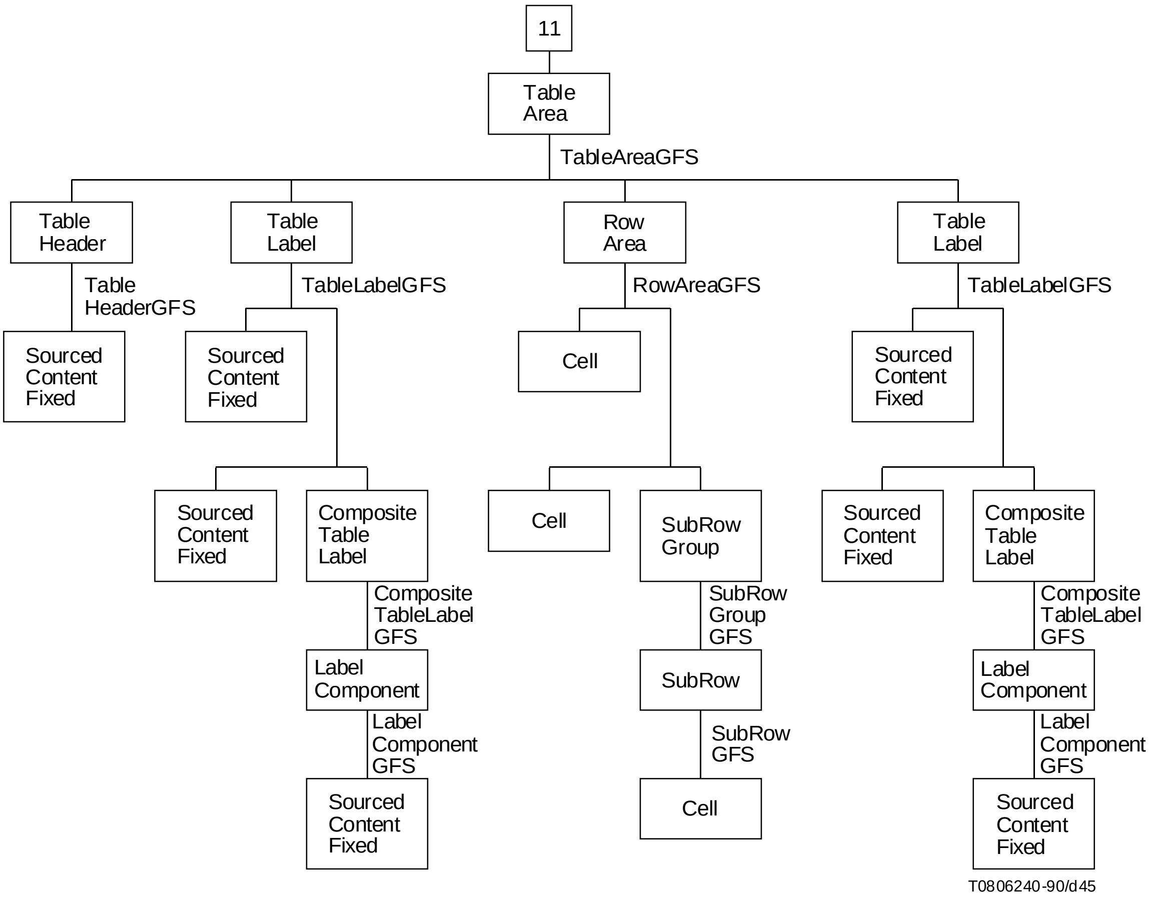


FIGURE 45/T.506

TableArea

7.1.3 Notation

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

a) [Clarification]

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

b) [Clarification]

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

c) [Extension]

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

1) Table 7 defines the symbols that are introduced to denote shift functions.

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

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

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

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

<escape-sequence>...

d) [Clarification]

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

e) [Extension]

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

f) [Extension]

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

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

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

This expression is equivalent to the following expressions:

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

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

. . . .

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

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

TABLE 7/T.506

Symbols to denote shift functions

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

7.2 Document profile constituent constraints

7.2.1 Macro definitions

```
DEFINE(FC, "ASN.1{2 8 2 6 0}" -- formatted character content --)
DEFINE(PC, "ASN.1{2 8 2 6 1}" -- processable character content --)
DEFINE(FPC, "ASN.1{2 8 2 6 2}" -- formatted processable character content --)
DEFINE(FPR, "ASN.1{2 8 2 7 2}" -- formatted processable raster graphics content --)
DEFINE(FPG, "ASN.1{2 8 2 8 0}" -- formatted processable geometric graphics content --)
```

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

```
DEFINE(DAC, "DocumentProfile (Document-architecture-class)")
DEFINE(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 ANSI A. Both Portrait and Landscape may be specified. Note that the above macro is defined for clarification of the specification and is not used in any other part of this DAP.* --

```
DEFINE(NonBasicPageDimensions, "
              REQ #horizontal-dimension
                     {REQ #fixed-dimension{<=39680}},
              REQ #vertical-dimension
                     {REQ #fixed-dimension {12401..56120}}
              |REQ #horizontal-dimension
                     {REQ #fixed-dimension{9241..39680}},
              REQ #vertical-dimension
                     {REQ #fixed-dimension {<=56120}}
                                    -- up to ISO A0 portrait --
              |REQ #horizontal-dimension
                     {REQ #fixed-dimension {<=56120}},
              REQ #vertical-dimension
                     {REQ #fixed-dimension {9241..39680}}
```

```
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {12401..56120}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {<=39680}}
                        -- up to ISO A0 landscape --
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {<=40800}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {12401..52800}}
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {9241..40800}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {<=52800}}
                        -- up to ANSI E portrait --
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {<=52800}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {9241..40800}}
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {12401..52800}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {<=40800}}
                        -- up to ANSI E landscape --
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {<=12141}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {12401..17196}}
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {9241..12141}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {<=17196}}
                        -- up to JIS B4(Japanese legal) portrait --
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {<=17196}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {9241..12141}}
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {12401..17196}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {<=12141}}
                        -- up to JIS B4(Japanese legal) landscape --
")

DEFINE(PermissiblePageDimensions, "
        REQ #horizontal-dimension
            {REQ #fixed-dimension {<=39680}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {<=56120}} -- up to ISO A0 portrait --
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {<=56120}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {<=39680}} -- up to ISO A0 landscape --
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {<=40800}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {<=52800}} -- up to ANSI E portrait --
        |REQ #horizontal-dimension
            {REQ #fixed-dimension {<=52800}},
        REQ #vertical-dimension
            {REQ #fixed-dimension {<=40800}} -- up to ANSI E landscape --
")

DEFINE(NominalPageSizes,  "
        REQ #horizontal-dimension {7015},           REQ #vertical-dimension {9920}
                                                    -- ISO A5 portrait --
        | REQ #horizontal-dimension {9920},         REQ #vertical-dimension {7015}
                                                    -- ISO A5 landscape --
            | REQ #horizontal-dimension {9920},     REQ #vertical-dimension {14030}
                                                    -- ISO A4 portrait --
            | REQ #horizontal-dimension {14030},    REQ #vertical-dimension {9920}
                                                    -- ISO A4 landscape --
            | REQ #horizontal-dimension {14030},    REQ #vertical-dimension {19840}
                                                    -- ISO A3 portrait --
            | REQ #horizontal-dimension {19840},    REQ #vertical-dimension {14030}
                                                    -- ISO A3 landscape --
            | REQ #horizontal-dimension {19840},    REQ #vertical-dimension {28060}
                                                    -- ISO A2 portrait --
            | REQ #horizontal-dimension {28060},    REQ #vertical-dimension {19840}
                                                    -- ISO A2 landscape --
            | REQ #horizontal-dimension {28060},    REQ #vertical-dimension {39680}
                                                    -- ISO A1 portrait --
            | REQ #horizontal-dimension {39680},    REQ #vertical-dimension {28060}
                                                    -- ISO A1 landscape --
            | REQ #horizontal-dimension {39680},    REQ #vertical-dimension {56120}
                                                    -- ISO A0 portrait --
            | REQ #horizontal-dimension {56120},    REQ #vertical-dimension {39680}
                                                    -- ISO A0 landscape --
            | REQ #horizontal-dimension {10200},    REQ #vertical-dimension {16800}
                                                    -- ANSI legal portrait --
            | REQ #horizontal-dimension {16800},    REQ #vertical-dimension {10200}
                                                    -- ANSI legal landscape --
            | REQ #horizontal-dimension {10200},    REQ #vertical-dimension {13200}
                                                    -- ANSI A portrait --
            | REQ #horizontal-dimension {13200},    REQ #vertical-dimension {10200}
                                                    -- ANSI A landscape --
            | REQ #horizontal-dimension {13200},    REQ #vertical-dimension {20400}
                                                    -- ANSI B portrait --
            | REQ #horizontal-dimension {20400},    REQ #vertical-dimension {13200}
                                                    -- ANSI B landscape --
            | REQ #horizontal-dimension {20400},    REQ #vertical-dimension {26400}
                                                    -- ANSI C portrait --
            | REQ #horizontal-dimension {26400},    REQ #vertical-dimension {20400}
                                                    -- ANSI C landscape --
            | REQ #horizontal-dimension {26400},    REQ #vertical-dimension {40800}
                                                    -- ANSI D portrait --
            | REQ #horizontal-dimension {40800},    REQ #vertical-dimension {26400}
                                                    -- ANSI D landscape --
            | REQ #horizontal-dimension {40800},    REQ #vertical-dimension {52800}
                                                    -- ANSI E portrait --
            | REQ #horizontal-dimension {52800},    REQ #vertical-dimension {40800}
                                                    -- ANSI E landscape --
            | REQ #horizontal-dimension {33600},    REQ #vertical-dimension {48000}
                                                    -- ANSI F portrait --
            | REQ #horizontal-dimension {48000},    REQ #vertical-dimension {33600}
                                                    -- ANSI F 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 --
")

DEFINE(GRAPHICRENDITIONS, "
            {'cancel'|'increased-intensity' |'decreased-intensity'
            |'italicised'| 'underlined' |'slowly-blinking'
            |'rapidly-blinking'|'negative-image'| 'crossed-out'
            |'primary-font'|'first-alternative-font'
            |'second-alternative-font' |'third-alternative-font'
            |'fourth-alternative-font' |'fifth-alternative-font'
```

```
              |’sixth-alternative-font’|’seventh-alternative-font’
              |’eighth-alternative-font’ |’ninth-alternative-font’
              |’doubly-underlined’ | ’normal-intensity’ |’not-italicised’
              |’not-underlined’|’steady’|’positive-image’
              |’not-crossed-out’ }... ")
```

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

```
DEFINE(CDEXTEN,    "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  --
")
```

-- Macro defining code extension announcers for DAP defaults --

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

-- Macros defining final character for designation --

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

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

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

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

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

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

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

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

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

-- Macros defining designation sequences --

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

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

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

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

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

-- Designate any character set to G3 --

```
DEFINE(DEG-EMPTY-G1,  "ESC 02/09 $FEMPTY")
```

-- Designate the empty set to G1 --

-- Macro defining permissible graphic character sets --

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

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

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

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

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

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

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

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

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

-- Macro defining comments character sets --

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

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

```
            { $DEG-CORE-G0 | $DEG-646-G0 }      -- designate G0 --

            {{$DEG-ANY-G1                       -- designate G1 --
            |$DEG-ANY-G2                        -- designate G2 --
            |$DEG-ANY-G3}...                    -- designate G3 --
            |$DEG-EMPTY-G1}
")
```

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

DEFINE(ALTCHAR, "$PROFCHAR")

7.2.2 Constituent constraints

7.2.2.1 DocumentProfile

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

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

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

            PERM      External-document-class           {ANY_VALUE},

            PERM      Resource-document                 {ANY_VALUE},

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

            -- document characteristics --
       REQ       Document-application-profile                  {-- See clause 8 for a definition of the permitted values for this
                                                                  attribute --},

PERM        Document-application-profile-defaults  {

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

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

            PERM  #transparency        {ANY_VALUE},

            PERM  #colour              {ANY_VALUE},

            PERM  #layout-path         {ANY_VALUE},

            PERM  #block-alignment     {ANY_VALUE},

            PERM  #border              {ANY_VALUE},

            PERM  #page-position       {ANY_VALUE},

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

            PERM     #character-content-defaults   {
                 PERM  #alignment                        {ANY_VALUE},
                 PERM  #character-fonts                  {ANY_VALUE},
                 PERM  #character-path                   {ANY_VALUE},
                 PERM  #character-spacing                {ANY_VALUE},
                 PERM  #character-orientation            {ANY_VALUE},
                 PERM  #code-extension-announcers        {$DAP-DEFAULT-CDEXTEN},
                 PERM  #first-line-offset                {ANY_VALUE},
                 PERM  #graphic-character-sets           {$DAP-DEFAULT-GRCHAR},
                 PERM  #graphic-character-subrepertoire  {ANY_VALUE},
                 PERM  #graphic-rendition                {$GRAPHICRENDITIONS},
                 PERM  #itemization                      {ANY_VALUE},
                 PERM  #kerning-offset                   {ANY_VALUE},
                 PERM  #line-layout-table                {ANY_VALUE},
                 PERM  #line-progression                 {ANY_VALUE},
                 PERM  #line-spacing                     {ANY_VALUE},
                 PERM  #pairwise-kerning                 {ANY_VALUE},
                 PERM  #indentation                      {ANY_VALUE},
                 PERM  #orphan-size                      {ANY_VALUE},
                 PERM  #proportional-line-spacing        {ANY_VALUE},
                 PERM  #widow-size                       {ANY_VALUE},
                 PERM  #formatting-indicator             {ANY_VALUE},
                 PERM  #initial-offset                   {ANY_VALUE}
       },

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

            PERM     #geometric-graphics-content-defaults {ANY_VALUE}
      },

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

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

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

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

            -- non basic document characteristics --
```

```
PERM      Profile-character-sets                               {$PROFCHAR},

PERM      Comments-character-sets                              {$COMCHAR},

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

PERM      Page-dimensions       {PMUL{$NonBasicPageDimensions}},

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

PERM  Layout-paths              {{'0-degrees'|'90-degrees'|'180-degrees'}...},
      -- These values need not be declared, they may be specified here for upwards compatibility from FOD26 --

PERM      Borders      {ANY_VALUE},
      -- Any values need not be declared, they may be specified here for upwards compatibility from FOD26 --

PERM      Coding-attributes   {

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

PERM      Presentation-features {
    PERM  #character-presentation-features {
        PERM    #character-orientation              {'90-degrees'},
          -- This value need not be declared, they may be specified here for upwards compatibility from FOD26 --

       PMUL {PERM #character-path                   {'90-degrees'|'180-degrees'|'270-degrees'}},
          -- These values need not be declared, they may be specified here for upwards compatibility from FOD26 --

       PMUL {PERM #character-spacing                {<100 | 100 | 160 | 200}},
          -- These values need not be declared, they may be specified here for upwards compatibility from FOD26 --

       PMUL {PERM #graphic-rendition                {'crossed-out'|'not-crossed-out'}},
          -- These values need not be declared, they may be specified here for upwards compatibility from FOD26 --

       PMUL {PERM #line-spacing             {ANY_VALUE} EXCEPT{200, 300, 400}},
          -- Values 100,150 and 600 need not to be declared, they may be specified here for upwards compatibility
             from FOD26 --

       PERM #line-progression                       {'90-degrees'},
          -- This value need not be declared, they may be specified here for upwards compatibility from FOD26 --

       PMUL {PERM #graphic-character-sets                      {$NON-BASIC-GRCHAR}},
       PMUL {PERM #graphic-character-subrepertoire                      {ANY_VALUE}}
},

    PERM #raster-graphics-presentation-features {
       PMUL {PERM #pel-path                 {'90-degrees'|'180-degrees'|'270-degrees'}},
       PERM #line-progression               {'90-degrees'},
       PMUL {PERM #pel-spacing              {ANY_VALUE}EXCEPT{16, 12, 8, 6, 5, 4, 3, 2, 1}},
          -- Any value of #pel spaces is permitted as basic --
          -- Basic values of #length are multiples of #pel spaces as listed --

       PMUL {PERM #spacing-ratio
            {REQ #line-spacing-value        {ANY_VALUE} EXCEPT {1},
             REQ #pel-spacing-value         {ANY_VALUE} EXCEPT {1}}}
}
},

          -- additional document characteristics --

    PERM  Unit-scaling           {REQ #unit-scaling-m {ANY_INTEGER},
                                  REQ #unit-scaling-n {ANY_INTEGER}},

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

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

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

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

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

        -- external references --
    PERM  References-to-other-documents    {ANY_VALUE},
    PERM  Superseded-documents             {ANY_VALUE},

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

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

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

7.3 Logical constituent constraints

7.3.1 Macro definitions

-- Defines any logical objects in a specific logical structure. --

```
DEFINE(LogicalObjects, "
 <logical-objects> ::=     OBJECT_ID_OF(DocumentLogicalRoot)
                          | OBJECT_ID_OF(Passage)
                          | OBJECT_ID_OF(NumberedSegment)
                          | OBJECT_ID_OF(Number)
                          | OBJECT_ID_OF(Title)
                          | OBJECT_ID_OF(Caption)
                          | OBJECT_ID_OF(Paragraph)
                          | OBJECT_ID_OF(Phrase)
                          | OBJECT_ID_OF(Footnote)
                          | OBJECT_ID_OF(FootnoteNumber)
```

```
                              | OBJECT_ID_OF(FootnoteReference)
                              | OBJECT_ID_OF(FootnoteBody)
                              | OBJECT_ID_OF(FootnoteText)
                              | OBJECT_ID_OF(Figure)
                              | OBJECT_ID_OF(BodyText)
                              | OBJECT_ID_OF(Reference)
                              | OBJECT_ID_OF(ReferencedContent)
                              | OBJECT_ID_OF(BodyRaster)
                              | OBJECT_ID_OF(BodyGeometric)
                              | OBJECT_ID_OF(Description)
                              | OBJECT_ID_OF(Artwork)
                              | OBJECT_ID_OF(NumberedList)
                              | OBJECT_ID_OF(UnNumberedList)
                              | OBJECT_ID_OF(DefinitionList)
                              | OBJECT_ID_OF(ListItem)
                              | OBJECT_ID_OF(ListTerm)
                              | OBJECT_ID_OF(Table)
                              | OBJECT_ID_OF(Row)
                              | OBJECT_ID_OF(TableComponent)
                              | OBJECT_ID_OF(RowComponent)
                              | OBJECT_ID_OF(Form)
                              | OBJECT_ID_OF(EntryElement)
                              | OBJECT_ID_OF(EntryGroup)
                              | OBJECT_ID_OF(EntryText)
                              | OBJECT_ID_OF(EntryRaster)
                              | OBJECT_ID_OF(EntryGeometric);
")
```

-- Defines any logical object classes other than classes referred by logical source. --

```
DEFINE(LogicalObjectClasses, "
 <logical-object-classes> ::=  OBJECT_CLASS_ID_OF(DocumentLogicalRoot)
                               | OBJECT_CLASS_ID_OF(Passage)
                               | OBJECT_CLASS_ID_OF(NumberedSegment)
                               | OBJECT_CLASS_ID_OF(Number)
                               | OBJECT_CLASS_ID_OF(Title)
                               | OBJECT_CLASS_ID_OF(Caption)
                               | OBJECT_CLASS_ID_OF(Paragraph)
                               | OBJECT_CLASS_ID_OF(Phrase)
                               | OBJECT_CLASS_ID_OF(Footnote)
                               | OBJECT_CLASS_ID_OF(FootnoteNumber)
                               | OBJECT_CLASS_ID_OF(FootnoteReference)
                               | OBJECT_CLASS_ID_OF(FootnoteBody)
                               | OBJECT_CLASS_ID_OF(FootnoteText)
                               | OBJECT_CLASS_ID_OF(Figure)
                               | OBJECT_CLASS_ID_OF(BodyText)
                               | OBJECT_CLASS_ID_OF(Reference)
                               | OBJECT_CLASS_ID_OF(ReferencedContent)
                               | OBJECT_CLASS_ID_OF(BodyRaster)
                               | OBJECT_CLASS_ID_OF(BodyGeometric)
                               | OBJECT_CLASS_ID_OF(Description)
                               | OBJECT_CLASS_ID_OF(Artwork)
                               | OBJECT_CLASS_ID_OF(NumberedList)
                               | OBJECT_CLASS_ID_OF(UnNumberedList)
                               | OBJECT_CLASS_ID_OF(DefinitionList)
                               | OBJECT_CLASS_ID_OF(ListItem)
                               | OBJECT_CLASS_ID_OF(ListTerm)
                               | OBJECT_CLASS_ID_OF(Table)
                               | OBJECT_CLASS_ID_OF(Row)
                               | OBJECT_CLASS_ID_OF(TableComponent)
                               | OBJECT_CLASS_ID_OF(RowComponent)
                               | OBJECT_CLASS_ID_OF(Form)
                               | OBJECT_CLASS_ID_OF(EntryElement)
                               | OBJECT_CLASS_ID_OF(EntryGroup)
                               | OBJECT_CLASS_ID_OF(EntryText)
                               | OBJECT_CLASS_ID_OF(EntryRaster)
                               | OBJECT_CLASS_ID_OF(EntryGeometric);
")
```

```
DEFINE(N, "<n>::={""0""|""1""|""2""|""3""|""4""|""5""|""6""|""7""|""8""|""9""}...;")
```

-- any string of characters from the set of characters: ’0’...’9’--

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

```
DEFINE(PREFIX,     "
 <prefix>  ::=    ""prefix-""<n>;
 $N
")
```

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

```
DEFINE(SUFFIX,     "
 <suffix>  ::=    ""suffix-""<n>;
 $N
")
```

-- Defines the separator binding. This binding is used to provide a separator character for a hierarchical form of a segment number, figure number, table number, footnote number or page number. The instances are differentiated by a suffix number. --

```
DEFINE(SEPARATOR,    "
 <separator>    ::=    ""separator-""<n>;
 $N
")
```

-- Defines the general number binding. This binding may be instanced for use as the numeric value such as in segment number, figure number, table number, list number, footnote number or page number bindings. The instances are differentiated by a suffix number. --

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

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

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

-- Defines the general string binding. The instances are differentiated by a suffix number. --

```
DEFINE(STRING, "
 <string>  ::=    ""string-""<n>;
 $N
")
```

-- Defines the names for footnote categories. The instances are differentiated by a suffix number. --

```
DEFINE(FOOTNOTECATEGORY, "
 <footnotecategory> ::=    ""Footnote""[""-""<n>];
 $N
")
```

```
DEFINE(INITIALISEBINDINGS, "
                 REQ  #binding-name                  {$PREFIX},
                 REQ  #binding-value                 {ANY_STRING}
                |REQ  #binding-name                  {$SUFFIX},
                 REQ  #binding-value                 {ANY_STRING}
                |REQ  #binding-name                  {$SEPARATOR},
                 REQ  #binding-value                  {ANY_STRING}
                |REQ  #binding-name                   {$NUMBER},
                 REQ  #binding-value                  {ANY_INTEGER}
                |REQ  #binding-name                   {$NUMBERSTRING},
                 REQ  #binding-value                  {ANY_STRING}
                |REQ  #binding-name                   {$STRING},
                 REQ  #binding-value                  {ANY_STRING}
")
```

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

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

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

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

$NUMBERSTRING
$SEPARATOR
$NUMBER}

")
```

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

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

$NUMBER}
")
```

-- Used to initialize/specify or manipulate the bindings. The bindings defined by this macro are permitted to:

- *any logical object class,*
- *any logical object,*
- *any layout object class except frame classes and block classes.*

--

```
DEFINE(SPECIFYBINDINGS, "
 $INITIALISEBINDINGS | $USENUMBERS | $USENUMBERSTRINGS
")
```

-- Used to initialize fnotenumber and fnotestring bindings. --

-- Note that footnote numbering is realized as a particular case of the general number binding mechanism supported by this DAP using the bindings <number> and <numberstring>. “fnotenumber” and “fnotestring” may be used for the compatibility with FOD26. --

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

-- Used to increment fnotenumber binding. --

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

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

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

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

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

-- Used to initialise PGnum binding. --

-- Note that a page numbering is realized as a particular case of the general number binding mechanism supported by this DAP using the bindings <number> and <numberstring>. “PGnum” may be used for the compatibility with FOD26. --

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

-- Used to increment PGnum binding. --

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

-- This string expression is allowed in a content generator for Number to automatically generate text for segment numbers, figure numbers or list numbers. (Note: B_REF(CURR-OBJ) is used for list numbers.) --

```
DEFINE(SEGMENTNUMBER, "
 <string-expr>  ::=  [<pre-str>]<num-str>[<suf-str>];
 <num-str>           ::=  B_REF(SUP(CURR-OBJ))(<numberstring>)
                          | B_REF(CURR-OBJ)(<numberstring>);
 <pre-str>           ::=  B_REF(SUP(CURR-OBJ))(<prefix>) | ANY_STRING;
 <suf-str>           ::=  B_REF(SUP(CURR-OBJ))(<suffix>) | ANY_STRING;
 $NUMBERSTRING
 $PREFIX
 $SUFFIX
")
```

-- This string expression is allowed in a content generator for TableNumber to automatically generate text for a table number. --

```
DEFINE(TABLENUMBER, "
 <string-expr>  ::=  [<pre-str>]<num-str>[<suf-str>];
 <num-str>      ::=  B_REF
                      (CURR-INST(OBJECT_CLASS_ID_OF(Table),(CURR-OBJ)))
                      (<numberstring>);
 <pre-str>      ::=  B_REF(SUP
                      (CURR-INST(OBJECT_CLASS_ID_OF(Table),(CURR-OBJ))))
                      (<prefix>)
                     | ANY_STRING;
<suf-str>        ::=  B_REF(SUP
                        (CURR-INST(OBJECT_CLASS_ID_OF(Table),(CURR-OBJ))))
                        (<suffix>)
                       | ANY_STRING;
 $NUMBERSTRING
 $PREFIX
 $SUFFIX
")
```

-- This string expression is allowed in a content generator for PageNumber to automatically generate text for a page number. --

-- Note that a page number may be generated either from <number> or <numberstring> or from PGnum. PGnum is kept for the compatibility with FOD26. --

```
DEFINE(PGNUMBER, "
 <string-expr>  ::= [<pre-str>]<num-str>[<suf-str>];
 <pre-str>      ::= B_REF(SUP(<current-layout-object>))(<prefix>) | ANY_STRING;
 <suf-str>      ::= B_REF(SUP(<current-layout-object>))(<suffix>) | ANY_STRING;

 <num-str>      ::= MK-STR(<numeric-expr>)
                  | U-ALPHA(<numeric-expr>)
                  | L-ALPHA(<numeric-expr>)
                  | U-ROM(<numeric-expr>)
                  | L-ROM(<numeric-expr>)
                  | B_REF(SUP(<layout-object-1>))(<numberstring>)
                  | B_REF(<layout-object-2>)(<numberstring>);

 <numeric-expr>   ::= B_REF(SUP(<layout-object-1>))(<number>)
                  | B_REF(SUP(<layout-object-1>))(""PGnum"")
                  | B_REF(<layout-object-2>)(<number>)
                  | B_REF(<layout-object-2>)(""PGnum"");

 <current-layout-object>  ::= <layout-object-1> | <layout-object-2>;
 <layout-object-1>        ::= CURR-INST(<class-or-type-1>,(CURR-OBJ));
 <layout-object-2>        ::= CURR-INST(<class-or-type-2>,(CURR-OBJ));
 <class-or-type-1>        ::= 'frame';
 <class-or-type-2>        ::= 'page'
                            | OBJECT_CLASS_ID_OF(Page)
                            | OBJECT_CLASS_ID_OF(RectoPage)
                            | OBJECT_CLASS_ID_OF(VersoPage);
 $PREFIX
 $SUFFIX
 $NUMBER
 $NUMBERSTRING
")
```

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

-- Note that a footnote number may be generated either from <numberstring> or from "fnotestring". "fnotestring" is kept for the compatibility with FOD26. --

```
DEFINE(FNNUMBER, "
 <string-expr>       ::=   [<pre-str>]<num-str>[<suf-str>];
 <num-str>                 ::=      B_REF(SUP(CURR-OBJ))(<numberstring>)
                             | B_REF(SUP(CURR-OBJ))(""fnotestring"")
                             | ANY_STRING;
 <pre-str>                 ::=      B_REF(SUP(CURR-OBJ))(<prefix>)
                             | ANY_STRING;
 <suf-str>                 ::=      B_REF(SUP(CURR-OBJ))(<suffix>)
                             | ANY_STRING;
 $NUMBERSTRING
 $PREFIX
 $SUFFIX
")
```

-- This string expression is allowed in a content generator for ReferencedContent to automatically generate text for references such as to segment numbers, table numbers, figure numbers, list numbers, footnote numbers and <string> bindings associated with a referring (i.e. a target) logical object, or to page numbers, pageset numbers and <string> bindings associated with a layout object in which the referring logical object is laid out. --

```
DEFINE(REF, "
 <string-expr>          ::= [<pre-str>]<ref-str>[<suf-str>];

-- These are a prefix and a suffix of ReferencedContent itself, not those of referring text. e.g. '(See' and ')'. --

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

 <ref-str>    ::= { <ref-numberstring>
                  | <ref-fnotestring>
                  | <ref-pgnum>
                  | <ref-number>
                  | <ref-string>
                  | ANY_STRING }... ;

 <ref-numberstring> ::= [<pre-str-a>] <ref-str-a> [<suf-str-a>];
 <pre-str-a> ::= B_REF(SUP(<target-object-1>))(<prefix>)
                  | B_REF(<target-object>)(<prefix>) ;
 <suf-str-a> ::= B_REF(SUP(<target-object-1>))(<suffix>)
                  | B_REF(<target-object>)(<suffix>) ;
 <ref-str-a> ::= B_REF(SUP(<target-object-1>))(<numberstring>)
                  | B_REF(<target-object>)(<numberstring>);

 <ref-fnotestring> ::= [<pre-str-b>] <ref-str-b> [<suf-str-b>];
 <pre-str-b> ::= B_REF(SUP(<target-logical-object-1>))(<prefix>)
                  | B_REF(<target-logical-object>)(<prefix>) ;
 <suf-str-b> ::= B_REF(SUP(<target-logical-object-1>))(<suffix>)
                  | B_REF(<target-logical-object>)(<suffix>) ;
 <ref-str-b> ::= B_REF(SUP(<target-logical-object-1>))(""fnotestring"")
                  | B_REF(<target-logical-object>)(""fnotestring"");

 <ref-pgnum> ::= [<pre-str-c>] <ref-str-c> [<suf-str-c>];
 <pre-str-c> ::= B_REF(SUP(<target-layout-object>))(<prefix>) ;
 <suf-str-c> ::= B_REF(SUP(<target-layout-object>))(<suffix>) ;
 <ref-str-c> ::=   MK-STR(B_REF(SUP(<layout-object-1>))(""PGnum""))
                  | U-ALPHA(B_REF(SUP(<layout-object-1>))(""PGnum""))
                  | L-ALPHA(B_REF(SUP(<layout-object-1>))(""PGnum""))
                  | U-ROM(B_REF(SUP(<layout-object-1>))(""PGnum""))
                  | L-ROM(B_REF(SUP(<layout-object-1>))(""PGnum""))
                  | MK-STR(B_REF(<layout-object-2>)(""PGnum""))
                  | U-ALPHA(B_REF(<layout-object-2>)(""PGnum""))
                  | L-ALPHA(B_REF(<layout-object-2>)(""PGnum""))
                  | U-ROM(B_REF(<layout-object-2>)(""PGnum""))
                  | L-ROM(B_REF(<layout-object-2>)(""PGnum""));

 <ref-number> ::= [<pre-str-d>] <ref-str-d> [<suf-str-d>];
 <pre-str-d> ::= B_REF(SUP(<target-object-1>))(<prefix>)
                  | B_REF(<target-object>)(<prefix>) ;
 <suf-str-d> ::= B_REF(SUP(<target-object-1>))(<suffix>)
                  | B_REF(<target-object>)(<suffix>) ;
 <ref-str-d> ::=   MK-STR(B_REF(SUP(<target-object-1>))(<number>))
                  | U-ALPHA(B_REF(SUP(<target-object-1>))(<number>))
                  | L-ALPHA(B_REF(SUP(<target-object-1>))(<number>))
                  | U-ROM(B_REF(SUP(<target-object-1>))(<number>))
                  | L-ROM(B_REF(SUP(<target-object-1>))(<number>))
                  | MK-STR(B_REF(<target-object>)(<number>))
                  | U-ALPHA(B_REF(<target-object>)(<number>))
                  | L-ALPHA(B_REF(<target-object>)(<number>))
                  | U-ROM(B_REF(<target-object>)(<number>))
                  | L-ROM(B_REF(<target-object>)(<number>));

 <ref-string> ::= [<pre-str-e>] <ref-str-e> [<suf-str-e>];
 <pre-str-e> ::= B_REF(SUP(<target-object-1>))(<prefix>)
                  | B_REF(<target-object>)(<prefix>) ;
 <suf-str-e> ::= B_REF(SUP(<target-object-1>))(<suffix>)
                  | B_REF(<target-object>)(<suffix>) ;
 <ref-str-e> ::= B_REF(SUP(<target-object-1>))(<string>)
                  | B_REF(<target-object>)(<string>);
 <target-object> ::= <target-logical-object> | <target-layout-object>;
 <target-object-1> ::= <target-logical-object-1> | <target-layout-object>;

 <target-logical-object> ::= <logical-objects> | CURR-INST(<class-or-type-logical>,<logical-objects>);
 <target-logical-object-x> ::= <logical-objects> | CURR-INST(<class-or-type-logical>,<logical-objects>));
 <target-logical-object-1> ::= CURR-INST(<class-or-type-logical>,<logical-objects>);
 <class-or-type-logical> ::= <logical-object-classes>
                             | 'composite-logical-object'
                             | 'basic-logical-object';

 <target-layout-object> ::= <layout-object-1> | <layout-object-2>;
 <layout-object-1> ::= CURR-INST(<class-or-type-layout-1>,<target-logical-object-x>);
 <layout-object-2> ::= CURR-INST(<class-or-type-layout-2>,<target-logical-object-x>);
 <class-or-type-layout-1> : <class-or-type-layout-1> ::= 'frame';
 <class-or-type-layout-2> ::= 'page'
                             | OBJECT_CLASS_ID_OF(Page)
                             | OBJECT_CLASS_ID_OF(RectoPage)
                             | OBJECT_CLASS_ID_OF(VersoPage);
 $PREFIX
 $SUFFIX
 $NUMBERSTRING
 $NUMBER
 $STRING
 $LogicalObjects
 $LogicalObjectClasses
 ")
```

-- This string expression is allowed in a content generator for CommonReference to automatically generate text for references such as to segment numbers, table numbers, figure numbers, list numbers, footnote numbers and <string> bindings associated with a logical object which is laid out in a current layout object, or to page numbers, pageset numbers and <string> bindings associated with a current or a superior layout object. --

```
DEFINE(COMMONREF, "
<string-expr>  ::= [<pre-str>]<ref-str>[<suf-str>];

<pre-str>::= B_REF(SUP(<current-layout-object>))(<prefix>) | ANY_STRING;
<suf-str> ::= B_REF(SUP(<current-layout-object>))(<suffix>) | ANY_STRING;

<ref-str> ::= { <ref-numberstring>
               | <ref-fnotestring>
               | <ref-pgnum>
               | <ref-number>
               | <ref-string>
               | ANY_STRING   } ... ;

<ref-numberstring> ::= [<pre-str-a>] <ref-str-a> [<suf-str-a>];
<pre-str-a> ::= B_REF(SUP(<current-object>))(<prefix>)
                    | B_REF(<current-object>)(<prefix>) ;
<suf-str-a> ::= B_REF(SUP(<current-object>))(<suffix>)
                    | B_REF(<current-object>)(<suffix>) ;
<ref-str-a> ::= B_REF(SUP(<current-object>))(<numberstring>)
                    | B_REF(<current-object>)(<numberstring>);

<ref-fnotestring> ::= [<pre-str-b>] <ref-str-b> [<suf-str-b>];
<pre-str-b> ::= B_REF(SUP(<current-logical-object>))(<prefix>)
                    | B_REF(<current-logical-object>)(<prefix>) ;
<suf-str-b> ::= B_REF(SUP(<current-logical-object>))(<suffix>)
                    | B_REF(<current-logical-object>)(<suffix>) ;
<ref-str-b> ::= B_REF(SUP(<current-logical-object>))(""fnotestring"")
                    | B_REF(<current-logical-object>)(""fnotestring"");

<ref-pgnum> ::= [<pre-str-c>] <ref-str-c> [<suf-str-c>];
<pre-str-c> ::= B_REF(SUP(<current-layout-object>))(<prefix>) ;
<suf-str-c> ::= B_REF(SUP(<current-layout-object>))(<suffix>) ;
<ref-str-c> ::=          MK-STR(B_REF(SUP(<layout-object-1>))(""PGnum""))
                            | U-ALPHA(B_REF(SUP(<layout-object-1>))(""PGnum""))
                            | L-ALPHA(B_REF(SUP(<layout-object-1>))(""PGnum""))
                            | U-ROM(B_REF(SUP(<layout-object-1>))(""PGnum""))
                            | L-ROM(B_REF(SUP(<layout-object-1>))(""PGnum""))
```

```
                   | MK-STR(B_REF(<layout-object-2>)(""PGnum""))
                   | U-ALPHA(B_REF(<layout-object-2>)(""PGnum""))
                   | L-ALPHA(B_REF(<layout-object-2>)(""PGnum""))
                   | U-ROM(B_REF(<layout-object-2>)(""PGnum""))
                   | L-ROM(B_REF(<layout-object-2>)(""PGnum""));

 <ref-number> ::= [<pre-str-d>] <ref-str-d> [<suf-str-d>];
 <pre-str-d> ::= B_REF(SUP(<current-object>))(<prefix>)
              | B_REF(<current-object>)(<prefix>) ;
 <suf-str-d> ::= B_REF(SUP(<current-object>))(<suffix>)
              | B_REF(<current-object>)(<suffix>) ;
 <ref-str-d> ::=      MK-STR(B_REF(SUP(<current-object>))(<number>))
                   | U-ALPHA(B_REF(SUP(<current-object>))(<number>))
                   | L-ALPHA(B_REF(SUP(<current-object>))(<number>))
                   | U-ROM(B_REF(SUP(<current-object>))(<number>))
                   | L-ROM(B_REF(SUP(<current-object>))(<number>))
                   | MK-STR(B_REF(<current-object>)(<number>))
                   | U-ALPHA(B_REF(<current-object>)(<number>))
                   | L-ALPHA(B_REF(<current-object>)(<number>))
                   | U-ROM(B_REF(<current-object>)(<number>))
                   | L-ROM(B_REF(<current-object>)(<number>));

 <ref-string> ::= [<pre-str-e>] <ref-str-e> [<suf-str-e>];
 <pre-str-e> ::= B_REF(SUP(<current-object>))(<prefix>)
              | B_REF(<current-object>)(<prefix>) ;
 <suf-str-e> ::= B_REF(SUP(<current-object>))(<suffix>)
              | B_REF(<current-object>)(<suffix>) ;
 <ref-str-e> ::= B_REF(SUP(<current-object>))(<string>)
              | B_REF(<current-object>)(<string>);

 <current-object> ::= <current-logical-object> | <current-layout-object>;

 <current-logical-object> ::= CURR-INST(<class-or-type-logical>,(CURR-OBJ));
 <class-or-type-logical> ::= <logical-object-classes>
                   | ’composite-logical-object’
                   | ’basic-logical-object’;

 <current-layout-object> ::=        <layout-object-1> | <layout-object-2>;
 <layout-object-1> ::= CURR-INST(<class-or-type-layout-1>,(CURR-OBJ));
 <layout-object-2> ::= CURR-INST(<class-or-type-layout-2>,(CURR-OBJ));
 <class-or-type-layout-1> ::= ’frame’;
 <class-or-type-layout-2> ::= ’page’
                   | OBJECT_CLASS_ID_OF(Page)
                   | OBJECT_CLASS_ID_OF(RectoPage)
                   | OBJECT_CLASS_ID_OF(VersoPage);

 $PREFIX
 $SUFFIX
 $NUMBERSTRING
 $NUMBER
 $STRING
 $LogicalObjectClasses
")
```

-- This string expression is allowed in a content generator for CommonNumber to automatically generate text for general references to any kinds of numbers including segment numbers, table numbers, figure numbers, list numbers, footnote numbers, page numbers or pageset numbers, etc. --

```
DEFINE(COMMONNUMBER, "
 <string-expr>  ::= <ref-numberstring>
                 | <ref-number>;

 <ref-numberstring> ::= [<pre-str-a>] <ref-str-a> [<suf-str-a>];
 <pre-str-a> ::= B_REF(SUP(<current-object>))(<prefix>)
              | B_REF(<current-object>)(<prefix>) | ANY_STRING;
 <suf-str-a> ::= B_REF(SUP(<current-object>))(<suffix>)
              | B_REF(<current-object>)(<suffix>) | ANY_STRING;
 <ref-str-a> ::= B_REF(SUP(<current-object>))(<numberstring>)
              | B_REF(<current-object>)(<numberstring>);
 <ref-number> ::= [<pre-str-b>] <ref-str-b> [<suf-str-b>];
 <pre-str-b> ::= B_REF(SUP(<current-object>))(<prefix>)
              | B_REF(<current-object>)(<prefix>) | ANY_STRING;
 <suf-str-b> ::= B_REF(SUP(<current-object>))(<suffix>)
              | B_REF(<current-object>)(<suffix>) | ANY_STRING;
 <ref-str-b> ::=  MK-STR(B_REF(SUP(<current-object>))(<number>))
              | U-ALPHA(B_REF(SUP(<current-object>))(<number>))
              | L-ALPHA(B_REF(SUP(<current-object>))(<number>))
              | U-ROM(B_REF(SUP(<current-object>))(<number>))
              | L-ROM(B_REF(SUP(<current-object>))(<number>))
              | MK-STR(B_REF(<current-object>)(<number>))
              | U-ALPHA(B_REF(<current-object>)(<number>))
              | L-ALPHA(B_REF(<current-object>)(<number>))
              | U-ROM(B_REF(<current-object>)(<number>))
              | L-ROM(B_REF(<current-object>)(<number>));

 <current-object>     ::= <current-logical-object> | <current-layout-object>;


 <current-logical-object>  ::= CURR-INST(<class-or-type-logical>,(CURR-OBJ));
 <class-or-type-logical>    ::= <logical-object-classes>
                                 | 'composite-logical-object'
                                 | 'basic-logical-object';

 <current-layout-object>  ::= CURR-INST(<class-or-type-layout>,(CURR-OBJ));
 <class-or-type-layout>    ::= 'frame'
                                 | 'page'
                                 | OBJECT_CLASS_ID_OF(Page)
                                 | OBJECT_CLASS_ID_OF(RectoPage)
                                 | OBJECT_CLASS_ID_OF(VersoPage);

 $PREFIX
 $SUFFIX
 $NUMBERSTRING
 $NUMBER
 $LogicalObjectClasses
")
```

-- *This string expression is allowed in a content generator for CurrentInstance to automatically generate text for general references to <string> bindings associated with a current logical or layout object.* --

```
DEFINE(CURRENTINSTANCE, "
 <string-expr>  ::= [<pre-str>]<ref-str>[<suf-str>];
 <pre-str>        ::= B_REF(SUP(<current-object>))(<prefix>)
                       | B_REF(<current-object>)(<prefix>) | ANY_STRING;
 <suf-str>        ::= B_REF(SUP(<current-object>))(<suffix>)
                       | B_REF(<current-object>)(<suffix>) | ANY_STRING;
 <ref-str>        ::= B_REF(SUP(<current-object>))(<string>)
                       | B_REF(<current-object>)(<string>);

 <current-object>     ::= <current-logical-object> | <current-layout-object>;

 <current-logical-object>  ::= CURR-INST(<class-or-type-logical>,(CURR-OBJ));
 <class-or-type-logical>    ::= <logical-object-classes>
                                 | 'composite-logical-object'
                                 | 'basic-logical-object';

 <current-layout-object>  ::= CURR-INST(<class-or-type-layout>,(CURR-OBJ));
 <class-or-type-layout>    ::= 'frame'
                                 | 'page'
                                 | OBJECT_CLASS_ID_OF(Page)
                                 | OBJECT_CLASS_ID_OF(RectoPage)
                                 | OBJECT_CLASS_ID_OF(VersoPage);
 $PREFIX
 $SUFFIX
 $STRING
 $LogicalObjectClasses
")
```

-- This string expression is allowed in a content generator for GenericBlock to automatically generate text for general references to bindings associated with a current layout object. --

```
DEFINE(GENERICBLOCKREF, "
 <string-expr>  ::= [<pre-str>]<ref-str>[<suf-str>];

 <pre-str>       ::= B_REF(SUP(CURR-OBJ))(<prefix>) | ANY_STRING;
 <suf-str>       ::= B_REF(SUP(CURR-OBJ))(<suffix>) | ANY_STRING;
 <ref-str>       ::=     MK-STR(B_REF(SUP(CURR-OBJ))(<number>))
                       | U-ALPHA(B_REF(SUP(CURR-OBJ))(<number>))
                       | L-ALPHA(B_REF(SUP(CURR-OBJ))(<number>))
                       | U-ROM(B_REF(SUP(CURR-OBJ))(<number>))
                       | L-ROM(B_REF(SUP(CURR-OBJ))(<number>))
                       | MK-STR(B_REF(SUP(CURR-OBJ))(""PGnum""))
                       | U-ALPHA(B_REF(SUP(CURR-OBJ))(""PGnum""))
                       | L-ALPHA(B_REF(SUP(CURR-OBJ))(""PGnum""))
                       | U-ROM(B_REF(SUP(CURR-OBJ))(""PGnum""))
                       | L-ROM(B_REF(SUP(CURR-OBJ))(""PGnum""))
                       | B_REF(SUP(CURR-OBJ))(<numberstring>)
                       | B_REF(SUP(CURR-OBJ))(<string>);

 $PREFIX
 $SUFFIX
 $NUMBER
 $NUMBERSTRING
 $STRING
")

DEFINE(DocLogRootGFS, "
 <construction-expr>       ::=     <construction-term>
                                         |<construction-type>;

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

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

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

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

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

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

 <construction-factor>     ::=     OBJECT_CLASS_ID_OF(Passage)
                                         | OBJECT_CLASS_ID_OF(NumberedSegment)
                                         | OBJECT_CLASS_ID_OF(Paragraph)
                                         | OBJECT_CLASS_ID_OF(BodyText)
                                         | OBJECT_CLASS_ID_OF(BodyRaster)
                                         | OBJECT_CLASS_ID_OF(BodyGeometric)
                                         | OBJECT_CLASS_ID_OF(Figure)
```

```
                                    | OBJECT_CLASS_ID_OF(Table)
                                    | OBJECT_CLASS_ID_OF(NumberedList)
                                    | OBJECT_CLASS_ID_OF(UnNumberedList)
                                    | OBJECT_CLASS_ID_OF(DefinitionList)
                                    | <construction-type>;
")

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

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

 $CONSTRAINT-1
 $CONSTRAINT-2
")

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

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

 $CONSTRAINT-1
 $CONSTRAINT-2
")

DEFINE(CONSTRAINT-3,     "
 <construction-expr>     ::=     <construction-term>
                                    | <construction-type>;

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

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

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

DEFINE(ParagraphGFS, "$CONSTRAINT-3")

DEFINE(TitleGFS, "$CONSTRAINT-3")

DEFINE(FigureGFS, "
 <construction-expr>     ::=
           SEQ([<term-1>][<term-2>][<term-3>]<term-4>)
          | SEQ([<term-1>][<term-2>]<term-4>[<term-3>])
          | SEQ([<term-3>][<term-1>][<term-2>]<term-4>)
          | SEQ(<term-4>[<term-1>][<term-2>][<term-3>])
          | SEQ([<term-3>]<term-4>[<term-1>][<term-2>])
          | SEQ(<term-4>[<term-3>][<term-1>][<term-2>]);

 <term-1>        ::=   OBJECT_CLASS_ID_OF(Number)
                       | OPT OBJECT_CLASS_ID_OF(Number);
 <term-2>        ::=   OBJECT_CLASS_ID_OF(Caption)
                       | OPT OBJECT_CLASS_ID_OF(Caption);
 <term-3>        ::=   OBJECT_CLASS_ID_OF(Description)
                       | OPT OBJECT_CLASS_ID_OF(Description);
 <term-4>        ::=   OBJECT_CLASS_ID_OF(Artwork)
                       | OBJECT_CLASS_ID_OF(Form);
")
```

```
DEFINE(ArtworkGFS, "
 <construction-expr>       ::=      <construction-term>
                                          | <construction-type>;

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

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

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

DEFINE(CONSTRAINT-5,      "
 <construction-expr>       ::=      <construction-term>
                                          | <construction-type>;

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

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

 <construction-factor>     ::=      OBJECT_CLASS_ID_OF(Phrase)
                                          | OBJECT_CLASS_ID_OF(Footnote)
                                          | OBJECT_CLASS_ID_OF(Reference)
                                          | OBJECT_CLASS_ID_OF(BodyText)
                                          | <construction-type>;
")

DEFINE(PhraseGFS,   "$CONSTRAINT-5")

DEFINE(CaptionGFS,  "$CONSTRAINT-5")

DEFINE(DescriptionGFS,  "$CONSTRAINT-5")

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

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

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

DEFINE(ReferenceGFS, "
 <construction-expr>       ::=      OBJECT_CLASS_ID_OF(ReferencedContent)
                                    | SEQ([<term>]
                                          OBJECT_CLASS_ID_OF(ReferencedContent)
                                          [<term>]);
 <term> ::=                         OBJECT_CLASS_ID_OF(BodyText)
                                    | OPT OBJECT_CLASS_ID_OF(BodyText)
                                    | CHO( {OBJECT_CLASS_ID_OF(BodyText)}... );
")
```

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

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

DEFINE(TableGFS,"
 <construction-expr> ::=         REP CHO(OBJECT_CLASS_ID_OF(Row)...)
                                 | REP OBJECT_CLASS_ID_OF(Row)
                                 | SEQ(OBJECT_CLASS_ID_OF(Row)...);
")

DEFINE(RowGFS,"
 <construction-expr> ::=         <simple-table> | <complex-table>;
 <simple-table>           ::=    REP OBJECT_CLASS_ID_OF(EntryElement)
                                 | REP CHO(OBJECT_CLASS_ID_OF(EntryElement)...)
                                 | SEQ(OBJECT_CLASS_ID_OF(EntryElement)...);
 <complex-table>          ::=    SEQ(OBJECT_CLASS_ID_OF(EntryElement)
                                       OBJECT_CLASS_ID_OF(TableComponent));
")

DEFINE(TableComponentGFS,"
 <construction-expr> ::=         REP OBJECT_CLASS_ID_OF(RowComponent);
")

DEFINE(RowComponentGFS,"
 <construction-expr> ::=         REP OBJECT_CLASS_ID_OF(EntryElement)
                                 |REP CHO(OBJECT_CLASS_ID_OF(EntryElement)...)
                                 |SEQ(OBJECT_CLASS_ID_OF(EntryElement)...);
")

DEFINE(FormGFS,"
 <construction-expr> ::=         AGG(<factor>...);
 <factor>                 ::=    OBJECT_CLASS_ID_OF(EntryElement)
                                 |OBJECT_CLASS_ID_OF(EntryGroup);
")

DEFINE(EntryGroupGFS,"$FormGFS")

DEFINE(EntryElementGFS,"
 <construction-expr> ::=         OBJECT_CLASS_ID_OF(EntryText)
                                 |OBJECT_CLASS_ID_OF(EntryRaster)
                                 |OBJECT_CLASS_ID_OF(EntryGeometric);
")

DEFINE(NumberedListGFS,"
 <construction-expr> ::=         SEQ(OBJECT_CLASS_ID_OF(Number)
                                     OBJECT_CLASS_ID_OF(ListItem))
                                 | REP SEQ(OBJECT_CLASS_ID_OF(Number)
                                     OBJECT_CLASS_ID_OF(ListItem));
")

DEFINE(UnNumberedListGFS,"
 <construction-expr> ::=  OBJECT_CLASS_ID_OF(ListItem)
                                 | REP OBJECT_CLASS_ID_OF(ListItem)
                                 | SEQ(<separator-obj> OBJECT_CLASS_ID_OF(ListItem))
                                 | REP SEQ(<separator-obj> OBJECT_CLASS_ID_OF(ListItem));

 <separator-obj>       ::= OBJECT_CLASS_ID_OF(BodyText)
                                 |OBJECT_CLASS_ID_OF(BodyRaster)
                                 |OBJECT_CLASS_ID_OF(BodyGeometric);
")
```

```
DEFINE(DefinitionListGFS,"
 <construction-expr>     ::= SEQ(OBJECT_CLASS_ID_OF(ListTerm)
                         OBJECT_CLASS_ID_OF(ListItem))
                         |REP SEQ(OBJECT_CLASS_ID_OF(ListTerm)
                         OBJECT_CLASS_ID_OF(ListItem));
")

DEFINE(ListItemGFS,"
 <construction-expr>     ::= <term> | CHO(<term>...) ;

 <term>                  ::= REP OBJECT_CLASS_ID_OF(Phrase)
                          | OBJECT_CLASS_ID_OF(NumberedList)
                          | OBJECT_CLASS_ID_OF(UnNumberedList)
                          | OBJECT_CLASS_ID_OF(DefinitionList);
")

DEFINE(ListTermGFS,"$CONSTRAINT-3")
```

7.3.2 Factor constraints

7.3.2.1 FACTOR ANY-LOGICAL

```
{
GENERIC:
        REQObject-type                  {VIRTUAL},
        REQObject-class-identifier      {ANY_VALUE}
SPECIFIC:
        PERM    Object-type                     {VIRTUAL},
        REQObject-identifier            {ANY_VALUE},
        REQObject-class                 {VIRTUAL}
SPECIFIC_AND_GENERIC:
        PERM    Protection                      {ANY_VALUE},
        PERM    User-readable-comments  {ANY_STRING},
        PERM    User-visible-name               {ANY_STRING}
}
```

7.3.2.2 FACTOR COMP-LOGICAL

```
:ANY-LOGICAL {
GENERIC:
        REQObject-type                  {'composite-logical-object'}
SPECIFIC:
        REQSubordinates                 {VIRTUAL},
        PERM    Object-type                     {'composite-logical-object'}
SPECIFIC_AND_GENERIC:
        PERM    Bindings                                {PMUL {$SPECIFYBINDINGS}},
        PERM    Default-value-lists             {REQ #basic-logical-attributes
                                                {PERM #presentation-style {ANY_VALUE},
                                                PERM #layout-style        {ANY_VALUE}}}
}
```

7.3.2.3 FACTOR BASIC-LOGICAL

```
:ANY-LOGICAL {
GENERIC:
        REQObject-type                  {'basic-logical-object'},
        PERM    Resource                                {ANY_VALUE}
SPECIFIC:
        PERM    Object-type                     {'basic-logical-object'}
SPECIFIC_AND_GENERIC:
        PERM    Bindings                                {PMUL {$SPECIFYBINDINGS}}
}
```

7.3.2.4 FACTOR ANY-COMMON

```
{
GENERIC:
          REQObject-type                  {VIRTUAL},
          REQObject-class-identifier      {ANY_VALUE},
          PERM     Bindings                                 {PMUL {$SPECIFYBINDINGS}},
          PERM     Protection                      {ANY_VALUE},
          PERM     User-readable-comments  {ANY_STRING},
          PERM     User-visible-name               {ANY_STRING}
}
```

7.3.3 Constituent constraints

7.3.3.1 DocumentLogicalRoot

```
:ANY-LOGICAL {
GENERIC:
          REQObject-type                  {'document-logical-root'},
          REQGenerator-for-subordinates   {$DocLogRootGFS},
          REQApplication-comments         {REQ #constraint-name {"0"},
                                                    PERM #external-data        {ANY_VALUE}}
SPECIFIC:
          PERM     Object-type                     {'document-logical-root'},
          REQObject-class                 {OBJECT_CLASS_ID_OF(DocumentLogicalRoot)},
          REQSubordinates                 {SUB_ID_OF(NumberedSegment)+,
                                                   SUB_ID_OF(Passage)+},
          PERM   Application-comments              {REQ #constraint-name {"0"},
                                                   PERM #external-data      {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
          PERM     Bindings                                 {PMUL {$SPECIFYBINDINGS},
                                                   PERM {$INITIALISEFNOTE}},
          PERM     Default-value-lists             {REQ #basic-logical-attributes
                                                    {PERM #presentation-style          {ANY_VALUE},
                                                    PERM #layout-style         {ANY_VALUE}}}
}
```

7.3.3.2 Passage

```
:COMP-LOGICAL {
GENERIC:
          REQGenerator-for-subordinates   {$PassageGFS},
          REQApplication-comments         {REQ #constraint-name {"1"},
                                                   PERM #external-data      {ANY_VALUE}}
SPECIFIC:
          REQObject-class                 {OBJECT_CLASS_ID_OF(Passage)},
          REQSubordinates                 {SUB_ID_OF(Title),
                                                   SUB_ID_OF(Passage)+,
                                                   SUB_ID_OF(NumberedSegment)+,
                                                   SUB_ID_OF(BodyText)+,
                                                   SUB_ID_OF(BodyRaster)+,
                                                   SUB_ID_OF(BodyGeometric)+,
                                                   SUB_ID_OF(Figure)+,
                                                   SUB_ID_OF(Paragraph)+,
                                                   SUB_ID_OF(Table)+,
                                                   SUB_ID_OF(NumberedList)+,
                                                   SUB_ID_OF(DefinitionList)+,
                                                   SUB_ID_OF(UnNumberedList)+},
          PERM     Application-comments    {REQ  #constraint-name {"1"},
                                                   PERM #external-data   {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
          PERM     Bindings                                 {PMUL {$SPECIFYBINDINGS},
                                                   PERM {$INITIALISEFNOTE}},
          PERM     Layout-style                    {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.3 NumberedSegment

```
:COMP-LOGICAL {
GENERIC:
         REQGenerator-for-subordinates   {$NumberedSegmentGFS},
         REQApplication-comments         {REQ #constraint-name {"2"},
                                                  PERM #external-data {ANY_VALUE}}
SPECIFIC:
         REQObject-class                 {OBJECT_CLASS_ID_OF(NumberedSegment)},
         REQSubordinates                 {SUB_ID_OF(Number),
                                                 SUB_ID_OF(Title),
                                                 SUB_ID_OF(Passage)+,
                                                 SUB_ID_OF(NumberedSegment)+,
                                                 SUB_ID_OF(BodyText)+,
                                                 SUB_ID_OF(BodyRaster)+,
                                                 SUB_ID_OF(BodyGeometric)+,
                                                 SUB_ID_OF(Paragraph)+,
                                                 SUB_ID_OF(Figure)+,
                                                 SUB_ID_OF(Table)+,
                                                 SUB_ID_OF(NumberedList)+,
                                                 SUB_ID_OF(DefinitionList)+,
                                                 SUB_ID_OF(UnNumberedList)+},
         PERM    Application-comments    {REQ  #constraint-name {"2"},
                                                  PERM #external-data      {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
         PERM    Layout-style                    {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.4 Number

```
:BASIC-LOGICAL {
GENERIC:
         REQContent-generator            {$SEGMENTNUMBER},
         REQApplication-comments         {REQ  #constraint-name {"3"},
                                                  PERM #external-data {ANY_VALUE}}
SPECIFIC:
         REQObject-class                 {OBJECT_CLASS_ID_OF(Number)},
         PERM    Content-generator               {$SEGMENTNUMBER},
         PERM    Application-comments    {REQ #constraint-name {"3"},
                                                  PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
         PERM    Layout-style                    {STYLE_ID_OF(LStyle2)},
         PERM    Presentation-style              {STYLE_ID_OF(PStyle1)},
         PERM    Content-architecture-class      {$FC|$PC|$FPC}
}
```

7.3.3.5 Title

```
:COMP-LOGICAL {
GENERIC:
         REQGenerator-for-subordinates   {$TitleGFS},
         REQApplication-comments         {REQ  #constraint-name {"4"},
                                                  PERM #external-data {ANY_VALUE}}
SPECIFIC:
         REQObject-class                 {OBJECT_CLASS_ID_OF(Title)},
         REQSubordinates                 {SUB_ID_OF(BodyText)+,
                                                  SUB_ID_OF(BodyRaster)+,
                                                  SUB_ID_OF(BodyGeometric)+,
                                                  SUB_ID_OF(Phrase)+,
                                                  SUB_ID_OF(Footnote)+,
                                                  SUB_ID_OF(Reference)+},
         PERM    Application-comments    {REQ #constraint-name {"4"},
                                                  PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
         PERM    Layout-style                    {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.6 Caption

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates   {$CaptionGFS},
        REQApplication-comments         {REQ #constraint-name {"5"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQObject-class                 {OBJECT_CLASS_ID_OF(Caption)},
        REQSubordinates                 {SUB_ID_OF(BodyText)+,
                                                SUB_ID_OF(Phrase)+,
                                                SUB_ID_OF(Footnote)+,
                                                SUB_ID_OF(Reference)+},
        PERM    Application-comments    {REQ #constraint-name {"5"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                    {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.7 Paragraph

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates   {$ParagraphGFS},
        REQApplication-comments         {REQ #constraint-name {"6"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQObject-class                 {OBJECT_CLASS_ID_OF(Paragraph)},
        REQSubordinates                 {SUB_ID_OF(BodyText)+,
                                                SUB_ID_OF(Footnote)+,
                                                SUB_ID_OF(BodyRaster)+,
                                                SUB_ID_OF(BodyGeometric)+,
                                                SUB_ID_OF(Phrase)+,
                                                SUB_ID_OF(Reference)+},
        PERM    Application-comments    {REQ #constraint-name {"6"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                    {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.8 Phrase

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates   {$PhraseGFS},
        REQApplication-comments         {REQ #constraint-name {"7"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQObject-class                 {OBJECT_CLASS_ID_OF(Phrase)},
        REQSubordinates                 {SUB_ID_OF(BodyText)+,
                                                SUB_ID_OF(Footnote)+,
                                                SUB_ID_OF(Phrase)+,
                                                SUB_ID_OF(Reference)+},
        PERM    Application-comments    {REQ #constraint-name {"7"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                    {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.9 Footnote

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates   {$FootnoteGFS},
        PERM    Bindings                                {PMUL {$SPECIFYBINDINGS},
                                                PERM {REQ{$INCFNOTENUMBER,$FNOTENUMBERSTRING}
                                                        | $FNOTESTRINGLITERAL}},
        REQApplication-comments         {REQ  #constraint-name {"8"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQObject-class                 {OBJECT_CLASS_ID_OF(Footnote)},
        REQSubordinates                 {SUB_ID_OF(FootnoteReference),
                                                SUB_ID_OF(FootnoteBody)},
        PERM    Bindings                                {PMUL {$SPECIFYBINDINGS},
                                                PERM {$FNOTESTRINGLITERAL}},
        PERM    Application-comments    {REQ  #constraint-name {"8"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                    {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.10 FootnoteNumber

```
:BASIC-LOGICAL {
GENERIC:
        REQContent-generator            {$FNNUMBER},
        REQApplication-comments         {REQ #constraint-name {"9"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQObject-class                 {OBJECT_CLASS_ID_OF(FootnoteNumber)},
        PERM    Content-generator               {$FNNUMBER},
        PERM    Application-comments    {REQ #constraint-name {"9"},
                                               PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                    {STYLE_ID_OF(LStyle9)},
        PERM    Presentation-style              {STYLE_ID_OF(PStyle1)},
        PERM    Content-architecture-class      {$FC|$PC|$FPC}
}
```

7.3.3.11 FootnoteReference

```
:BASIC-LOGICAL {
GENERIC:
        REQContent-generator            {$FNNUMBER},
        REQApplication-comments         {REQ #constraint-name {"10"},
                                               PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQObject-class                 {OBJECT_CLASS_ID_OF(FootnoteReference)},
        PERM    Content-generator               {$FNNUMBER},
        PERM    Application-comments    {REQ #constraint-name {"10"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                    {STYLE_ID_OF(LStyle10)},
        PERM    Presentation-style              {STYLE_ID_OF(PStyle1)},
        PERM    Content-architecture-class      {$FC|$PC|$FPC}
}
```

7.3.3.12 FootnoteBody

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates   {$FootnoteBodyGFS},
        REQApplication-comments         {REQ #constraint-name {"11"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQObject-class                 {OBJECT_CLASS_ID_OF(FootnoteBody)},
        REQSubordinates                 {SUB_ID_OF(FootnoteNumber),
                                                SUB_ID_OF(FootnoteText)+,
                                                SUB_ID_OF(Reference)+},
        PERM    Application-comments    {REQ #constraint-name {"11"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                    {STYLE_ID_OF(LStyle11)}
}
```

7.3.3.13 FootnoteText

```
:BASIC-LOGICAL {
GENERIC:
        REQApplication-comments       {REQ #constraint-name {"12"},
                                               PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQObject-class               {OBJECT_CLASS_ID_OF(FootnoteText)},
        PERM    Application-comments  {REQ #constraint-name {"12"},
                                               PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                  {STYLE_ID_OF(LStyle6)},
        PERM    Presentation-style            {STYLE_ID_OF(PStyle1)},
        PERM    Content-architecture-class    {$FC|$PC|$FPC},

        PERM    Content-portions              {CONTENT_ID_OF(Character-content-portion)+}
}
```

7.3.3.14 Figure

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates {$FigureGFS},
        REQApplication-comments       {REQ #constraint-name {"13"},
                                               PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQObject-class               {OBJECT_CLASS_ID_OF(Figure)},
        REQSubordinates               {SUB_ID_OF(Number),
                                               SUB_ID_OF(Caption),
                                               SUB_ID_OF(Description),
                                               SUB_ID_OF(Artwork),
                                               SUB_ID_OF(Form)},
        PERM    Application-comments  {REQ #constraint-name {"13"},
                                               PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                  {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.15 BodyText

```
:BASIC-LOGICAL {
GENERIC:
        REQApplication-comments       {REQ #constraint-name {"14"},
                                               PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQObject-class               {OBJECT_CLASS_ID_OF(BodyText)},
        PERM    Application-comments  {REQ #constraint-name {"14"},
                                               PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                  {STYLE_ID_OF(LStyle2)},
        PERM    Presentation-style            {STYLE_ID_OF(PStyle1)},
        PERM    Content-architecture-class    {$FC|$PC|$FPC},
        PERM    Content-portions              {CONTENT_ID_OF(Character-content-portion)+}
        -- If the attribute "content portions" is specified neither in the specific nor in the generic part, then the
        attribute "resource" shall be specified. --
}
```

7.3.3.16 Reference

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates {$ReferenceGFS},
        REQApplication-comments       {REQ #constraint-name {"15"},
                                               PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQ Object-class                {OBJECT_CLASS_ID_OF(Reference)},
        REQ Subordinates                {SUB_ID_OF(BodyText)+,
                                                SUB_ID_OF(ReferencedContent)},
        PERM    Application-comments    {REQ #constraint-name {"15"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                    {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.17 ReferencedContent

```
:BASIC-LOGICAL {
GENERIC:
        REQ Application-comments        {REQ #constraint-name {"16"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQ Object-class                {OBJECT_CLASS_ID_OF(ReferencedContent)},
        PERM    Content-generator               {$REF},
        PERM    Content-portions                {CONTENT_ID_OF(Character-content-portion)+},
                                        -- Either Content-generator or Content-portions is specified. --
        PERM    Application-comments    {REQ  #constraint-name {"16"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:

        PERM    Layout-style                    {STYLE_ID_OF(LStyle10)},
        PERM    Presentation-style              {STYLE_ID_OF(PStyle1)},
        PERM    Content-architecture-class      {$FC|$PC|$FPC}
}
```

7.3.3.18 BodyRaster

```
:BASIC-LOGICAL {
GENERIC:
        REQ Content-architecture-class  {$FPR},
        REQ Application-comments        {REQ #constraint-name {"17"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQ Object-class                {OBJECT_CLASS_ID_OF(BodyRaster)},
        PERM    Content-architecture-class      {$FPR},
        PERM    Application-comments    {REQ #constraint-name {"17"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                    {STYLE_ID_OF(LStyle5)},
        PERM    Presentation-style              {STYLE_ID_OF(PStyle3)},
        PERM    Content-portions                {CONTENT_ID_OF(Raster-graphics-content-portion)}
        -- If the attribute “content portions” is specified neither in the specific nor in the generic part, then the
        attribute “resource” shall be specified. --
}
```

7.3.3.19 BodyGeometric

```
:BASIC-LOGICAL {
GENERIC:
        REQ Content-architecture-class  {$FPG},
        REQ Application-comments        {REQ #constraint-name {"18"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQ Object-class                {OBJECT_CLASS_ID_OF(BodyGeometric)},
        PERM    Content-architecture-class      {$FPG},
        PERM    Application-comments    {REQ #constraint-name {"18"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                    {STYLE_ID_OF(LStyle5)},
        PERM    Presentation-style              {STYLE_ID_OF(PStyle2)},
        PERM    Content-portions                {CONTENT_ID_OF(Geometric-graphics-content-portion)}
        -- If the attribute “content portions” is specified neither in the specific nor in the generic part, then the
        attribute "resource" shall be specified. --
}
```

7.3.3.20 CommonContent

```
:ANY-COMMON {
GENERIC:
            REQObject-type                  {'composite-logical-object'},
            REQGenerator-for-subordinates   {$CommonContentGFS},
            REQApplication-comments         {REQ #constraint-name {"19"},
                                                         PERM #external-data {ANY_VALUE}},
            PERM      Default-value-lists             {REQ #basic-logical-attributes
                                                                  {PERM #presentation-style           {ANY_VALUE},
                                                                   PERM #layout-style           {ANY_VALUE}}}
}
```

7.3.3.21 CommonText

```
:ANY-COMMON {
GENERIC:
            REQObject-type                  {'basic-logical-object'},
            PERM      Content-portions                {CONTENT_ID_OF(Character-content-portion)+},
            PERM      Resource                                  {ANY_VALUE},
            PERM      Layout-style                    {STYLE_ID_OF(LStyle3)},
            PERM      Presentation-style              {STYLE_ID_OF(PStyle4)},
            PERM      Content-architecture-class      {$FC|$PC|$FPC},
            REQApplication-comments         {REQ #constraint-name {"20"},
                                                         PERM #external-data {ANY_VALUE}}
```

-- *Either the attribute "content portions" or "resource" shall be specified in the above constituent constraint.* --

```
}
```

7.3.3.22 CommonReference

```
:ANY-COMMON {
GENERIC:
            REQObject-type                  {'basic-logical-object'},
            PERM      Content-generator               {$COMMONREF},
            PERM      Layout-style                    {STYLE_ID_OF(LStyle3)},
            PERM      Presentation-style              {STYLE_ID_OF(PStyle4)},
            PERM      Content-architecture-class      {$FC|$PC|$FPC},
            REQApplication-comments         {REQ #constraint-name {"37"},
                                                         PERM #external-data {ANY_VALUE}}
}
```

7.3.3.23 CommonNumber

```
:ANY-COMMON {
GENERIC:
            REQObject-type                  {'basic-logical-object'},
            PERM      Content-generator               {$COMMONNUMBER},
            PERM      Layout-style                    {STYLE_ID_OF(LStyle3)},
            PERM      Presentation-style              {STYLE_ID_OF(PStyle4)},
            PERM      Content-architecture-class      {$FC|$PC|$FPC},
            REQApplication-comments         {REQ #constraint-name {"38"},
                                                         PERM #external-data {ANY_VALUE}}
}
```

7.3.3.24 CurrentInstance

```
:ANY-COMMON {
GENERIC:
            REQObject-type                  {'basic-logical-object'},
            PERM      Content-generator               {$CURRENTINSTANCE},
            PERM      Layout-style                    {STYLE_ID_OF(LStyle3)},
            PERM      Presentation-style              {STYLE_ID_OF(PStyle4)},
            PERM      Content-architecture-class      {$FC|$PC|$FPC},
            REQApplication-comments         {REQ #constraint-name {"39"},
                                                         PERM #external-data {ANY_VALUE}}
}
```

7.3.3.25 CommonRaster

```
:ANY-COMMON {
GENERIC:
        REQObject-type                  {'basic-logical-object'},
        PERM    Content-portions                {CONTENT_ID_OF(Raster-graphics-content-portion)},
        PERM    Resource                                {ANY_VALUE},
        PERM    Layout-style                    {STYLE_ID_OF(LStyle8)},
        PERM    Presentation-style              {STYLE_ID_OF(PStyle3)},
        REQContent-architecture-class   {$FPR},
        REQApplication-comments         {REQ #constraint-name {"21"},
                                                PERM #external-data {ANY_VALUE}}
        -- Either the attribute "content portions" or "resource" shall be specified in the above constituent
        constraint. --
}
```

7.3.3.26 CommonGeometric

```
:ANY-COMMON {
GENERIC:
        REQObject-type                  {'basic-logical-object'},
        PERM    Content-portions                {CONTENT_ID_OF(Geometric-graphics-content-portion)},
        PERM    Resource                                {ANY_VALUE},
        PERM    Layout-style                    {STYLE_ID_OF(LStyle8)},
        PERM    Presentation-style              {STYLE_ID_OF(PStyle2)},
        REQContent-architecture-class   {$FPG},
        REQApplication-comments         {REQ #constraint-name {"22"},
                                                PERM #external-data {ANY_VALUE}}
        -- Either the attribute "content portions" or "resource" shall be specified in the above constituent
        constraint --
}
```

7.3.3.27 PageNumber

```
:ANY-COMMON {
GENERIC:
        REQObject-type                  {'basic-logical-object'},
        PERM    Content-generator               {$PGNUMBER},
        PERM    Layout-style                    {STYLE_ID_OF(LStyle3)},
        PERM    Presentation-style              {STYLE_ID_OF(PStyle4)},
        PERM    Content-architecture-class      {$FC|$PC|$FPC},
        REQApplication-comments         {REQ #constraint-name {"40"},
                                                PERM #external-data {ANY_VALUE}}
}
```

7.3.3.28 TableNumber

```
:ANY-COMMON {
GENERIC:
        REQObject-type                  {'basic-logical-object'},
        PERM    Content-generator               {$TABLENUMBER},
        PERM    Layout-style                    {STYLE_ID_OF(LStyle3)},
        PERM    Presentation-style              {STYLE_ID_OF(PStyle4)},
        PERM    Content-architecture-class      {$FC|$PC|$FPC},
        REQApplication-comments         {REQ #constraint-name {"44"},
                                                PERM #external-data {ANY_VALUE}}
}
```

7.3.3.29 Description

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates   {$DescriptionGFS},
        REQApplication-comments         {REQ #constraint-name {"23"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQ Object-class                 {OBJECT_CLASS_ID_OF(Description)},
        REQ Subordinates                 {SUB_ID_OF(BodyText)+,
                                                 SUB_ID_OF(Footnote)+,
                                                 SUB_ID_OF(Phrase)+,
                                                 SUB_ID_OF(Reference)+},
        PERM    Application-comments     {REQ #constraint-name {"23"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                     {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.30 Artwork

```
:COMP-LOGICAL {
GENERIC:
        REQ Generator-for-subordinates   {$ArtworkGFS},
        REQ Application-comments         {REQ #constraint-name {"24"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQ Object-class                 {OBJECT_CLASS_ID_OF(Artwork)},
        REQ Subordinates                 {SUB_ID_OF(BodyRaster)+,
                                                 SUB_ID_OF(BodyGeometric)+,
                                                 SUB_ID_OF(Phrase)+},
        PERM    Application-comments     {REQ #constraint-name {"24"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                     {STYLE_ID_OF(LStyle12)}
}
```

7.3.3.31 NumberedList

```
:COMP-LOGICAL {
GENERIC:
        REQ Generator-for-subordinates   {$NumberedListGFS},
        REQ Application-comments         {REQ #constraint-name {"25"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQ Object-class                 {OBJECT_CLASS_ID_OF(NumberedList)},
        REQ Subordinates                 {SUB_ID_OF(Number)+,
                                                 SUB_ID_OF(ListItem)+},
        PERM    Application-comments     {REQ  #constraint-name {"25"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                     {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.32 UnNumberedList

```
:COMP-LOGICAL {
GENERIC:
        REQ Generator-for-subordinates   {$UnNumberedListGFS},
        REQ Application-comments         {REQ #constraint-name {"26"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC:
        REQ Object-class                 {OBJECT_CLASS_ID_OF(UnNumberedList)},
        REQ Subordinates                 {SUB_ID_OF(BodyText)+,
                                                 SUB_ID_OF(BodyRaster)+,
                                                 SUB_ID_OF(BodyGeometric)+,
                                                 SUB_ID_OF(ListItem)+},
        PERM    Application-comments     {REQ  #constraint-name {"26"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                     {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.33 DefinitionList

```
:COMP-LOGICAL {
GENERIC:
          REQGenerator-for-subordinates    {$DefinitionListGFS},
          REQApplication-comments          {REQ  #constraint-name {"27"},
                                                     PERM #external-data {ANY_VALUE}}
SPECIFIC:
          REQObject-class                  {OBJECT_CLASS_ID_OF(DefinitionList)},
          REQSubordinates                  {SUB_ID_OF(ListTerm)+,
                                                     SUB_ID_OF(ListItem)+},
          PERM     Application-comments    {REQ  #constraint-name {"27"},
                                                     PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
          PERM     Layout-style                     {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.34 ListItem

```
:COMP-LOGICAL {
GENERIC:
          REQGenerator-for-subordinates    {$ListItemGFS},
          REQApplication-comments          {REQ #constraint-name {"28"},
                                                     PERM #external-data {ANY_VALUE}}
SPECIFIC:
          REQObject-class                  {OBJECT_CLASS_ID_OF(ListItem)},
          REQSubordinates                  {SUB_ID_OF(Phrase)+,
                                                     SUB_ID_OF(NumberedList)+,
                                                     SUB_ID_OF(UnNumberedList)+,
                                                     SUB_ID_OF(DefinitionList)+},
          PERM     Application-comments    {REQ  #constraint-name {"28"},
                                                     PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
          PERM     Layout-style                     {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.35 ListTerm

```
:COMP-LOGICAL {
GENERIC:
          REQGenerator-for-subordinates    {$ListTermGFS},
          REQApplication-comments          {REQ  #constraint-name {"29"},
                                                     PERM #external-data {ANY_VALUE}}
SPECIFIC:
          REQObject-class                  {OBJECT_CLASS_ID_OF(ListTerm)},
          REQSubordinates                  {SUB_ID_OF(BodyText)+,
                                                     SUB_ID_OF(BodyRaster)+,
                                                     SUB_ID_OF(BodyGeometric)+,
                                                     SUB_ID_OF(Reference)+,
                                                     SUB_ID_OF(Phrase)+,
                                                     SUB_ID_OF(Footnote)+},
          PERM     Application-comments    {REQ  #constraint-name {"29"},
                                                     PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
          PERM     Layout-style                     {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.36 Table

```
:COMP-LOGICAL {
GENERIC:
          REQGenerator-for-subordinates    {$TableGFS},
          REQApplication-comments          {REQ  #constraint-name {"30"},
                                                     PERM #external-data {ANY_VALUE}},
          REQLayout-style                  {STYLE_ID_OF(LStyleT4)}
SPECIFIC:
        REQObject-class              {OBJECT_CLASS_ID_OF(Table)},
        REQSubordinates              {SUB_ID_OF(Row)+},
        PERM    Application-comments {REQ  #constraint-name {"30"},
                                             PERM #external-data {ANY_VALUE}},
        PERM    Layout-style                {STYLE_ID_OF(LStyleT8)}
}
```

7.3.3.37 Row

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates  {$RowGFS},
        REQApplication-comments        {REQ #constraint-name {"31"},
                                               PERM #external-data {ANY_VALUE}},
        REQLayout-style                {STYLE_ID_OF(LStyleT5)}
SPECIFIC:
        REQObject-class                {OBJECT_CLASS_ID_OF(Row)},
        REQSubordinates                {SUB_ID_OF(EntryElement)+,
                                               SUB_ID_OF(TableComponent)},
        PERM    Application-comments   {REQ #constraint-name {"31"},
                                               PERM #external-data {ANY_VALUE}}
}
```

7.3.3.38 TableComponent

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates  {$TableComponentGFS},
        REQApplication-comments        {REQ  #constraint-name {"32"},
                                               PERM #external-data {ANY_VALUE}},
        REQLayout-style                {STYLE_ID_OF(LStyleT6)}
SPECIFIC:
        REQObject-class                {OBJECT_CLASS_ID_OF(TableComponent)},
        REQSubordinates                {SUB_ID_OF(RowComponent)+},
        PERM    Application-comments   {REQ  #constraint-name {"32"},
                                               PERM #external-data   {ANY_VALUE}}
}
```

7.3.3.39 RowComponent

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates  {$RowComponentGFS},
        REQApplication-comments        {REQ  #constraint-name {"33"},
                                               PERM #external-data   {ANY_VALUE}},
        REQ   Layout-style                            {STYLE_ID_OF(LStyleT7)}
SPECIFIC:
        REQObject-class                {OBJECT_CLASS_ID_OF(RowComponent)},
        REQSubordinates                {SUB_ID_OF(EntryElement)+},
        PERM    Application-comments   {REQ  #constraint-name {"33"},
                                               PERM #external-data   {ANY_VALUE}}
}
```

7.3.3.40 Form

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates  {$FormGFS},
        REQApplication-comments        {REQ  #constraint-name {"34"},
                                               PERM #external-data   {ANY_VALUE}},
        REQLayout-style                {STYLE_ID_OF(LStyleT1)}
SPECIFIC:
        REQObject-class                {OBJECT_CLASS_ID_OF(Form)},
        REQSubordinates                {SUB_ID_OF(EntryElement)+,
                                               SUB_ID_OF(EntryGroup)+},
        PERM    Application-comments   {REQ  #constraint-name {"34"},
                                               PERM #external-data   {ANY_VALUE}}
}
```

7.3.3.41 EntryElement

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates   {$EntryElementGFS},
        REQApplication-comments         {REQ  #constraint-name {"35"},
                                                 PERM #external-data   {ANY_VALUE}},
        REQLayout-style                 {STYLE_ID_OF(LStyleT2)}
SPECIFIC:
        REQObject-class                 {OBJECT_CLASS_ID_OF(EntryElement)},
        REQSubordinates                 {SUB_ID_OF(EntryText),
                                                 SUB_ID_OF(EntryRaster),
                                                 SUB_ID_OF(EntryGeometric)},
        PERM    Application-comments    {REQ  #constraint-name {"35"},
                                                 PERM #external-data  {ANY_VALUE}}
}
```

7.3.3.42 EntryGroup

```
:COMP-LOGICAL {
GENERIC:
        REQGenerator-for-subordinates   {$EntryGroupGFS},
        REQApplication-comments         {REQ  #constraint-name {"36"},
                                                 PERM #external-data   {ANY_VALUE}}
SPECIFIC:
        REQObject-class                 {OBJECT_CLASS_ID_OF(EntryGroup)},
        REQSubordinates                 {SUB_ID_OF(EntryElement)+,
                                                 SUB_ID_OF(EntryGroup)+},
        PERM    Application-comments    {REQ  #constraint-name {"36"},
                                                 PERM #external-data   {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                   {STYLE_ID_OF(LStyleT3)}
}
```

7.3.3.43 EntryText

```
:BASIC-LOGICAL {
GENERIC:
        REQApplication-comments         {REQ  #constraint-name {"41"},
                                                 PERM #external-data       {ANY_VALUE}}
SPECIFIC:
        REQObject-class                 {OBJECT_CLASS_ID_OF(EntryText)},
        PERM    Application-comments    {REQ  #constraint-name {"41"},
                                                 PERM #external-data   {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM    Layout-style                   {STYLE_ID_OF(LStyleT9)},
        PERM    Presentation-style             {STYLE_ID_OF(PStyle1)},
        PERM    Content-architecture-class     {$FC|$PC|$FPC},
        PERM    Content-portions               {CONTENT_ID_OF(Character-content-portion)+}
        -- If the attribute "content portions" is specified neither in the specific nor in the generic part, then the
        attribute "resource" shall be specified. --
}
```

7.3.3.44 EntryRaster

```
:BASIC-LOGICAL {

GENERIC:
        REQContent-architecture-class   {$FPR},
        REQApplication-comments         {REQ  #constraint-name {"42"},
                                                 PERM #external-data   {ANY_VALUE}}
SPECIFIC:
        REQObject-class                 {OBJECT_CLASS_ID_OF(EntryRaster)},
        PERM    Content-architecture-class     {$FPR},
        PERM    Application-comments    {REQ  #constraint-name {"42"},
                                                 PERM #external-data   {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
        PERM     Layout-style                       {STYLE_ID_OF(LStyleT9)},
        PERM     Presentation-style                 {STYLE_ID_OF(PStyle3)},
        PERM     Content-portions                   {CONTENT_ID_OF(Raster-graphics-content-portion)}
        -- If the attribute "content portions" is specified neither in the specific nor in the generic part, then the
        attribute "resource" shall be specified. --
}
```

7.3.3.45 EntryGeometric

```
:BASIC-LOGICAL {
GENERIC:
            REQContent-architecture-class      {$FPG},
            REQApplication-comments            {REQ  #constraint-name {"43"},
                                                          PERM #external-data   {ANY_VALUE}}
SPECIFIC:
            REQObject-class                    {OBJECT_CLASS_ID_OF(EntryGeometric)},
            PERM     Content-architecture-class          {$FPG},
            PERM     Application-comments      {REQ  #constraint-name {"43"},
                                                          PERM #external-data   {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
            PERM     Layout-style                       {STYLE_ID_OF(LStyleT9)},
            PERM     Presentation-style                 {STYLE_ID_OF(PStyle2)},
            PERM     Content-portions                   {CONTENT_ID_OF(Geometric-graphics-content-portion)}
            -- If the attribute "content portions" is specified neither in the specific nor in the generic part, then the
            attribute "resource" shall be specified. --
}
```

7.4 Layout constituent constraints

7.4.1 Macro definitions

```
DEFINE(DocLayRootGFS,   "
 <construction-expr>       ::= <construction-term> | <construction-type>;

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

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

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

DEFINE(PageSetGFS,        "
 <construction-expr>       ::= <construction-term>|<construction-type>;

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

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

 <construction-factor>     ::= OBJECT_CLASS_ID_OF(Page)
                                    | OBJECT_CLASS_ID_OF(RectoPage)
                                    | OBJECT_CLASS_ID_OF(VersoPage)
                                    | <construction-type>;
")

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

 <headerarea>              ::= OBJECT_CLASS_ID_OF(BasicHeader)
                                    | OBJECT_CLASS_ID_OF(CompositeHeader);
 <bodyarea>                   ::= OBJECT_CLASS_ID_OF(VariableCompositeBody)
                                        | OBJECT_CLASS_ID_OF(FixedCompositeBody)
                                        | OBJECT_CLASS_ID_OF(BasicBody);

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

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

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

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

DEFINE(HeaderFooterGFS, "$CompositeCommonGFS")

DEFINE(FixedCompositeBodyGFS,    "
 <construction-expr>        ::= SEQ(<construction-term>...);

 <construction-term>        ::= <construction-factor1>
                                        | OPT <construction-factor1>
                                        | CHO({<construction-factor1>}...)
                                        | <construction-factor2>;

 <construction-factor1>    ::= OBJECT_CLASS_ID_OF(BasicFixture)
                                        | OBJECT_CLASS_ID_OF(ColumnFixed)
                                        | OBJECT_CLASS_ID_OF(CompositeFixtureFixed)
                                        | OBJECT_CLASS_ID_OF(VariableCompositeBody);

 <construction-factor2>    ::= OBJECT_CLASS_ID_OF(CompositeCommon)
                                        | OBJECT_CLASS_ID_OF(SourcedContentFixed)
                                        | OBJECT_CLASS_ID_OF(ArrangedContentFixed);
")

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

 <construction-term>        ::= <construction-factor1>
                                        | OPT <construction-factor1>
                                        | REP <construction-factor1>
                                        | OPT REP <construction-factor1>;

 <construction-type>        ::= SEQ({<construction-term>|<construction-factor2>}...)
                                        | CHO({<construction-term>}...);

 <construction-factor1>    ::= OBJECT_CLASS_ID_OF(BasicFloat)
                                        | OBJECT_CLASS_ID_OF(SnakingColumns)
                                        | OBJECT_CLASS_ID_OF(SynchronizedColumns)
                                        | OBJECT_CLASS_ID_OF(CompositeFloat)
                                        | OBJECT_CLASS_ID_OF(CompositeFixtureVariable)
                                        | OBJECT_CLASS_ID_OF(TableArea)
                                        | OBJECT_CLASS_ID_OF(FootnoteArea)
                                        | <construction-type>;

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

 <construction-factor2>    ::= OBJECT_CLASS_ID_OF(ArrangedContentVariable)
                                        | OBJECT_CLASS_ID_OF(SourcedContentVariable);
")
```

```
DEFINE(SnakingColumnsGFS,   "
 <construction-expr>        ::= REP <construction-factor1>
                                    | <construction-term>
                                    | SEQ(<construction-type>...);
 <construction-term>        ::= SEQ(<construction-type><following-term>);
 <following-term>    ::= OPT<construction-factor1>
                                    | <construction-factor2>
                                    | OPT<construction-term> ;

 <construction-type>        ::= <construction-factor1>|<construction-factor2>;
 <construction-factor1>     ::= OBJECT_CLASS_ID_OF(ColumnVariable)
                                    | OBJECT_CLASS_ID_OF(CompositeColumnVariable);
 <construction-factor2>     ::= OBJECT_CLASS_ID_OF(ArrangedContentVariable)
                                    | OBJECT_CLASS_ID_OF(SourcedContentVariable);
")
DEFINE(SynchronizedColumnsGFS,   "
 <construction-expr>        ::= SEQ({<construction-type>}...)|<construction-term>;

 <construction-term>        ::= SEQ(<construction-type><following-term>);
 <following-term>           ::= OPT<construction-factor1>
                                    | <construction-factor2>
                                    | OPT<construction-term> ;
 <construction-type>        ::= <construction-factor1>|<construction-factor2>;
 <construction-factor1>     ::= OBJECT_CLASS_ID_OF(ColumnFixed)
                                    | OBJECT_CLASS_ID_OF(CompositeColumnFixed);
 <construction-factor2>     ::= OBJECT_CLASS_ID_OF(ArrangedContentFixed)
                                    | OBJECT_CLASS_ID_OF(SourcedContentFixed);
")
DEFINE(CompositeFloatGFS,   "
 <construction-expr>        ::=  SEQ(<construction-term1>[<construction-term2>]...);

 <construction-term1>       ::= <construction-factor1>|<construction-factor2>;
 <construction-term2>       ::= <construction-term1>
                                    | OPT<construction-factor1>;
 <construction-factor1>     ::= OBJECT_CLASS_ID_OF(BasicColumn)
                                    | OBJECT_CLASS_ID_OF(CompositeFixtureVariable)
                                    | OBJECT_CLASS_ID_OF(TableArea);
 <construction-factor2>     ::= OBJECT_CLASS_ID_OF(ArrangedContentVariable)
                                    | OBJECT_CLASS_ID_OF(SourcedContentVariable);
")
DEFINE(CompositeColumnGFS,   "
 <construction-expr>        ::= <construction-term>
                                    | <construction-type>
                                    | SEQ(<construction-term> <construction-footnote>)
                                    | SEQ(<construction-type> <construction-footnote>);

 <construction-term>        ::= <construction-factor1>
                                    | OPT <construction-factor1>
                                    | REP <construction-factor1>
                                    | OPT REP <construction-factor1>;

 <construction-type>        ::= SEQ({<construction-term> | <construction-factor2>}...)
                                    | CHO({<construction-term>}...);

 <construction-factor1>     ::= OBJECT_CLASS_ID_OF(BasicFloat)
                                    | OBJECT_CLASS_ID_OF(TableArea)
                                    | OBJECT_CLASS_ID_OF(CompositeFloat)
                                    | OBJECT_CLASS_ID_OF(CompositeFixtureVariable)
                                    | OBJECT_CLASS_ID_OF(FootnoteArea)
                                    | <construction-type>;

 <construction-footnote>  ::= OBJECT_CLASS_ID_OF(FootnoteArea)
                                    | OPT OBJECT_CLASS_ID_OF(FootnoteArea);
 <construction-factor2>     ::= OBJECT_CLASS_ID_OF(ArrangedContentVariable)
                                    | OBJECT_CLASS_ID_OF(SourcedContentVariable);
")
```

```
DEFINE(CompositeColumnVariableGFS,  "$CompositeColumnGFS")

DEFINE(CompositeColumnFixedGFS,  "$CompositeColumnGFS")

DEFINE(CompositeFixtureGFS,   "
 <construction-expr>       ::= <construction-factor>
                                | REP CHO(<construction-factor>...);

 <construction-factor>     ::= OBJECT_CLASS_ID_OF(BasicFloat)
                                | OBJECT_CLASS_ID_OF(FootnoteArea)
                                | OBJECT_CLASS_ID_OF(CompositeArtwork)
                                | OBJECT_CLASS_ID_OF(FormArea);
")

DEFINE(CompositeFixtureFixedGFS,  "$CompositeFixtureGFS")

DEFINE(CompositeFixtureVariableGFS,  "$CompositeFixtureGFS")

DEFINE(CompositeArtworkGFS,  "
 <construction-expr>       ::= REP OBJECT_CLASS_ID_OF(BasicFixture);
")

DEFINE(TableAreaGFS,   "
 <construction-expr>       ::= <row-area>
                  | SEQ([<table-header>] [<table-label>] <row-area>[<table-label>]);
 <table-header>            ::= OBJECT_CLASS_ID_OF(TableHeader);
 <table-label>             ::= OBJECT_CLASS_ID_OF(TableLabel);
 <row-area>                ::= REP OBJECT_CLASS_ID_OF(RowArea)
                                | REP CHO(OBJECT_CLASS_ID_OF(RowArea)...);
")

DEFINE(RowAreaGFS,   "
 <construction-expr>       ::= SEQ(OBJECT_CLASS_ID_OF(Cell)...)
                                | SEQ(OBJECT_CLASS_ID_OF(Cell)
                                        OBJECT_CLASS_ID_OF(SubRowGroup));
")

DEFINE(SubRowGroupGFS,   "
 <construction-expr>       ::= REP OBJECT_CLASS_ID_OF(SubRow);
")

DEFINE(SubRowGFS,   "
 <construction-expr>       ::= SEQ(OBJECT_CLASS_ID_OF(Cell)...);
")

DEFINE(TableHeaderGFS,   "
 <construction-expr>       ::= SEQ(OBJECT_CLASS_ID_OF(SourcedContentFixed)...);
")

DEFINE(TableLabelGFS,   "
 <construction-expr>       ::= SEQ(OBJECT_CLASS_ID_OF(TableLabelContent)...)
                                | SEQ(OBJECT_CLASS_ID_OF(TableLabelContent)
                                        OBJECT_CLASS_ID_OF(CompositeTableLabel));
")

DEFINE(CompositeTableLabelGFS,   "
 <construction-expr>       ::= SEQ(OBJECT_CLASS_ID_OF(LabelComponent)...);
")

DEFINE(LabelComponentGFS,   "
 <construction-expr>       ::= SEQ(OBJECT_CLASS_ID_OF(TableLabelContent)...);
")

DEFINE(FormAreaGFS,   "
 <construction-expr>       ::= AGG(<factor>...);
 <factor>                  ::= OBJECT_CLASS_ID_OF(ArrangedContentFixed)
                                | OBJECT_CLASS_ID_OF(FormEntryArea)
                                | OBJECT_CLASS_ID_OF(EntryGroupArea);
")

DEFINE(EntryGroupAreaGFS,"$FormAreaGFS")
```

7.4.2 Factor constraints

7.4.2.1 FACTOR ANY-LAYOUT

```
{
GENERIC:
        REQObject-type                  {VIRTUAL},
        REQObject-class-identifier      {ANY_VALUE}
SPECIFIC:
        PERM    Object-type                     {VIRTUAL},
        REQObject-identifier            {ANY_VALUE},
        CASE $DAC OF {
            $FDA:  PERM        Object-class     {VIRTUAL}
            $FPDA: REQ         Object-class     {VIRTUAL}
            },
        REQSubordinates                 {VIRTUAL}
SPECIFIC_AND_GENERIC:
        PERM    User-readable-comments  {ANY_STRING},
        PERM    User-visible-name               {ANY_STRING}
}
```

7.4.2.2 FACTOR ANY-PAGE

```
:ANY-LAYOUT {
GENERIC:
        REQ Object-type                         {'page'},

    CASE $DAC OF {
            $PDA-FPDA:
                REQ    Generator-for-subordinates       {$PageGFS},
                PERM Bindings           {PMUL {$SPECIFYBINDINGS},
                                                PERM {$INITIALISEPGNUMBER,$USEPGNUMBERS}}
            }
SPECIFIC:
        PERM    Object-type                     {'page'},
        REQSubordinates                 {SUB_ID_OF(BasicHeader),
                                                SUB_ID_OF(CompositeHeader),
                                                SUB_ID_OF(BasicBody),
                                                SUB_ID_OF(FixedCompositeBody),
                                                SUB_ID_OF(VariableCompositeBody),
                                                SUB_ID_OF(BasicFooter),
                                                SUB_ID_OF(CompositeFooter)}
SPECIFIC_AND_GENERIC:
        PERM    Dimensions                      {$PermissiblePageDimensions},
        PERM    Transparency            {ANY_VALUE},
        PERM    Colour                          {ANY_VALUE},
        PERM    Page-position           {ANY_VALUE}
}
```

7.4.2.3 FACTOR ANY-FRAME-FIXED

```
:ANY-LAYOUT {
GENERIC:
        REQObject-type                  {'frame'}
SPECIFIC:
        PERM    Object-type                     {'frame'}
SPECIFIC_AND_GENERIC:
        PERM    Position                                {REQ #fixed-position
                                                 {REQ #horizontal-position          {ANY_VALUE},
                                                  REQ #vertical-position            {ANY_VALUE}}},
        PERM    Dimensions                      {REQ #horizontal-dimension
                                                 {REQ #fixed-dimension              {ANY_VALUE}},
                                                REQ #vertical-dimension
                                                 {REQ #fixed-dimension              {ANY_VALUE}}},
        PERM    Transparency            {ANY_VALUE},
        PERM    Colour                          {ANY_VALUE},
        PERM    Border                          {ANY_VALUE}
}
```

7.4.2.4 FACTOR ANY-FRAME-VARIABLE

```
:ANY-LAYOUT {
GENERIC:
            REQObject-type                      {'frame'}
SPECIFIC:
            PERM      Object-type                          {'frame'},
            CASE $DAC OF {
                  $FPDA:
                        REQ Position                {REQ #fixed-position
                                                                {REQ #horizontal-position              {ANY_VALUE},
                                                                REQ #vertical-position                 {ANY_VALUE}}},
                        REQ Dimensions                     {REQ #horizontal-dimension
                                                                {REQ #fixed-dimension      {ANY_VALUE}},
                                                           REQ #vertical-dimension
                                                                {REQ #fixed-dimension      {ANY_VALUE}}}
                  }
SPECIFIC_AND_GENERIC:
            CASE $DAC OF {
                  $FDA:
                        PERM Position               {REQ #fixed-position
                                                                {REQ #horizontal-position              {ANY_VALUE},
                                                                REQ #vertical-position                 {ANY_VALUE}}},
                        PERM Dimensions                    {REQ #horizontal-dimension
                                                                {REQ #fixed-dimension      {ANY_VALUE}},
                                                           REQ #vertical-dimension
                                                                {REQ #fixed-dimension      {ANY_VALUE}}}
                  },
            PERM      Transparency              {ANY_VALUE},
            PERM      Colour                               {ANY_VALUE},
            PERM      Border                               {ANY_VALUE}
}
```

7.4.3 Constituent constraints

7.4.3.1 DocumentLayoutRoot

```
:ANY-LAYOUT {
GENERIC:
            REQObject-type                                 {'document-layout-root'},
            CASE $DAC OF {
                  $PDA-FPDA:
                        REQ    Generator-for-subordinates          {$DocLayRootGFS},
                        PERM Bindings               {PMUL {$SPECIFYBINDINGS},
                                                            PERM {$INITIALISEPGNUMBER}}
                  },
            REQApplication-comments             {REQ #constraint-name {"0"},
                                                            PERM #external-data {ANY_VALUE}}
SPECIFIC:
            PERM      Object-type                          {'document-layout-root'},
            CASE $DAC OF {
                  $FDA:
                        PERM    Object-class        {OBJECT_CLASS_ID_OF
                                                                (DocumentLayoutRoot)}
                  $FPDA:
                        REQ       Object-class      {OBJECT_CLASS_ID_OF
                                                                (DocumentLayoutRoot)}
                  },
            REQSubordinates                     {SUB_ID_OF(PageSet)+},
            PERM      Application-comments      {REQ #constraint-name {"0"},
                                                            PERM #external-data {ANY_VALUE}}
}
```

7.4.3.2 PageSet

```
:ANY-LAYOUT {
GENERIC:
            REQObject-type                      {'page-set'},
        CASE $DAC OF {
                $PDA-FPDA:
                        REQ    Generator-for-subordinates              {$PageSetGFS},
                        PERM Bindings               {PMUL {$SPECIFYBINDINGS},
                                                            PERM {$INITIALISEPGNUMBER}}
                },
            REQApplication-comments            {REQ  #constraint-name {"1"},
                                                            PERM #external-data {ANY_VALUE}}
SPECIFIC:
            PERM      Object-type                             {'page-set'},
            CASE $DAC OF {
                $FDA:
                        PERM Object-class          {OBJECT_CLASS_ID_OF(PageSet)}
                $FPDA:
                        REQ    Object-class          {OBJECT_CLASS_ID_OF(PageSet)}
                },
            REQSubordinates                         {SUB_ID_OF(Page)+,
                                                            SUB_ID_OF(RectoPage)+,
                                                            SUB_ID_OF(VersoPage)+},
            PERM      Application-comments     {REQ  #constraint-name {"1"},
                                                            PERM #external-data {ANY_VALUE}}
}
```

7.4.3.3 Page

```
:ANY-PAGE {
GENERIC:
            REQApplication-comments            {REQ #constraint-name {"2"},
                                                            PERM #external-data {ANY_VALUE}}
SPECIFIC:
            CASE $DAC OF {
                $FDA:
                        PERM Object-class                       {OBJECT_CLASS_ID_OF(Page)}
                $FPDA:
                        REQ   Object-class                       {OBJECT_CLASS_ID_OF(Page)}
                },
            PERM      Application-comments     {REQ #constraint-name {"2"},
                                                            PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
            PERM      Medium-type                  {PERM #nominal-page-size {$NominalPageSizes},
                                                            PERM #side-of-sheet {ANY_VALUE}}
}
```

7.4.3.4 RectoPage

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

7.4.3.5 VersoPage

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

7.4.3.6 CompositeHeader

```
:ANY-FRAME-FIXED {
GENERIC:
        CASE $DAC OF {
            $PDA-FPDA:
                REQ   Generator-for-subordinates        {$HeaderFooterGFS}},
        PERM    Layout-path                     {'270-degrees'-- H/F layouts A1,B2 --
                                                |'180-degrees' -- H/F layout B1 --
                                                |'0-degrees' -- H/F layout A2 --},
        REQApplication-comments         {REQ  #constraint-name {"5"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        CASE $DAC OF {
            $FDA:
                PERM Object-class               {OBJECT_CLASS_ID_OF(CompositeHeader)}
            $FPDA:
                REQ  Object-class               {OBJECT_CLASS_ID_OF(CompositeHeader)}
            },
        REQSubordinates                 {SUB_ID_OF(SourcedContentFixed)+,
                                                SUB_ID_OF(ArrangedContentFixed)+,
                                                SUB_ID_OF(SourcedContentVariable)+,
                                                SUB_ID_OF(ArrangedContentVariable)+},
        PERM    Imaging-order           {SUB_ID_OF(SourcedContentFixed)+,
                                                SUB_ID_OF(ArrangedContentFixed)+},
        PERM    Application-comments    {REQ  #constraint-name {"5"},
                                                PERM #external-data {ANY_VALUE}}
}
```

7.4.3.7 CompositeFooter

```
:ANY-FRAME-FIXED {
GENERIC:
        CASE $DAC OF {
            $PDA-FPDA:
                REQ   Generator-for-subordinates        {$HeaderFooterGFS}},
        PERM    Layout-path                     {'270-degrees' -- H/F layouts A1,B2 --
                                                |'180-degrees' -- H/F layout B1 --
                                                |'0-degrees' -- H/F layout A2 --},
        REQApplication-comments         {REQ  #constraint-name {"32"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        CASE $DAC OF {
            $FDA:
                PERM Object-class       {OBJECT_CLASS_ID_OF(CompositeFooter)}
            $FPDA:
                REQ   Object-class      {OBJECT_CLASS_ID_OF(CompositeFooter)}
            },
        REQSubordinates                 {SUB_ID_OF(SourcedContentFixed)+,
                                                SUB_ID_OF(ArrangedContentFixed)+,
                                                SUB_ID_OF(SourcedContentVariable)+,
                                                SUB_ID_OF(ArrangedContentVariable)+},
        PERM    Imaging-order           {SUB_ID_OF(SourcedContentFixed)+,
                                                SUB_ID_OF(ArrangedContentFixed)+},
        PERM    Application-comments    {REQ  #constraint-name {"32"},
                                                PERM #external-data {ANY_VALUE}}
}
```

7.4.3.8 FixedCompositeBody

```
:ANY-FRAME-FIXED {
GENERIC:
        CASE $DAC OF {
            $PDA-FPDA:
                REQ    Generator-for-subordinates          {$FixedCompositeBodyGFS},
                PERM Layout-path        {'270-degrees' -- body layout A --
                                                |'0-degrees'   -- body layout B --
                                                |'180-degrees' -- body layout C --}
            },
        REQApplication-comments         {REQ #constraint-name {"6"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        CASE $DAC OF {
            $FDA:
                PERM Object-class       {OBJECT_CLASS_ID_OF
                                                            (FixedCompositeBody)}
            $FPDA:
                REQ    Object-class     {OBJECT_CLASS_ID_OF
                                                            (FixedCompositeBody)}
            },
        REQSubordinates                 {SUB_ID_OF(CompositeCommon)+,
                                                SUB_ID_OF(BasicFixture)+,
                                                SUB_ID_OF(ColumnFixed)+,
                                                SUB_ID_OF(CompositeFixtureFixed)+,
                                                SUB_ID_OF(VariableCompositeBody)+,
                                                SUB_ID_OF(SourcedContentFixed)+,
                                                SUB_ID_OF(ArrangedContentFixed)+},
        PERM    Imaging-order           {SUB_ID_OF(CompositeCommon)+,
                                                SUB_ID_OF(BasicFixture)+,
                                                SUB_ID_OF(ColumnFixed)+,
                                                SUB_ID_OF(CompositeFixtureFixed)+,
                                                SUB_ID_OF(VariableCompositeBody)+,
                                                SUB_ID_OF(SourcedContentFixed)+,
                                                SUB_ID_OF(ArrangedContentFixed)+},
        PERM    Application-comments    {REQ  #constraint-name {"6"},
                                                PERM #external-data {ANY_VALUE}}
}
```

7.4.3.9 VariableCompositeBody

```
:ANY-FRAME-FIXED {
GENERIC:
        CASE $DAC OF {
            $PDA-FPDA:
                REQ    Generator-for-subordinates          {$VariableCompositeBodyGFS},
                PERM Layout-path        {'270-degrees' -- body layout A --
                                                |'0-degrees'   -- body layout B --
                                                |'180-degrees' -- body layout C --}
            },
        REQApplication-comments         {REQ  #constraint-name {"7"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
        CASE $DAC OF {
            $FDA:
                PERM Object-class         {OBJECT_CLASS_ID_OF
                                                              (VariableCompositeBody)}
            $FPDA:
                REQ    Object-class       {OBJECT_CLASS_ID_OF
                                                              (VariableCompositeBody)}
            },
        REQSubordinates                   {SUB_ID_OF(SnakingColumns)+,
                                                  SUB_ID_OF(SynchronizedColumns)+,
                                                  SUB_ID_OF(BasicFloat)+,
                                                  SUB_ID_OF(FootnoteArea)+,
                                                  SUB_ID_OF(CompositeFloat)+,
                                                  SUB_ID_OF(CompositeFixtureVariable)+,
                                                  SUB_ID_OF(TableArea)+,
                                                  SUB_ID_OF(ArrangedContentVariable)+,
                                                  SUB_ID_OF(SourcedContentVariable)+},
        PERM      Application-comments    {REQ  #constraint-name {"7"},
                                                  PERM #external-data {ANY_VALUE}}
}
```

7.4.3.10 ColumnFixed

```
:ANY-FRAME-VARIABLE {
GENERIC:
        CASE $DAC OF {
            $PDA-FPDA:
                    REQ                 Position            {REQ #fixed-position
                                                            {REQ #horizontal-position {ANY_VALUE},
                                                             REQ #vertical-position {ANY_VALUE}}},
                    CASE SUPERIOR ({VariableCompositeBody
                                | FixedCompositeBody} (Layout-path)) OF {

                {'270-degrees'}: -- body layout A --
                      REQ   Dimensions          {REQ #horizontal-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}
                                                        |REQ #maximum-size {'applies'}},
                                                       REQ #vertical-dimension
                                                        {REQ #rule-b {ANY_VALUE}
                                                        |REQ #maximum-size {'applies'}}},
                      PERM Layout-path                  {'270-degrees'}

                {'0-degrees'}: -- body layout B --
                    REQ   Dimensions            {REQ #horizontal-dimension
                                                        {REQ #rule-b {ANY_VALUE}
                                                        |REQ #maximum-size {'applies'}},
                                                       REQ #vertical-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}
                                                        |REQ #maximum-size {'applies'}}},
                      REQ  Layout-path                  {'0-degrees'}

                {'180-degrees'}: -- body layout C -
                      REQ   Dimensions          {REQ #horizontal-dimension
                                                        {REQ #maximum-size {'applies'}},
                                                       REQ #vertical-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}
                                                        |REQ #maximum-size {'applies'}}},
                      REQ   Layout-path                 {'180-degrees'}
            } },
        REQApplication-comments           {REQ #constraint-name {"8"},
                                                  PERM #external-data {ANY_VALUE}}
SPECIFIC:
        CASE $DAC OF {
            $FDA:
                    PERM Object-class  {OBJECT_CLASS_ID_OF(ColumnFixed)}
            $FPDA:
                    REQ   Object-class   {OBJECT_CLASS_ID_OF(ColumnFixed)}
            },
          REQ       Subordinates              {SUB_ID_OF(SpecificBlock)+},
          PERM      Imaging-order             {SUB_ID_OF(SpecificBlock)+},
          PERM      Application-comments      {REQ  #constraint-name {"8"},
                                                        PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
          PERM      Permitted-categories               {ANY_STRING...}
}
```

7.4.3.11 ColumnVariable

```
:ANY-FRAME-VARIABLE {
GENERIC:
          CASE $DAC OF {
               $PDA-FPDA:
                    REQ    Position            {REQ #variable-position {
                                                                PERM #offset        {ANY_VALUE},
                                                                PERM #separation {ANY_VALUE},
                                                                PERM #alignment  {ANY_VALUE},
                                                                PERM #fill-order {'normal-order'}}},
                    CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {

               {'270-degrees'}: -- body layout A --
                         REQ  Dimensions              {REQ #horizontal-dimension
                                                                 {REQ #fixed-dimension {ANY_VALUE}},
                                                                REQ #vertical-dimension
                                                                 {REQ #rule-b {ANY_VALUE}
                                                                 |REQ #maximum-size {'applies'}}},
                         PERM Layout-path             {'270-degrees'}

               {'0-degrees'}: -- body layout B --
                         REQ  Dimensions              {REQ #horizontal-dimension
                                                                 {REQ #rule-b {ANY_VALUE}
                                                                 |REQ #maximum-size {'applies'}},
                                                                REQ #vertical-dimension
                                                                 {REQ #fixed-dimension {ANY_VALUE}}},
                         REQ  Layout-path             {'0-degrees'}

               {'180-degrees'}: -- body layout C --
                         REQ  Dimensions              {REQ #horizontal-dimension
                                                                 {REQ #rule-b {ANY_VALUE}
                                                                 |REQ #maximum-size {'applies'}},
                                                                REQ #vertical-dimension
                                                                 {REQ #fixed-dimension {ANY_VALUE}}},
                         REQ  Layout-path             {'180-degrees'}
               }    },
          REQ       Application-comments               {REQ  #constraint-name {"9"},
                                                        PERM #external-data {ANY_VALUE}}
SPECIFIC:
          CASE $DAC OF {
               $FDA:
                    PERM  Object-class                 {OBJECT_CLASS_ID_OF(ColumnVariable)}
               $FPDA:
                    REQ   Object-class                 {OBJECT_CLASS_ID_OF(ColumnVariable)}
               },
          REQSubordinates                     {SUB_ID_OF(SpecificBlock)+},
          PERM      Imaging-order             {SUB_ID_OF(SpecificBlock)+},
          PERM      Application-comments      {REQ  #constraint-name {"9"},
                                                        PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
          PERM  Permitted-categories                   {ANY_STRING...}
}
```

7.4.3.12 SnakingColumns

```
:ANY-FRAME-VARIABLE {
GENERIC:
        CASE $DAC OF {
            $PDA-FPDA:
                REQ   Generator-for-subordinates          {$SnakingColumnsGFS},
                REQ   Position            {REQ #variable-position {
                                                          PERM #offset    {ANY_VALUE},
                                                          PERM #separation {ANY_VALUE},
                                                          PERM #alignment  {ANY_VALUE},
                                                          PERM #fill-order {'normal-order'}}},
                PERM Balance              {ANY_VALUE},

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

            {'270-degrees'}: -- body layout A --
                    REQ                 Dimensions                {REQ #horizontal-dimension
                                                           {REQ #fixed-dimension {ANY_VALUE}
                                                           |REQ #maximum-size {'applies'}},
                                                          REQ #vertical-dimension
                                                           {REQ #rule-b {ANY_VALUE}}},
                    REQ                 Layout-path       {'0-degrees'|'180-degrees'}

            {'0-degrees'}: -- body layout B --
                    REQ                 Dimensions                {REQ #horizontal-dimension
                                                           {REQ #rule-b {ANY_VALUE}},
                                                          REQ #vertical-dimension
                                                           {REQ #fixed-dimension {ANY_VALUE}
                                                           |REQ #maximum-size {'applies'}}},
                    PERM Layout-path            {'90-degrees'|'270-degrees'}

            {'180-degrees'}: -- body layout C --
                    REQ                 Dimensions                {REQ #horizontal-dimension
                                                           {REQ #rule-b {ANY_VALUE}},
                                                          REQ #vertical-dimension
                                                           {REQ #fixed-dimension {ANY_VALUE}
                                                           |REQ #maximum-size {'applies'}}},
                    PERM Layout-path            {'270-degrees'}

        }  },
        REQApplication-comments           {REQ  #constraint-name {"10"},
                                                    PERM #external-data {ANY_VALUE}}
SPECIFIC:
        CASE $DAC OF {
            $FDA:
                    PERM  Object-class {OBJECT_CLASS_ID_OF(Snakingcolumns)}
            $FPDA:
                    REQ   Object-class  {OBJECT_CLASS_ID_OF(Snakingcolumns)}
            },
        REQSubordinates                   {SUB_ID_OF(ColumnVariable)+,
                                                  SUB_ID_OF(CompositeColumnVariable)+,
                                                  SUB_ID_OF(ArrangedContentVariable)+,
                                                  SUB_ID_OF(SourcedContentVariable)+},
        PERM      Application-comments    {REQ  #constraint-name {"10"},
                                                    PERM #external-data {ANY_VALUE}}
}
```

7.4.3.13 SynchronizedColumns

```
:ANY-FRAME-VARIABLE {
GENERIC:
        CASE $DAC OF {
            $PDA-FPDA:
                REQ   Generator-for-subordinates          {$SynchronizedColumnsGFS},
                REQ   Position            {REQ #variable-position {
                                                          PERM #offset    {ANY_VALUE},
                                                          PERM #separation {ANY_VALUE},
                                                          PERM #alignment  {ANY_VALUE},
                                                          PERM #fill-order {'normal-order'}}},
                CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {

                {'270-degrees'}: -- body layout A --
                        REQ  Dimensions              {REQ #horizontal-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}
                                                            |REQ #maximum-size {'applies'}},
                                                           REQ #vertical-dimension
                                                            {REQ #rule-b {ANY_VALUE}}},
                        PERM Layout-path             {'270-degrees'}

                {'0-degrees'}: -- body layout B --
                        REQ  Dimensions              {REQ #horizontal-dimension
                                                            {REQ #rule-b {ANY_VALUE}},
                                                           REQ #vertical-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}
                                                            |REQ #maximum-size {'applies'}}},
                        REQ  Layout-path             {'0-degrees'}

                {'180-degrees'}: -- body layout C --
                        REQ  Dimensions              {REQ #horizontal-dimension
                                                            {REQ #rule-b{ANY_VALUE}},
                                                           REQ #vertical-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}
                                                            |REQ #maximum-size {'applies'}}},
                        REQ  Layout-path             {'180-degrees'}
                }   },
            REQApplication-comments          {REQ  #constraint-name {"11"},
                                                     PERM #external-data {ANY_VALUE}}
SPECIFIC:
            CASE $DAC OF {
                $FDA:
                        PERM Object-class  OBJECT_CLASS_ID_OF(SynchronizedColumns)}
                $FPDA:
                        REQ  Object-class  {OBJECT_CLASS_ID_OF(SynchronizedColumns)}
                },
            REQSubordinates                  {SUB_ID_OF(ColumnFixed)+,
                                                     SUB_ID_OF(CompositeColumnFixed)+,
                                                     SUB_ID_OF(ArrangedContentFixed)+,
                                                     SUB_ID_OF(SourcedContentFixed)+},
            PERM     Application-comments    {REQ #constraint-name {"11"},
                                                     PERM #external-data {ANY_VALUE}}
}
```

7.4.3.14 BasicFloat

```
:ANY-FRAME-VARIABLE {
GENERIC:
            CASE $DAC OF {
                $PDA-FPDA:
                    REQ    Position          {REQ #variable-position {
                                                           PERM #offset     {ANY_VALUE},
                                                           PERM #separation {ANY_VALUE},
                                                           PERM #alignment  {ANY_VALUE},
                                                           PERM #fill-order {'normal-order'}}},

                CASE SUPERIOR ({VariableCompositeBody
                            | CompositeColumnVariable  | CompositeColumnFixed
                            | CompositeFixtureVariable | CompositeFixtureFixed} (Layout-path)) OF {

                {'270-degrees'}: -- body layout A --
                        REQ  Dimensions              {REQ #horizontal-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}
                                                            |REQ #maximum-size {'applies'}},
                                                           REQ #vertical-dimension
                                                            {REQ #rule-b {ANY_VALUE}}},
                        PERM Layout-path             {'270-degrees'}
```

```
          {’0-degrees’}: -- body layout B --
                    REQ  Dimensions           {REQ #horizontal-dimension
                                                        {REQ #rule-b {ANY_VALUE}},
                                                       REQ #vertical-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}
                                                        |REQ #maximum-size {’applies’}}},
                    REQ  Layout-path          {’0-degrees’}

          {’180-degrees’}: -- body layout C --
                    REQ  Dimensions           {REQ #horizontal-dimension
                                                        {REQ #rule-b {ANY_VALUE}},
                                                       REQ #vertical-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}
                                                        |REQ #maximum-size {’applies’}}},
                    REQ  Layout-path          {’180-degrees’}
          }    },
     REQApplication-comments                  {REQ  #constraint-name {"12"},
                                                        PERM #external-data {ANY_VALUE}}
SPECIFIC:
     CASE $DAC OF {
          $FDA:
               PERM Object-class      {OBJECT_CLASS_ID_OF(BasicFloat)}
          $FPDA:
               REQ    Object-class    {OBJECT_CLASS_ID_OF(BasicFloat)}
          },
     REQSubordinates                  {SUB_ID_OF(SpecificBlock)+},
     PERM      Imaging-order          {SUB_ID_OF(SpecificBlock)+},
     PERM      Application-comments   {REQ  #constraint-name {"12"},
                                                PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
     PERM      Permitted-categories           {ANY_STRING...}
}
```

7.4.3.15 CompositeFloat

```
:ANY-FRAME-VARIABLE {
GENERIC:
     CASE $DAC OF {
          $PDA-FPDA:
               REQ    Generator-for-subordinates          {$CompositeFloatGFS},
               REQ    Position            {REQ #variable-position {
                                                        PERM #offset        {ANY_VALUE},
                                                        PERM #separation {ANY_VALUE},
                                                        PERM #alignment  {ANY_VALUE},
                                                        PERM #fill-order {’normal-order’}}},

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

          {’270-degrees’}: -- body layout A --
                    REQ  Dimensions           {REQ #horizontal-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}
                                                        |REQ #maximum-size {’applies’}},
                                                       REQ #vertical-dimension
                                                        {REQ #rule-a {ANY_VALUE}}},
                    PERM Layout-path          {’0-degrees’|’180-degrees’}

          {’0-degrees’}: -- body layout B --
                    REQ  Dimensions           {REQ #horizontal-dimension
                                                        {REQ #rule-a {ANY_VALUE}},
                                                       REQ #vertical-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}
                                                        |REQ #maximum-size {’applies’}}},
                    REQ  Layout-path          {’90-degrees’|’270-degrees’}
```

```
        {'180-degrees'}: -- body layout C --
                REQ  Dimensions             {REQ #horizontal-dimension
                                                    {REQ #rule-a {ANY_VALUE}},
                                                   REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}
                                                    |REQ #maximum-size {'applies'}}},
                REQ  Layout-path            {'90-degrees'|'270-degrees'}
        }  },
    REQApplication-comments          {REQ  #constraint-name {"13"},
                                            PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
            PERM Object-class        {OBJECT_CLASS_ID_OF(CompositeFloat)}
        $FPDA:
            REQ    Object-class      {OBJECT_CLASS_ID_OF(CompositeFloat)}
        },
    REQSubordinates                  {SUB_ID_OF(CompositeFixtureVariable)+,
                                            SUB_ID_OF(TableArea)+,
                                            SUB_ID_OF(BasicColumn)+,
                                            SUB_ID_OF(ArrangedContentVariable)+,
                                            SUB_ID_OF(SourcedContentVariable)+},
    PERM      Application-comments          {REQ  #constraint-name {"13"},
                                             PERM #external-data {ANY_VALUE}}
}
```

7.4.3.16 BasicColumn

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
            REQ   Position           {REQ #variable-position {
                                                  PERM #offset     {ANY_VALUE},
                                                  PERM #separation {ANY_VALUE},
                                                  PERM #alignment  {ANY_VALUE},
                                                  PERM #fill-order {'normal-order'}}},

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

        {'270-degrees'}: -- body layout A --
                REQ  Dimensions             {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}
                                                    |REQ #rule-b {ANY_VALUE}},
                                                   REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}
                                                    |REQ #rule-b {ANY_VALUE}
                                                    |REQ #maximum-size {'applies'}}},
                PERM Layout-path            {'270-degrees'}

        {'0-degrees'}: -- body layout B --
                REQ  Dimensions             {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}
                                                    |REQ #rule-b {ANY_VALUE}
                                                    |REQ #maximum-size{'applies'}},
                                                   REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}
                                                    |REQ #rule-b {ANY_VALUE}}},
                REQ  Layout-path            {'0-degrees'}

        {'180-degrees'}: -- body layout C --
                REQ  Dimensions             {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}
                                                    |REQ #rule-b {ANY_VALUE}
                                                    |REQ #maximum-size{'applies'}},
                                                   REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}
                                                    |REQ #rule-b {ANY_VALUE}}},
                REQ  Layout-path            {'180-degrees'}
        }  },
          REQ   Application-comments                    {REQ  #constraint-name {"14"},
                                                          PERM #external-data {ANY_VALUE}}
SPECIFIC:
          CASE $DAC OF {
               $FDA:
                    PERM  Object-class                  {OBJECT_CLASS_ID_OF(BasicColumn)}
               $FPDA:
                    REQ   Object-class                  {OBJECT_CLASS_ID_OF(BasicColumn)}
               },
          REQSubordinates                     {SUB_ID_OF(SpecificBlock)+},
          PERM      Application-comments      {REQ  #constraint-name {"14"},
                                                          PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
          PERM   Permitted-categories                   {ANY_STRING...}
}
```

7.4.3.17 FootnoteArea

```
:ANY-FRAME-VARIABLE {
GENERIC:
          CASE $DAC OF {
               $PDA-FPDA:
                    RQ   Position             {REQ #variable-position {
                                                        PERM #offset     {ANY_VALUE},
                                                        PERM #separation {ANY_VALUE},
                                                        PERM #alignment  {ANY_VALUE},
                                                        PERM #fill-order {'reverse-order'}}},

               CASE SUPERIOR ({VariableCompositeBody
                           | CompositeColumnVariable  | CompositeColumnFixed
                           | CompositeFixtureVariable | CompositeFixtureFixed} (Layout-path)) OF {

               {'270-degrees'}: -- body layout A --
                         REQ   Dimensions             {REQ #horizontal-dimension
                                                                {REQ #fixed-dimension {ANY_VALUE}
                                                                |REQ #maximum-size {'applies'}},
                                                               REQ #vertical-dimension
                                                                {REQ #rule-b {ANY_VALUE}}},
                           PERM Layout-path                     {'270-degrees'}

               {'0-degrees'}: -- body layout B --
                         REQ   Dimensions             {REQ #horizontal-dimension
                                                                {REQ #rule-b {ANY_VALUE}},
                                                               REQ #vertical-dimension
                                                                {REQ #fixed-dimension {ANY_VALUE}
                                                                |REQ #maximum-size {'applies'}}},
                         REQ   Layout-path            {'0-degrees'}

               {'180-degrees'}: -- body layout C --
                         REQ   Dimensions             {REQ #horizontal-dimension
                                                                {REQ #rule-b {ANY_VALUE}},
                                                               REQ #vertical-dimension
                                                                {REQ #fixed-dimension {ANY_VALUE}
                                                                |REQ #maximum-size {'applies'}}},
                         REQ   Layout-path            {'180-degrees'}
          }  },
          REQPermitted-categories  {$FOOTNOTECATEGORY},
               -- For example,
                    For CompositeBody                 "Footnote-1"
                    For SnakingColumns                "Footnote-2"
                    For SynchronizedColumns           "Footnote-3", "Footnote-4",
                                                      "Footnote-5"
                    For CompositeFixture              "Footnote-6"
                    --
          REQ   Application-comments                    {REQ  #constraint-name {"15"},
                                                          PERM #external-data {ANY_VALUE}}
SPECIFIC:
        CASE $DAC OF {
            $FDA:
                PERM  Object-class              {OBJECT_CLASS_ID_OF(FootnoteArea)}
            $FPDA:
                REQ   Object-class              {OBJECT_CLASS_ID_OF(FootnoteArea)}
            },
        REQSubordinates                 {SUB_ID_OF(SpecificBlock)+},
        PERM      Permitted-categories          {$FOOTNOTECATEGORY},
        PERM      Application-comments  {REQ  #constraint-name {"15"},
                                                 PERM #external-data {ANY_VALUE}}
}
```

7.4.3.18 ArrangedContentFixed

```
:ANY-FRAME-FIXED {
GENERIC:
        CASE $DAC OF {
            $PDA-FPDA:
                REQ    Generator-for-subordinates
                                                {<construction-expr>::=
SEQ(OBJECT_CLASS_ID_OF(GenericBlock)...);}},
        REQ            Application-comments     {REQ  #constraint-name {"16"},
                                                       PERM #external-data {ANY_VALUE}}
SPECIFIC:
        CASE $DAC OF {
            $FDA:
                PERM  Object-class              {OBJECT_CLASS_ID_OF
                                                       (ArrangedContentFixed)}
            $FPDA:
                REQ   Object-class              {OBJECT_CLASS_ID_OF
                                                       (ArrangedContentFixed)}
            },
        REQSubordinates                 {SUB_ID_OF(GenericBlock)+},
        PERM      Imaging-order         {SUB_ID_OF(GenericBlock)+},
        PERM      Application-comments  {REQ  #constraint-name {"16"},
                                                 PERM #external-data {ANY_VALUE}}
}
```

7.4.3.19 ArrangedContentVariable

```
:ANY-FRAME-VARIABLE {
GENERIC:
        CASE $DAC OF {
            $PDA-FPDA:
                REQ    Generator-for-subordinates {<construction-expr>::=
                                                       SEQ(OBJECT_CLASS_ID_OF(GenericBlock)...)},
                REQ    Position         {REQ #variable-position {
                                                       PERM #offset     {ANY_VALUE},
                                                       PERM #separation {ANY_VALUE},
                                                       PERM #alignment  {ANY_VALUE},
                                                       PERM #fill-order {'normal-order'}}},
                REQ    Dimensions               {REQ #horizontal-dimension
                                                       {REQ #fixed-dimension {ANY_VALUE}},
                                                 REQ #vertical-dimension
                                                       {REQ #fixed-dimension {ANY_VALUE}}}
            },
        REQApplication-comments         {REQ  #constraint-name {"17"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC:
        CASE $DAC OF {
            $FDA:
                PERM  Object-class              {OBJECT_CLASS_ID_OF
                                                             (ArrangedContentVariable)}
            $FPDA:
                REQ   Object-class              {OBJECT_CLASS_ID_OF
                                                             (ArrangedContentVariable)}
            },
        REQSubordinates                   {SUB_ID_OF(GenericBlock)+},
        PERM     Imaging-order            {SUB_ID_OF(GenericBlock)+},
        PERM     Application-comments     {REQ  #constraint-name {"17"},
                                                 PERM #external-data {ANY_VALUE}}
}
```

7.4.3.20 SourcedContentFixed

```
:ANY-FRAME-VARIABLE {
GENERIC:
          CASE $DAC OF {
               $PDA-FPDA:
                    REQ    Logical-source             {OBJECT_CLASS_ID_OF(CommonContent)},
                    REQ    Position                   {REQ #fixed-position
                                                                {REQ #horizontal-position {ANY_VALUE},
                                                                 REQ #vertical-position {ANY_VALUE}}},
                    REQ    Dimensions                          {REQ #horizontal-dimension
                                                                {REQ #fixed-dimension {ANY_VALUE}},
                                                               REQ #vertical-dimension
                                                                {REQ #fixed-dimension {ANY_VALUE}}},

               CASE SUPERIOR ({CompositeHeader | CompositeFooter | TableHeader
                           | FixedCompositeBody | CompositeCommon
                           | SynchronizedColumns} (Layout-path)) OF {

               {'270-degrees'}:      -- H/F layout A1 or B2 when the immediate superior is CompositeHeader, CompositeFooter
                                     or TableHeader, or --
                                              -- body layout A when the immediate superior is FixedCompositeBody,
                                     CompositeCommon or
                                     SynchronizedColumns --
                         PERM Layout-path  {'270-degrees'  -- H/F layout A1 or body layout A --
                                                    |'180-degrees'} -- H/F layout B2 --
               {'180-degrees'}:      -- H/F layout B1 when the immediate superior is CompositeHeader or CompositeFooter, or
                                     -- body layout C when the immediate superior is FixedCompositeBody, CompositeCommon
                                     or SynchronizedColumns --
                         REQ  Layout-path   {'180-degrees'} -- H/F layout B1 or body layout C --
               {'  0-degrees'}:      -- H/F layout A2 when the immediate superior is CompositeHeader or CompositeFooter, o
                                    -- body layout B when the immediate superior is FixedCompositeBody, CompositeCommon or
                                    SynchronizedColumns --
                         PERM Layout-path  {'270-degrees'    -- H/F layout A2 --
                                                    |'0-degrees'} -- body layout B --
               }    },
          REQApplication-comments           {REQ  #constraint-name {"18"},
                                                      PERM #external-data {ANY_VALUE}}
SPECIFIC:
          CASE $DAC OF {
               $FDA:
                    PERM  Object-class                {OBJECT_CLASS_ID_OF
                                                               (SourcedContentFixed)}
               $FPDA:
                    REQ   Object-class                {OBJECT_CLASS_ID_OF
                                                               (SourcedContentFixed)}
               },
          REQSubordinates                   {SUB_ID_OF(SpecificBlock)+},
          PERM     Application-comments     {REQ  #constraint-name {"18"},
                                                      PERM #external-data {ANY_VALUE}}
}
```

7.4.3.21 SourcedContentVariable

```
:ANY-FRAME-VARIABLE {
GENERIC:
          CASE $DAC OF {
               $PDA-FPDA:
                  REQ      Logical-source     {OBJECT_CLASS_ID_OF(CommonContent)},
                    REQ    Position           {REQ #variable-position {
                                                               PERM #offset     {ANY_VALUE},
                                                               PERM #separation {ANY_VALUE},
                                                               PERM #alignment  {ANY_VALUE},
                                                               PERM #fill-order {'normal-order'}}},
        CASE SUPERIOR ({CompositeHeader | CompositeFooter | VariableCompositeBody
                | CompositeColumnVariable | CompositeColumnFixed | CompositeCommon
                | SnakingColumns | CompositeFloat} (Layout-path)) OF {

        {'270-degrees'}:    -- H/F layout A1 or B2 when the immediate superior is CompositeHeader or
                            CompositeFooter, or --
                                -- body layout A when the immediate superior is VariableCompositeBody,
                                CompositeColumnVariable, CompositeColumnFixed or CompositeCommon, or --
                                -- body layout B or C when the immediate superior is SnakingColumns or
CompositeFloat --
                REQ  Dimensions         {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #rule-b {ANY_VALUE}
                                                |REQ #maximum-size {'applies'}},
                                               REQ #vertical-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #rule-b {ANY_VALUE}
                                                |REQ #maximum-size {'applies'}}},
                PERM Layout-path                {'270-degrees'  -- H/F layout A1 or body layout A --
                                                |'180-degrees'  -- H/F layout B2 or body layout C --
                                                |'0-degrees'}   -- body layout B --

        {'180-degrees'}:    -- H/F layout B1 when the immediate superior is CompositeHeader or CompositeFooter, or --
                                -- body layout C when the immediate superior is VariableCompositeBody,
                                CompositeColumnVariable, CompositeColumnFixed or CompositeCommon, or --
                                -- body layout A when the immediate superior is SnakingColumns or
CompositeFloat --
                REQ  Dimensions         {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #rule-b {ANY_VALUE}},
                                               REQ #vertical-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #rule-b {ANY_VALUE}
                                                |REQ #maximum-size {'applies'}}},
                REQ  Layout-path                {'180-degrees'  -- H/F layout B1 or body layout C --
                                                |'270-degrees'}-- body layout A --
            {'  0-degrees'}:  -- H/F layout A2 when the immediate superior is CompositeHeader or CompositeFooter, or --
                            -- body layout B when the immediate superior is VariableCompositeBody,
                            CompositeColumnVariable, CompositeColumnFixed or CompositeCommon, or --
                                -- body layout A when the immediate superior is SnakingColumns or
                                CompositeFloat --
                REQ  Dimensions         {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #rule-b {ANY_VALUE}}
                                                |REQ #maximum-size {'applies'}},
                                               REQ #vertical-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #rule-b {ANY_VALUE}
                                                |REQ #maximum-size {'applies'}}},
                PERM Layout-path                {'270-degrees'  -- H/F layout A2 or body layout A --
                                                |'0-degrees'} -- body layout B --

        {'90-degrees'}:     -- body layout B when the immediate superior is SnakingColumns, or --
                                -- body layout B or C when the immediate superior is CompositeFloat --
                REQ  Dimensions         {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #rule-b {ANY_VALUE}
                                                |REQ #maximum-size {'applies'}},
                                               REQ #vertical-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #rule-b {ANY_VALUE}}},
                PERM Layout-path                {'0-degrees'  -- body layout B --
                                                |'180-degrees'}-- body layout C --
            }   },

    REQ     Application-comments    {REQ  #constraint-name {"19"},
                                            PERM #external-data {ANY_VALUE}}
```

```
SPECIFIC:
            CASE $DAC OF {
                  $FDA:
                        PERM  Object-class                    {OBJECT_CLASS_ID_OF
                                                                            (SourcedContentVariable)}
                  $FPDA:
                        REQ   Object-class                    {OBJECT_CLASS_ID_OF
                                                                            (SourcedContentVariable)}
                  },
            REQSubordinates                        {SUB_ID_OF(SpecificBlock)+},
            PERM      Application-comments     {REQ  #constraint-name {"19"},
                                                          PERM #external-data {ANY_VALUE}}
}
```

7.4.3.22 CompositeFixtureVariable

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
            $PDA-FPDA:
                    REQ       Generator-for-subordinates            {$CompositeFixtureVariableGFS},
                    REQ       Position             {REQ #variable-position {
                                                            PERM #offset     {ANY_VALUE},
                                                            PERM #separation {ANY_VALUE},
                                                            PERM #alignment  {ANY_VALUE},
                                                            PERM #fill-order {'normal-order'}}},

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

                    {'270-degrees'}: -- body layout A --
                                 REQ  Dimensions           {REQ #horizontal-dimension
                                                             {REQ #fixed-dimension {ANY_VALUE}
                                                             |REQ #maximum-size {'applies'}
                                                             |REQ #rule-b {ANY_VALUE}},
                                                            REQ #vertical-dimension
                                                             {REQ #fixed-dimension {ANY_VALUE}
                                                             |REQ #rule-b {ANY_VALUE}}},
                                 PERM Layout-path          {'270-degrees'|'180-degrees'
                                                           |'0-degrees'}

                    {'0-degrees'}: -- body layout B --
                                 REQ  Dimensions           {REQ #horizontal-dimension
                                                             {REQ #fixed-dimension {ANY_VALUE}
                                                             |REQ #rule-b {ANY_VALUE}},
                                                            REQ #vertical-dimension
                                                             {REQ #fixed-dimension {ANY_VALUE}
                                                             |REQ #maximum-size {'applies'}
                                                             |REQ #rule-b {ANY_VALUE}}},
                                 REQ  Layout-path          {'0-degrees'|'90-degrees'
                                                           |'270-degrees'}

                    {'180-degrees'}: -- body layout C --
                                 REQ  Dimensions           {REQ #horizontal-dimension
                                                             {REQ #fixed-dimension {ANY_VALUE}
                                                             |REQ #rule-b {ANY_VALUE}},
                                                            REQ #vertical-dimension
                                                             {REQ #fixed-dimension {ANY_VALUE}
                                                             |REQ #maximum-size {'applies'}
                                                             |REQ #rule-b {ANY_VALUE}}},
                                 REQ  Layout-path          {'180-degrees'|'270-degrees'}
          }  },
    REQ    Application-comments                 {REQ  #constraint-name {"20"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
            $FDA:
                    PERM  Object-class            {OBJECT_CLASS_ID_OF
                                                          (CompositeFixtureVariable)}
            $FPDA:
                    REQ   Object-class            {OBJECT_CLASS_ID_OF
                                                          (CompositeFixtureVariable)}
            },
    REQ    Subordinates                           {SUB_ID_OF(BasicFloat)+,
                                                   SUB_ID_OF(FootnoteArea)+,
                                                   SUB_ID_OF(CompositeArtwork)+,
                                                   SUB_ID_OF(FormArea)+},
    PERM Application-comments                     {REQ  #constraint-name {"20"},
                                                   PERM #external-data {ANY_VALUE}}
}
```

7.4.3.23 CompositeFixtureFixed

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
            $PDA-FPDA:
                    REQ         Generator-for-subordinates            {$CompositeFixtureFixedGFS},
                    REQ         Position              {REQ #fixed-position
                                                        {REQ #horizontal-position {ANY_VALUE},
                                                         REQ #vertical-position {ANY_VALUE}}},

                    CASE SUPERIOR (FixedCompositeBody(Layout-path)) OF {

                    {'270-degrees'}: -- body layout A --
                                REQ  Dimensions           {REQ #horizontal-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}
                                                            |REQ #maximum-size {'applies'}},
                                                           REQ #vertical-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}
                                                            |REQ #rule-b {ANY_VALUE}}},
                                PERM Layout-path          {'270-degrees'|'180-degrees'
                                                          |'0-degrees'}

                    {'0-degrees'}: -- body layout B --
                                REQ  Dimensions           {REQ #horizontal-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}
                                                            |REQ #rule-b {ANY_VALUE}},
                                                           REQ #vertical-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}
                                                            |REQ #maximum-size {'applies'}}},
                                REQ  Layout-path          {'0-degrees'|'90-degrees'
                                                          |'270-degrees'}

                    {'180-degrees'}: -- body layout C --
                                REQ  Dimensions           {REQ #horizontal-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}
                                                            |REQ #rule-b {ANY_VALUE}},
                                                           REQ #vertical-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}
                                                            |REQ #maximum-size {'applies'}}},
                                REQ  Layout-path          {'180-degrees'|'270-degrees'}
            }  },
    REQ    Application-comments                           {REQ  #constraint-name {"21"},
                                                           PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
            $FDA:
                    PERM  Object-class            {OBJECT_CLASS_ID_OF
                                                          (CompositeFixtureFixed)}
            $FPDA:
                    REQ   Object-class            {OBJECT_CLASS_ID_OF
                                                          (CompositeFixtureFixed)}
            },
    REQ    Subordinates                    {SUB_ID_OF(BasicFloat)+,
                                            SUB_ID_OF(FootnoteArea)+,
                                            SUB_ID_OF(CompositeArtwork)+,
                                            SUB_ID_OF(FormArea)+},
    PERM Application-comments              {REQ  #constraint-name {"21"},
                                            PERM #external-data {ANY_VALUE}}
}
```

7.4.3.24 BasicFixture

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
            $PDA-FPDA:
                    REQ    Position          {REQ #fixed-position
                                               {REQ #horizontal-position {ANY_VALUE},
                                                REQ #vertical-position {ANY_VALUE}}},

  -- Note that values of position may usually be "0" for overlapping figure. --

                    REQ   Dimensions         {REQ #horizontal-dimension
                                               {REQ #fixed-dimension {ANY_VALUE}
                                               |REQ #rule-b {ANY_VALUE}},
                                              REQ #vertical-dimension
                                               {REQ #fixed-dimension {ANY_VALUE}
                                               |REQ #rule-b {ANY_VALUE}}},

                    CASE SUPERIOR ({FixedCompositeBody | CompositeArtwork} (Layout-path)) OF {

                    {'270-degrees'}: -- body layout A --
                                PERM Layout-path        {'270-degrees'|'180-degrees'}

                    {'0-degrees'}: -- body layout B --
                                REQ  Layout-path        {'0-degrees'|'270-degrees'}

                    {'180-degrees'}: -- body layout C --
                                PERM Layout-path        {'180-degrees'|'270-degrees'}
            }   },
    REQ    Application-comments            {REQ  #constraint-name {"22"},
                                            PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
            $FDA:
                    PERM  Object-class       {OBJECT_CLASS_ID_OF(BasicFixture)}
            $FPDA:
                    REQ   Object-class       {OBJECT_CLASS_ID_OF(BasicFixture)}
            },
    REQ    Subordinates                    {SUB_ID_OF(SpecificBlock)+},
    PERM Application-comments              {REQ  #constraint-name {"22"},
                                            PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Permitted-categories              {ANY_STRING...}
}
```

7.4.3.25 CompositeColumnFixed

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
            $PDA-FPDA:
                    REQ   Generator-for-subordinates    {$CompositeColumnFixedGFS},
                    REQ   Position           {REQ #fixed-position
                                               {REQ #horizontal-position {ANY_VALUE},
                                                REQ #vertical-position   {ANY_VALUE}}},

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

```
                {'270-degrees'}: -- body layout A --
                        REQ  Dimensions          {REQ #horizontal-dimension
                                                   {REQ #fixed-dimension {ANY_VALUE}
                                                   |REQ #maximum-size {'applies'}},
                                                  REQ #vertical-dimension
                                                   {REQ #rule-b {ANY_VALUE}
                                                   |REQ #maximum-size {'applies'}}},
                        PERM Layout-path         {'270-degrees'}

                {'0-degrees'}: -- body layout B --
                        REQ  Dimensions          {REQ #horizontal-dimension
                                                   {REQ #rule-b {ANY_VALUE}
                                                   |REQ #maximum-size {'applies'}},
                                                  REQ #vertical-dimension
                                                   {REQ #fixed-dimension {ANY_VALUE}
                                                   |REQ #maximum-size {'applies'}}},
                        REQ  Layout-path         {'0-degrees'}

                {'180-degrees'}: -- body layout C --
                        REQ  Dimensions          {REQ #horizontal-dimension
                                                   {REQ #maximum-size {'applies'}},
                                                  REQ #vertical-dimension
                                                   {REQ #fixed-dimension {ANY_VALUE}
                                                   |REQ #maximum-size {'applies'}}},
                        REQ  Layout-path         {'180-degrees'}
          }  },
    REQ   Application-comments                {REQ  #constraint-name {"23"},
                                               PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
          $FDA:
                PERM  Object-class            {OBJECT_CLASS_ID_OF
                                                       (CompositeColumnFixed)}
          $FPDA:
                REQ   Object-class            {OBJECT_CLASS_ID_OF
                                                       (CompositeColumnFixed)}
          },
    REQ   Subordinates                        {SUB_ID_OF(BasicFloat)+,
                                               SUB_ID_OF(CompositeFloat)+,
                                               SUB_ID_OF(FootnoteArea)+,
                                               SUB_ID_OF(CompositeFixtureVariable)+,
                                               SUB_ID_OF(TableArea)+,
                                               SUB_ID_OF(ArrangedContentVariable)+,
                                               SUB_ID_OF(SourcedContentVariable)+},
    PERM Application-comments                 {REQ  #constraint-name {"23"},
                                               PERM #external-data {ANY_VALUE}}
}
```

7.4.3.26 CompositeColumnVariable

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
          $PDA-FPDA:
                REQ   Generator-for-subordinates   {$CompositeColumnVariableGFS},
                REQ   Position                {REQ #variable-position {
                                                      PERM #offset     {ANY_VALUE},
                                                      PERM #separation {ANY_VALUE},
                                                      PERM #alignment  {ANY_VALUE},
                                                      PERM #fill-order {'normal-order'}}},

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

                {'270-degrees'}:  -- body layout A --
                        REQ  Dimensions          {REQ #horizontal-dimension
                                                   {REQ #fixed-dimension {ANY_VALUE}},
                                                  REQ #vertical-dimension
                                                   {REQ #rule-b {ANY_VALUE}
                                                   |REQ #maximum-size {'applies'}}},
                        PERM Layout-path         {'270-degrees'}
                {'0-degrees'}: -- body layout B --
                     REQ  Dimensions         {REQ #horizontal-dimension
                                              {REQ #rule-b {ANY_VALUE}
                                              |REQ #maximum-size {'applies'}},
                                             REQ #vertical-dimension
                                              {REQ #fixed-dimension {ANY_VALUE}}},
                     REQ  Layout-path        {'0-degrees'}

                {'180-degrees'}: -- body layout C --
                     REQ  Dimensions         {REQ #horizontal-dimension
                                              {REQ #rule-b {ANY_VALUE}
                                              |REQ #maximum-size {'applies'}},
                                             REQ #vertical-dimension
                                              {REQ #fixed-dimension {ANY_VALUE}}},
                     REQ  Layout-path        {'180-degrees'}
          }   },
     REQ   Application-comments           {REQ  #constraint-name {"24"},
                                           PERM #external-data {ANY_VALUE}}

SPECIFIC:
     CASE $DAC OF {
          $FDA:
                PERM  Object-class        {OBJECT_CLASS_ID_OF
                                                (CompositeColumnVariable)}
          $FPDA:
                REQ   Object-class        {OBJECT_CLASS_ID_OF
                                                (CompositeColumnVariable)}
          },
     REQ   Subordinates                   {SUB_ID_OF(BasicFloat)+,
                                           SUB_ID_OF(CompositeFloat)+,
                                           SUB_ID_OF(FootnoteArea)+,
                                           SUB_ID_OF(CompositeFixtureVariable)+,
                                           SUB_ID_OF(TableArea)+,
                                           SUB_ID_OF(ArrangedContentVariable)+,
                                           SUB_ID_OF(SourcedContentVariable)+},
     PERM Application-comments            {REQ  #constraint-name {"24"},
                                           PERM #external-data {ANY_VALUE}}
}
```

7.4.3.27 CompositeCommon

```
:ANY-FRAME-FIXED {
GENERIC:
     CASE $DAC OF {
          $PDA-FPDA:
                REQ   Generator-for-subordinates   {$CompositeCommonGFS},
                CASE SUPERIOR (FixedCompositeBody(Layout-path)) OF {

                     {'270-degrees'}: -- body layout A --
                          PERM  Layout-path       {'270-degrees'}

                     {'0-degrees'}: -- body layout B --
                          PERM  Layout-path       {'0-degrees'}

                     {'180-degrees'}: -- body layout C --
                          PERM  Layout-path       {'180-degrees'}
          } },
     REQ   Application-comments           {REQ  #constraint-name {"25"},
                                           PERM #external-data {ANY_VALUE}}
SPECIFIC:
     CASE $DAC OF {
          $FDA:
                PERM  Object-class   {OBJECT_CLASS_ID_OF(CompositeCommon)}
          $FPDA:
                REQ   Object-class   {OBJECT_CLASS_ID_OF(CompositeCommon)}
          },
     REQ   Subordinates                   {SUB_ID_OF(SourcedContentFixed)+,
                                           SUB_ID_OF(ArrangedContentFixed)+,
                                           SUB_ID_OF(SourcedContentVariable)+,
                                           SUB_ID_OF(ArrangedContentVariable)+},
    PERM Imaging-order                      {SUB_ID_OF(SourcedContentFixed)+,
                                             SUB_ID_OF(ArrangedContentFixed)+},
    PERM Application-comments               {REQ  #constraint-name {"25"},
                                             PERM #external-data {ANY_VALUE}}
}
```

7.4.3.28 CompositeArtwork

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
            REQ   Generator-for-subordinates                {$CompositeArtworkGFS},
            REQ   Position                  {REQ #variable-position {
                                                    PERM #offset     {ANY_VALUE},
                                                    PERM #separation {ANY_VALUE},
                                                    PERM #alignment  {ANY_VALUE},
                                                    PERM #fill-order {'normal-order'}}},

            REQ        Dimensions           {REQ #horizontal-dimension
                                                     {REQ #fixed-dimension {ANY_VALUE}
                                                     |REQ #rule-b {ANY_VALUE}},
                                             REQ #vertical-dimension
                                                     {REQ #fixed-dimension {ANY_VALUE}
                                                     |REQ #rule-b {ANY_VALUE}}},

            CASE SUPERIOR ({CompositeFixtureVariable | CompositeFixtureFixed}
                                                         (Layout-path)) OF {
                {'270-degrees'}:
                    PERM Layout-path        {'270-degrees'}
                {'0-degrees'}:
                    REQ  Layout-path        {'0-degrees'}

                {'180-degrees'}:
                    REQ  Layout-path        {'180-degrees'}
        } },
    REQ   Application-comments              {REQ  #constraint-name {"26"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
            PERM  Object-class   {OBJECT_CLASS_ID_OF(CompositeArtwork)}
        $FPDA:
            REQ   Object-class   {OBJECT_CLASS_ID_OF(CompositeArtwork)}
        },
    REQ   Subordinates                      {SUB_ID_OF(BasicFixture)+},
    PERM Imaging-order                      {SUB_ID_OF(BasicFixture)+},
    PERM Application-comments               {REQ  #constraint-name {"26"},
                                             PERM #external-data {ANY_VALUE}}
}
```

7.4.3.29 BasicHeader

```
:ANY-FRAME-FIXED {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
            REQ   Logical-source            {OBJECT_CLASS_ID_OF(CommonContent)}
        },
    PERM Layout-path                        {'270-degrees' -- H/F layout A1 --
                                            |'180-degrees' -- H/F layout B1 --},
    REQ   Application-comments              {REQ  #constraint-name {"27"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
            PERM  Object-class              {OBJECT_CLASS_ID_OF(BasicHeader)}
        $FPDA:
            REQ   Object-class              {OBJECT_CLASS_ID_OF(BasicHeader)}
        },
    REQ    Subordinates                      {SUB_ID_OF(SpecificBlock)+},
    PERM Application-comments                {REQ  #constraint-name {"27"},
                                              PERM #external-data {ANY_VALUE}}
}
```

7.4.3.30 BasicFooter

```
:ANY-FRAME-FIXED {
GENERIC:
    CASE $DAC OF {
            $PDA-FPDA:
                    REQ   Logical-source     {OBJECT_CLASS_ID_OF(CommonContent)}
            },
    PERM Layout-path                         {'270-degrees' -- H/F layout A1 --
                                             |'180-degrees' -- H/F layout B1 --},
    REQ    Application-comments              {REQ  #constraint-name {"33"},
                                              PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
            $FDA:
                    PERM  Object-class       {OBJECT_CLASS_ID_OF(BasicFooter)}
            $FPDA:
                    REQ   Object-class       {OBJECT_CLASS_ID_OF(BasicFooter)}
            },
    REQ    Subordinates                      {SUB_ID_OF(SpecificBlock)+},
    PERM Application-comments                {REQ  #constraint-name {"33"},
                                              PERM #external-data {ANY_VALUE}}
}
```

7.4.3.31 BasicBody

```
:ANY-FRAME-FIXED {
GENERIC:
    PERM Layout-path                         {'270-degrees' -- body layout A --
                                             |'0-degrees'   -- body layout B --
                                             |'180-degrees' -- body layout C --},
    REQ    Application-comments              {REQ  #constraint-name {"28"},
                                              PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
            $FDA:
                    PERM  Object-class       {OBJECT_CLASS_ID_OF(BasicBody)}
            $FPDA:
                    REQ   Object-class       {OBJECT_CLASS_ID_OF(BasicBody)}
            },
    REQ    Subordinates                      {SUB_ID_OF(SpecificBlock)+},
    PERM Application-comments                {REQ  #constraint-name {"28"},
                                              PERM #external-data {ANY_VALUE}}
}
```

7.4.3.32 GenericBlock

```
{
GENERIC:
    REQ    Object-type                       {'block'},
    REQ    Object-class-identifier           {ANY_VALUE},
    REQ    Content-architecture-class        {$FC | $FPC | $FPR | $FPG },
    PERM Content-generator                   {$GENERICBLOCKREF},
    PERM   Content-portions                  {CONTENT_ID_OF(Character-content-portion)+
                                             | CONTENT_ID_OF(Raster-graphics-content-portion)
                                             | CONTENT_ID_OF(Geometric-graphics-content-portion) },
    PERM  Presentation-style                 {STYLE_ID_OF(PStyle1)
                                             | STYLE_ID_OF(PStyle2)
                                             | STYLE_ID_OF(PStyle3)},
    PERM Resource                            {ANY_VALUE},
    REQ    Application-comments              {REQ  #constraint-name  {"29"},
                                              PERM #external-data {ANY_VALUE}}
SPECIFIC:
    REQ   Object-type                         {'block'},
    REQ   Object-identifier                   {ANY_VALUE},
    CASE $DAC OF {
        $FDA:
                PERM  Object-class            {OBJECT_CLASS_ID_OF(GenericBlock)}
        $FPDA:
                REQ   Object-class            {OBJECT_CLASS_ID_OF(GenericBlock)}
        },
    PERM   Presentation-style                 {STYLE_ID_OF(PStyle1)
                                              | STYLE_ID_OF(PStyle2)
                                              | STYLE_ID_OF(PStyle3)},
    PERM Content-architecture-class           {$FC | $FPC | $FPR | $FPG },
    CASE  GenericBlock (Object-class) OF {
        VOID:
                REQ   Content-portions    {CONTENT_ID_OF(Character-content-portion)+
                                              |CONTENT_ID_OF(Raster-graphics-content-portion)
                                              |CONTENT_ID_OF(Geometric-graphics-content-portion)}
        },
    PERM  Presentation-attributes             {
        PERM  #character-attributes           {
            PERM  #alignment                          {ANY_VALUE},
            PERM  #character-fonts                    {ANY_VALUE},
            PERM  #character-path                     {ANY_VALUE},
            PERM  #character-spacing                  {ANY_VALUE},
            PERM  #character-orientation              {ANY_VALUE},
            PERM  #code-extension-announcers          {$CDEXTEN},
            PERM  #first-line-offset                  {ANY_VALUE},
            PERM  #graphic-character-sets             {$PERMIT-GRCHAR},
            PERM  #graphic-character-subrepertoire    {ANY_VALUE},
            PERM  #graphic-rendition                  {$GRAPHICRENDITIONS},
            PERM  #itemization                        {ANY_VALUE},
            PERM  #kerning-offset                     {ANY_VALUE},
            PERM  #line-layout-table                  {ANY_VALUE},
            PERM  #line-progression                   {ANY_VALUE},
            PERM  #line-spacing                       {ANY_VALUE},
            PERM  #pairwise-kerning                   {ANY_VALUE},
            PERM  #formatting-indicator               {ANY_VALUE},
            PERM  #initial-offset                     {ANY_VALUE}
            }
    },
    PERM Application-comments                 {REQ  #constraint-name  {"29"},
                                               PERM #external-data{ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Position                             {REQ #fixed-position
                                                      {REQ #horizontal-position {ANY_VALUE},
                                                       REQ #vertical-position {ANY_VALUE}}},
    PERM Dimensions                           {REQ #horizontal-dimension
                                                      {REQ #fixed-dimension {ANY_VALUE}},
                                               REQ #vertical-dimension
                                                      {REQ #fixed-dimension {ANY_VALUE}}},
    PERM  Transparency                        {ANY_VALUE},
    PERM  Colour                              {ANY_VALUE},
    PERM  Border                              {ANY_VALUE},
    PERM  User-readable-comments              {ANY_STRING},
    PERM  User-visible-name                   {ANY_STRING}
}
```

7.4.3.33 SpecificBlock

```
{
SPECIFIC:
    REQ   Object-type                         {'block'},
    REQ   Object-identifier                   {ANY_VALUE},
    REQ   Content-portions                    {CONTENT_ID_OF(Character-content-portion)+
                                              |CONTENT_ID_OF(Raster-graphics-content-portion)
                                              |CONTENT_ID_OF(Geometric-graphics-content-portion)},
    PERM Position                          {REQ #fixed-position
                                                  {REQ #horizontal-position {ANY_VALUE},
                                                   REQ #vertical-position {ANY_VALUE}}},
    PERM Dimensions                        {REQ #horizontal-dimension
                                                  {REQ #fixed-dimension {ANY_VALUE}},
                                            REQ #vertical-dimension
                                                  {REQ #fixed-dimension {ANY_VALUE}}},
    PERM  Presentation-style               {STYLE_ID_OF(PStyle1)
                                           | STYLE_ID_OF(PStyle2)
                                           | STYLE_ID_OF(PStyle3)
                                           | STYLE_ID_OF(PStyle4)},
    PERM Content-architecture-class        {$FC | $FPC | $FPR | $FPG},
    PERM  Presentation-attributes          {
         PERM  #character-attributes    {
         PERM  #alignment                        {ANY_VALUE},
         PERM  #character-fonts                  {ANY_VALUE},
         PERM  #character-path                   {ANY_VALUE},
         PERM  #character-spacing                {ANY_VALUE},
         PERM  #character-orientation            {ANY_VALUE},
         PERM  #code-extension-announcers        {$CDEXTEN},
         PERM  #first-line-offset                {ANY_VALUE},
         PERM  #graphic-character-sets           {$PERMIT-GRCHAR },
         PERM  #graphic-character-subrepertoire  {ANY_VALUE},
         PERM  #graphic-rendition                {$GRAPHICRENDITIONS},
         PERM  #itemization                      {ANY_VALUE},
         PERM  #kerning-offset                   {ANY_VALUE},
         PERM  #line-layout-table                {ANY_VALUE},
         PERM  #line-progression                 {ANY_VALUE},
         PERM  #line-spacing                     {ANY_VALUE},
         PERM  #pairwise-kerning                 {ANY_VALUE},
         PERM  #formatting-indicator             {ANY_VALUE},
         PERM  #initial-offset                   {ANY_VALUE}
         }
    },
    PERM  Transparency                     {ANY_VALUE},
    PERM  Colour                           {ANY_VALUE},
    PERM  Border                           {ANY_VALUE },
    PERM  User-readable-comments           {ANY_STRING},
    PERM  User-visible-name                {ANY_STRING},
    PERM  Application-comments             {REQ  #constraint-name {"30"},
                                            PERM #external-data  {ANY_VALUE}}
}
```

7.4.3.34 FormArea

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
         $PDA-FPDA:
              REQ   Generator-for-subordinates   {$FormAreaGFS}},
    REQ    Position                        {REQ #variable-position {
                                                PERM #offset     {ANY_VALUE},
                                                PERM #separation {ANY_VALUE},
                                                PERM #alignment  {ANY_VALUE},
                                                PERM #fill-order {'normal-order'}}},
    REQ    Dimensions                      {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}},
                                            REQ #vertical-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Layout-path                       {'270-degrees'},

    REQ    Application-comments            {REQ  #constraint-name {"31"},
                                            PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
         $FDA:
              PERM  Object-class           {OBJECT_CLASS_ID_OF(FormArea)}
         $FPDA:
              REQ   Object-class           {OBJECT_CLASS_ID_OF(FormArea)}
         },
    REQ   Subordinates                      {SUB_ID_OF(ArrangedContentFixed)+,
                                             SUB_ID_OF(FormEntryArea)+,
                                             SUB_ID_OF(EntryGroupArea)+},
    PERM  Position                          {ANY_VALUE},
    PERM  Dimensions                        {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}},
                                             REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Application-comments               {REQ  #constraint-name {"31"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency                       {ANY_VALUE},
    PERM Colour                             {ANY_VALUE},
    PERM Border                             {ANY_VALUE}
}
```

7.4.3.35 EntryGroupArea

```
:ANY-FRAME-FIXED {
GENERIC:
    CASE $DAC OF {
            $PDA-FPDA:
                    REQ   Generator-for-subordinates    {$EntryGroupAreaGFS}
            },
    REQ   Position                          {REQ #fixed-position
                                                    {REQ #horizontal-position {ANY_VALUE},
                                                     REQ #vertical-position {ANY_VALUE}}},
    REQ   Dimensions                        {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}},
                                             REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}}},
    PERM  Layout-path                       {'270-degrees'},
    REQ   Application-comments              {REQ  #constraint-name {"35"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
            $FDA:
                    PERM  Object-class      {OBJECT_CLASS_ID_OF(EntryGroupArea)}
            $FPDA:
                    REQ   Object-class      {OBJECT_CLASS_ID_OF(EntryGroupArea)}
            },
    REQ   Subordinates                      {SUB_ID_OF(ArrangedContentFixed)+,
                                             SUB_ID_OF(FormEntryArea)+,
                                             SUB_ID_OF(EntryGroupArea)+},
    PERM Position                           {ANY_VALUE},
    PERM Dimensions                         {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}},
                                             REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Imaging-order                      {ANY_VALUE},
    PERM Application-comments               {REQ  #constraint-name {"35"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency                       {ANY_VALUE},
    PERM Colour                             {ANY_VALUE},
    PERM Border                             {ANY_VALUE}
}
```

7.4.3.36 TableArea

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
            $PDA-FPDA:
                    REQ   Generator-for-subordinates    {$TableAreaGFS}
            },
    REQ    Position                         {REQ #variable-position {
                                                    PERM #offset     {ANY_VALUE},
                                                    PERM #separation {ANY_VALUE},
                                                    PERM #alignment  {ANY_VALUE},
                                                    PERM #fill-order {'normal-order'}}},
    REQ    Dimensions                       {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}},
                                             REQ #vertical-dimension
                                                    {REQ #rule-b {ANY_VALUE}
                                                    |REQ #fixed-dimension {ANY_VALUE}}},
    PERM  Layout-path                       {'270-degrees'},
    REQ    Application-comments             {REQ  #constraint-name {"36"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
            PERM  Object-class              {OBJECT_CLASS_ID_OF(TableArea)}
        $FPDA:
            REQ   Object-class              {OBJECT_CLASS_ID_OF(TableArea)}
        },
    REQ    Subordinates                     {SUB_ID_OF(RowArea)+,
                                             SUB_ID_OF(TableLabel)+,
                                             SUB_ID_OF(TableHeader)},
    PERM Position                           {ANY_VALUE},
    PERM Dimensions                         {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}},
                                             REQ #vertical-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Application-comments               {REQ  #constraint-name {"36"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency                       {ANY_VALUE},
    PERM Colour                             {ANY_VALUE},
    PERM Border                             {ANY_VALUE}
}
```

7.4.3.37 TableHeader

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
            REQ   Generator-for-subordinates    {$TableHeaderGFS}
        },
    REQ    Position                         {REQ #variable-position {
                                                    PERM #offset     {ANY_VALUE},
                                                    PERM #separation {ANY_VALUE},
                                                    PERM #alignment  {ANY_VALUE},
                                                    PERM #fill-order {'normal-order'}}},
    REQ    Dimensions                       {REQ #horizontal-dimension
                                                    {REQ #fixed-dimension {ANY_VALUE}},
                                             REQ #vertical-dimension
                                                    {REQ #rule-b {ANY_VALUE}
                                                    |REQ #fixed-dimension {ANY_VALUE}}},
    PERM Layout-path                        {'270-degrees'},
    REQ    Application-comments             {REQ  #constraint-name {"34"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
            PERM  Object-class              {OBJECT_CLASS_ID_OF(TableHeader)}
        $FPDA:
            REQ   Object-class              {OBJECT_CLASS_ID_OF(TableHeader)}
        },
    REQ   Subordinates                          {SUB_ID_OF(SourcedContentFixed)+},
    PERM Position                               {ANY_VALUE},
    PERM Dimensions                             {REQ #horizontal-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}},
                                                 REQ #vertical-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Application-comments                   {REQ  #constraint-name {"34"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency                           {ANY_VALUE},
    PERM Colour                                 {ANY_VALUE},
    PERM Border                                 {ANY_VALUE}
}
```

7.4.3.38 TableLabel

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
          $PDA-FPDA:
                REQ   Generator-for-subordinates    {$TableLabelGFS}
          },
    REQ   Position                              {REQ #variable-position {
                                                        PERM #offset     {ANY_VALUE},
                                                        PERM #separation {ANY_VALUE},
                                                        PERM #alignment  {ANY_VALUE},
                                                        PERM #fill-order {'normal-order'}}},
    REQ   Dimensions                            {REQ #horizontal-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}},
                                                 REQ #vertical-dimension
                                                        {REQ #rule-b {ANY_VALUE}
                                                        |REQ #fixed-dimension {ANY_VALUE}}},
    PERM  Layout-path                           {'270-degrees'},
    REQ   Application-comments                  {REQ  #constraint-name {"37"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
          $FDA:
                PERM  Object-class              {OBJECT_CLASS_ID_OF(TableLabel)}
          $FPDA:
                REQ   Object-class              {OBJECT_CLASS_ID_OF(TableLabel)}
          },
    REQ   Subordinates                          {SUB_ID_OF(TableLabelContent)+,
                                                 SUB_ID_OF(CompositeTableLabel)},
    PERM Position                               {ANY_VALUE},
    PERM Dimensions                             {REQ #horizontal-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}},
                                                 REQ #vertical-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Application-comments                   {REQ  #constraint-name {"37"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency                           {ANY_VALUE},
    PERM Colour                                 {ANY_VALUE},
    PERM  Border                                {ANY_VALUE}
}
```

7.4.3.39 CompositeTableLabel

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
          $PDA-FPDA:
                REQ   Generator-for-subordinates    {$CompositeTableLabelGFS}
          },
    REQ   Position                              {REQ #fixed-position {
                                                        REQ #horizontal-position {ANY_VALUE},
                                                        REQ #vertical-position   {ANY_VALUE}}},
    REQ   Dimensions                       {REQ #horizontal-dimension
                                                 {REQ #fixed-dimension {ANY_VALUE}},
                                            REQ #vertical-dimension
                                                 {REQ #rule-b {ANY_VALUE}
                                                 |REQ #fixed-dimension {ANY_VALUE}
                                                 |REQ #maximum-size {'applies'}}},
    PERM  Layout-path                      {'270-degrees'},
    REQ   Application-comments             {REQ  #constraint-name {"38"},
                                            PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
          $FDA:
                PERM  Object-class         {OBJECT_CLASS_ID_OF
                                                 (CompositeTableLabel)}
          $FPDA:
                REQ   Object-class         {OBJECT_CLASS_ID_OF
                                                 (CompositeTableLabel)}
          },
    REQ   Subordinates                     {SUB_ID_OF(LabelComponent)+},
    PERM  Position                         {ANY_VALUE},
    PERM  Dimensions                       {REQ #horizontal-dimension
                                                 {REQ #fixed-dimension {ANY_VALUE}},
                                            REQ #vertical-dimension
                                                 {REQ #fixed-dimension {ANY_VALUE}}},
    PERM  Imaging-order                    {ANY_VALUE},
    PERM  Application-comments             {REQ  #constraint-name {"38"},
                                            PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM  Transparency                     {ANY_VALUE},
    PERM  Colour                           {ANY_VALUE},
    PERM  Border                           {ANY_VALUE}
}
```

7.4.3.40 LabelComponent

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
          $PDA-FPDA:
                REQ   Generator-for-subordinates   {$LabelComponentGFS}
          },
    REQ   Position                         {REQ #variable-position {
                                                 PERM #offset     {ANY_VALUE},
                                                 PERM #separation {ANY_VALUE},
                                                 PERM #alignment  {ANY_VALUE},
                                                 PERM #fill-order {'normal-order'}}},
    REQ   Dimensions                       {REQ #horizontal-dimension
                                                 {REQ #fixed-dimension {ANY_VALUE}},
                                            REQ #vertical-dimension
                                                 {REQ #rule-b {ANY_VALUE}
                                                 |REQ #fixed-dimension {ANY_VALUE}
                                                 |REQ #maximum-size {'applies'}}},
    PERM  Layout-path                      {'270-degrees'},
    REQ   Application-comments             {REQ  #constraint-name {"39"},
                                            PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
          $FDA:
                PERM  Object-class         {OBJECT_CLASS_ID_OF(LabelComponent)}
          $FPDA:
                REQ   Object-class         {OBJECT_CLASS_ID_OF(LabelComponent)}
          },
    REQ   Subordinates                     {SUB_ID_OF(TableLabelContent)+},
    PERM  Position                         {ANY_VALUE},
    PERM  Dimensions                       {REQ #horizontal-dimension
                                                 {REQ #fixed-dimension {ANY_VALUE}},
                                            REQ #vertical-dimension
                                                 {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Application-comments               {REQ  #constraint-name {"39"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency                       {ANY_VALUE},
    PERM Colour                             {ANY_VALUE},
    PERM Border                             {ANY_VALUE}
}
```

7.4.3.41 RowArea

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
            REQ   Generator-for-subordinates    {$RowAreaGFS}
        },
    REQ    Position                         {REQ #variable-position {
                                                PERM #offset     {ANY_VALUE},
                                                PERM #separation {ANY_VALUE},
                                                PERM #alignment  {ANY_VALUE},
                                                PERM #fill-order {'normal-order'}}},
    REQ    Dimensions                       {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}},
                                             REQ #vertical-dimension
                                                {REQ #rule-b {ANY_VALUE}
                                                |REQ #fixed-dimension {ANY_VALUE}}},
    PERM Layout-path                        {'270-degrees'},
    REQ    Application-comments             {REQ  #constraint-name {"40"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
            PERM  Object-class              {OBJECT_CLASS_ID_OF(RowArea)}
        $FPDA:
            REQ   Object-class              {OBJECT_CLASS_ID_OF(RowArea)}
        },
    REQ    Subordinates                     {SUB_ID_OF(Cell)+,
                                             SUB_ID_OF(SubRowGroup)},
    PERM Position                           {ANY_VALUE},
    PERM Dimensions                         {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}},
                                             REQ #vertical-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Application-comments               {REQ #constraint-name {"40"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency                       {ANY_VALUE},
        PERM Colour                         {ANY_VALUE},
    PERM Border                             {ANY_VALUE}
}
```

7.4.3.42 Cell

```
:ANY-FRAME-VARIABLE {
GENERIC:
    REQ    Position                         {REQ #fixed-position {
                                                REQ #horizontal-position {ANY_VALUE},
                                                REQ #vertical-position {ANY_VALUE}}},
    REQ    Dimensions                       {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}},
                                             REQ #vertical-dimension
                                                {REQ #rule-b {ANY_VALUE}
                                                |REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #maximum-size {'applies'}}},

    PERM Layout-path                            {'270-degrees'},
    REQ    Permitted-categories                 {ANY_STRING...},
            -- category name for tables should be specified --
    REQ    Application-comments             {REQ  #constraint-name {"41"},
                                             PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
            $FDA:
                    PERM  Object-class              {OBJECT_CLASS_ID_OF(Cell)}
            $FPDA:
                    REQ   Object-class              {OBJECT_CLASS_ID_OF(Cell)}
            },
    REQ    Subordinates                             {SUB_ID_OF(SpecificBlock)},
    PERM Position                                   {ANY_VALUE},
    PERM Dimensions                                 {REQ #horizontal-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}},
                                                     REQ #vertical-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Imaging-order                              {ANY_VALUE},
    PERM Application-comments                       {REQ  #constraint-name {"41"},
                                                     PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency                               {ANY_VALUE},
    PERM  Colour                                    {ANY_VALUE},
    PERM  Border                                    {ANY_VALUE}
}
```

7.4.3.43 SubRowGroup

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
            $PDA-FPDA:
                    REQ   Generator-for-subordinates    {$SubRowGroupGFS}
            },
    REQ    Position                                 {REQ #fixed-position {
                                                            REQ #horizontal-position {ANY_VALUE},
                                                            REQ #vertical-position {ANY_VALUE}}},
    REQ    Dimensions                               {REQ #horizontal-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}},
                                                     REQ #vertical-dimension
                                                            {REQ #rule-b {ANY_VALUE}
                                                            |REQ #fixed-dimension {ANY_VALUE}
                                                            |REQ #maximum-size {'applies'}}},
    PERM Layout-path                                {'270-degrees'},
    REQ    Application-comments                     {REQ  #constraint-name {"42"},
                                                     PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
            $FDA:
                    PERM  Object-class              {OBJECT_CLASS_ID_OF(SubRowGroup)}
            $FPDA:
                    REQ   Object-class              {OBJECT_CLASS_ID_OF(SubRowGroup)}
            },
    REQ    Subordinates                             {SUB_ID_OF(SubRow)+},
    PERM Position                                   {ANY_VALUE},
    PERM Dimensions                                 {REQ #horizontal-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}},
                                                     REQ #vertical-dimension
                                                            {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Application-comments                       {REQ  #constraint-name {"42"},
                                                     PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency                               {ANY_VALUE},
    PERM Colour                                     {ANY_VALUE},
    PERM Border                                     {ANY_VALUE}
}
```

7.4.3.44 SubRow

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
            $PDA-FPDA:
                    REQ   Generator-for-subordinates    {$SubRowGFS}
            },
    REQ    Position                     {REQ #variable-position {
                                                PERM #offset     {ANY_VALUE},
                                                PERM #separation {ANY_VALUE},
                                                PERM #alignment  {ANY_VALUE},
                                                PERM #fill-order {'normal-order'}}},
    REQ    Dimensions                   {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}},
                                         REQ #vertical-dimension
                                                {REQ #rule-b {ANY_VALUE}
                                                |REQ #fixed-dimension {ANY_VALUE}
                                                |REQ #maximum-size {'applies'}}},
    PERM Layout-path                    {'270-degrees'},
    REQ    Application-comments         {REQ  #constraint-name {"43"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
            $FDA:
                    PERM  Object-class          {OBJECT_CLASS_ID_OF(SubRow)}
            $FPDA:
                    REQ   Object-class          {OBJECT_CLASS_ID_OF(SubRow)}
            },
    REQ    Subordinates                 {SUB_ID_OF(Cell)+},
    PERM Position                       {ANY_VALUE},
    PERM Dimensions                     {REQ #horizontal-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}},
                                         REQ #vertical-dimension
                                                {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Application-comments           {REQ  #constraint-name {"43"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency                   {ANY_VALUE},
    PERM Colour                         {ANY_VALUE},
    PERM Border                         {ANY_VALUE}
}
```

7.4.3.45 TableLabelContent

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
            $PDA-FPDA:
                    REQ  Logical-source         {OBJECT_CLASS_ID_OF(CommonContent)},
                    REQ  Position               {REQ #fixed-position
                                                        {REQ #horizontal-position {ANY_VALUE},
                                                         REQ #vertical-position {ANY_VALUE}}},
                    REQ  Dimensions             {REQ #horizontal-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}},
                                                 REQ #vertical-dimension
                                                        {REQ #fixed-dimension {ANY_VALUE}
                                                        |REQ #maximum-size {'applies'}}},
                    PERM Layout-path                    {'270-degrees'}
            },
    REQ    Application-comments         {REQ  #constraint-name {"44"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
            $FDA:
                    PERM Object-class           {OBJECT_CLASS_ID_OF
                                                        (TableLabelContent)}
```

```
          $FPDA:
                    REQ  Object-class           {OBJECT_CLASS_ID_OF
                                                         (TableLabelContent)}
          },
     REQ    Subordinates                        {SUB_ID_OF(SpecificBlock)+},
     PERM Application-comments                  {REQ  #constraint-name {"44"},
                                                 PERM #external-data {ANY_VALUE}}
}
```

7.4.3.46 FormEntryArea

```
:ANY-FRAME-VARIABLE {
GENERIC:
     CASE $DAC OF {
          $PDA-FPDA:
                    REQ  Position                         {REQ #fixed-position
                                                          {REQ #horizontal-position {ANY_VALUE},
                                                           REQ #vertical-position {ANY_VALUE}}},
                    REQ  Dimensions             {REQ #horizontal-dimension
                                                          {REQ #fixed-dimension {ANY_VALUE}},
                                                 REQ #vertical-dimension
                                                          {REQ #fixed-dimension {ANY_VALUE}}},
                    PERM Layout-path            {'270-degrees'}
          },
     REQ    Application-comments                {REQ  #constraint-name {"45"},
                                                  PERM #external-data {ANY_VALUE}}
SPECIFIC:
     CASE $DAC OF {
          $FDA:
                    PERM Object-class           {OBJECT_CLASS_ID_OF(FormEntryArea)}
          $FPDA:
                    REQ  Object-class           {OBJECT_CLASS_ID_OF(FormEntryArea)}
          },
     REQ    Subordinates                        {SUB_ID_OF(SpecificBlock)},
     PERM Application-comments                  {REQ  #constraint-name {"45"},
                                                 PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
     PERM Permitted-categories                  {ANY_STRING...}
}
```

7.5 Layout style constituent constraints

7.5.1 Macro definitions

```
DEFINE(SameLayoutObject,"
REQ       {REQ #logical-object  {<object-id-expr>::=PREC-OBJ(CURR-OBJ);}
          |REQ #logical-object {'null'}},
           PERM #layout-object   {'page'        -- to layout object type --
                    | ANY_STRING                -- to layout category --
                    | OBJECT_CLASS_ID_OF(ColumnFixed)
                    | OBJECT_CLASS_ID_OF(ColumnVariable)
                    | OBJECT_CLASS_ID_OF(CompositeColumnFixed)
                    | OBJECT_CLASS_ID_OF(CompositeColumnVariable)}
")
```

7.5.2 Factor constraints

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

7.5.3 Constituent constraints

7.5.3.1 LStyle1

```
:ANY-LAYOUT-STYLE      {
-- This style is used for the constituent constraints Passage, NumberedSegment, Title, Caption, Paragraph, Phrase,
   Footnote, Figure, Reference, Description, NumberedList, UnNumberedList, DefinitionList, ListItem, ListTerm  --

CASE $GLAS OF{
 $COMPLETE:
    PERM     Indivisibility               {ANY_VALUE},
    PERM     Layout-object-class          {ANY_VALUE},
    PERM     New-layout-object            {ANY_VALUE},
    PERM     Same-layout-object           {$SameLayoutObject},
    PERM     Synchronization              {ANY_VALUE}
  VOID:
    PERM     Indivisibility               {ANY_STRING --to layout category--
                                          |'page'    --to layout object type--
                                          |'null'},
    PERM     New-layout-object            {ANY_STRING --to layout category--
                                          |'page'    --to layout object type--
                                          |'null'},
    PERM     Same-layout-object           {REQ  {REQ #logical-object
                                                {<object-id-expr>::=PREC-OBJ(CURR-OBJ);}
                                                | REQ #logical-object {'null'}},
                                           PERM #layout-object
                                                {ANY_STRING --to layout category--
                                                |'page'    --to layout object type--}
                                          },
    PERM     Synchronization              {ANY_VALUE}
}}
```

7.5.3.2 LStyle2

```
:ANY-LAYOUT-STYLE      {
-- This style is used for the constituent constraints BodyText and Number  --
CASE $GLAS OF{
  $COMPLETE:
    PERM     Block-alignment              {ANY_VALUE},
    PERM     Concatenation                {ANY_VALUE},
    PERM     Indivisibility               {ANY_VALUE},
    PERM     Layout-category              {ANY_VALUE},
    PERM     Layout-object-class          {ANY_VALUE},
    PERM     New-layout-object            {ANY_VALUE},
    PERM     Offset                       {ANY_VALUE},
    PERM     Same-layout-object           {$SameLayoutObject},
    PERM     Separation                   {ANY_VALUE},
    PERM     Synchronization              {ANY_VALUE}
  VOID:
    PERM     Block-alignment              {ANY_VALUE},
    PERM     Concatenation                {ANY_VALUE},
    PERM     Indivisibility               {ANY_STRING --to layout category--
                                          |'page'    --to layout object type--
                                          |'null'},
    PERM     Layout-category              {ANY_VALUE},
    PERM     New-layout-object            {ANY_STRING --to layout category--
                                          |'page'    --to layout object type--
                                          |'null'},
    PERM     Offset                       {ANY_VALUE},
    PERM     Same-layout-object           {REQ  {REQ #logical-object
                                          {<object-id-expr>::=PREC-OBJ(CURR-OBJ);}
                                          | REQ #logical-object {'null'}},
                                           PERM #layout-object
                                          {ANY_STRING --to layout category--
                                          |'page'    --to layout object type--}
                                           },
    PERM     Separation                   {ANY_VALUE},
    PERM     Synchronization              {ANY_VALUE}
 }}
```

7.5.3.3 LStyle3

```
:ANY-LAYOUT-STYLE{
-- This style is used for the constituent constraints CommonText, PageNumber, TableNumber, CurrentInstance,
   CommonNumber and CommonReference. --
PERM    Block-alignment              {ANY_VALUE},
PERM    Concatenation                {ANY_VALUE},
PERM    Offset                       {ANY_VALUE},
PERM    Separation                   {ANY_VALUE}
}
```

7.5.3.4 LStyle4

-- *This style is not used* --

7.5.3.5 LStyle5

```
:ANY-LAYOUT-STYLE      {
-- This style is used for the constituent constraints BodyRaster and BodyGeometric. --

CASE $GLAS OF{
 $COMPLETE:
   PERM    Block-alignment             {ANY_VALUE},
   PERM    Layout-category             {ANY_VALUE},
   PERM    Layout-object-class         {ANY_VALUE},
   PERM    New-layout-object           {ANY_VALUE},
   PERM    Offset                      {ANY_VALUE},
   PERM    Same-layout-object          {$SameLayoutObject},
   PERM    Separation                  {ANY_VALUE},
   PERM    Synchronization             {ANY_VALUE}
 VOID:
   PERM    Block-alignment             {ANY_VALUE},
   PERM    Layout-category             {ANY_VALUE},
   PERM    New-layout-object           {ANY_STRING --to layout category--
                                       |'page'    --to layout object type--
                                       |'null'},
   PERM    Offset                      {ANY_VALUE},
   PERM    Same-layout-object          {REQ  {REQ #logical-object
                                             {<object-id-expr>::=PREC-OBJ(CURR-OBJ);}
                                             | REQ #logical-object {'null'}},
                                        PERM #layout-object
                                             {ANY_STRING --to layout category--
                                             |'page'    --to layout object type--}

                                       },
   PERM    Separation                  {ANY_VALUE},
   PERM    Synchronization             {ANY_VALUE}
}}
```

7.5.3.6 LStyle6

```
:ANY-LAYOUT-STYLE      {
-- This style is used for the constituent constraint FootnoteText. --
CASE $GLAS OF{
 $COMPLETE:
   PERM    Indivisibility              {ANY_VALUE},
   PERM    Block-alignment             {ANY_VALUE},
   PERM    Concatenation               {ANY_VALUE},
   REQ     Layout-category             {$FOOTNOTECATEGORY},
   PERM    Offset                      {ANY_VALUE},
   PERM    Separation                  {ANY_VALUE}
 VOID:
   PERM    Indivisibility              {ANY_STRING --to layout category--
                                       |'page'    --to layout object type--
                                       |'null'},
   PERM    Block-alignment             {ANY_VALUE},
   PERM    Concatenation               {ANY_VALUE},
   REQ     Layout-category             {$FOOTNOTECATEGORY},
   PERM    Offset                      {ANY_VALUE},
   PERM    Separation                  {ANY_VALUE}
}}
```

7.5.3.7 LStyle7

-- This style is not used. --

7.5.3.8 LStyle8

```
:ANY-LAYOUT-STYLE       {
--  This style is used for the constituent constraints CommonRaster and CommonGeometric.  --
PERM     Block-alignment                 {ANY_VALUE},
PERM     Offset                          {ANY_VALUE},
PERM     Separation                      {ANY_VALUE}
}
```

7.5.3.9 LStyle9

```
:ANY-LAYOUT-STYLE       {
--  This style is used for the constituent constraint FootnoteNumber.  --
PERM     Block-alignment                 {ANY_VALUE},
REQ      Layout-category                 {$FOOTNOTECATEGORY},
PERM     Offset                          {ANY_VALUE},
PERM     Separation                      {ANY_VALUE}
}
```

7.5.3.10 LStyle10

```
:ANY-LAYOUT-STYLE       {
--  This style is used for the constituent constraints FootnoteReference and ReferencedContent.  --
CASE $GLAS OF{
 $COMPLETE:
   PERM     Block-alignment              {ANY_VALUE},
   PERM     Concatenation                {ANY_VALUE},
   PERM     Indivisibility               {ANY_VALUE},
   PERM     Layout-category              {ANY_VALUE},
   PERM     Offset                       {ANY_VALUE},
   PERM     Same-layout-object           {$SameLayoutObject},
   PERM     Separation                   {ANY_VALUE}
 VOID:
   PERM     Block-alignment              {ANY_VALUE},
   PERM     Concatenation                {ANY_VALUE},
   PERM     Indivisibility               {ANY_STRING --to layout category--
                                         |'page'    --to layout object type--
                                         |'null'},
   PERM     Layout-category              {ANY_VALUE},
   PERM     Offset                       {ANY_VALUE},
   PERM     Same-layout-object           {REQ  {REQ #logical-object
                                                {<object-id-expr>::=PREC-OBJ(CURR-OBJ);}
                                                | REQ #logical-object {'null'}},
                                          PERM #layout-object
                                                {ANY_STRING --to layout category--
                                                |'page'    --to layout object type--}
                                         },
   PERM     Separation                   {ANY_VALUE}
}}
```

7.5.3.11 LStyle11

```
:ANY-LAYOUT-STYLE       {
--  This style is used for the constituent constraint FootnoteBody.  --
CASE $GLAS OF{
 $COMPLETE:
   PERM     Indivisibility               {ANY_VALUE},
   PERM     Same-layout-object           {$SameLayoutObject},
   PERM     Synchronization              {ANY_VALUE}
 VOID:
   PERM     Indivisibility               {ANY_STRING --to layout category--
                                         |'page'    --to layout object type--
                                         |'null'},
    PERM    Same-layout-object             {REQ  {REQ #logical-object
                                                 {<object-id-expr>::=PREC-OBJ(CURR-OBJ);}
                                                 | REQ #logical-object {'null'}},
                                            PERM #layout-object
                                                 {ANY_STRING --to layout category--
                                                 |'page'     --to layout object type--}
                                           },
    PERM    Synchronization                {ANY_VALUE}
}}
```

7.5.3.12 LStyle12

```
:ANY-LAYOUT-STYLE      {
--  This style is used for the constituent constraint Artwork.  --
CASE $GLAS OF{
 $COMPLETE:
    PERM    Indivisibility                 {ANY_STRING    -- to layout category --
                                           |'page'        -- to layout object type --
                                           |'null'},
    PERM    Layout-object-class            {OBJECT_CLASS_ID_OF(CompositeArtwork)},
    PERM    New-layout-object              {OBJECT_CLASS_ID_OF(CompositeColumnFixed)
                                           |OBJECT_CLASS_ID_OF(CompositeColumnVariable)
                                                          -- to layout object class --
                                           |ANY_STRING    -- to layout category --
                                           |'page'        -- to layout object type --
                                           |'null'},
    PERM    Synchronization                {ANY_VALUE}
 VOID:
    PERM    Indivisibility                 {ANY_STRING --to layout category--
                                           |'page'    --to layout object type--
                                           |'null'},
    PERM    New-layout-object              {ANY_STRING --to layout category--
                                           |'page'    --to layout object type--
                                           |'null'},
    PERM    Synchronization                {ANY_VALUE}
}}
```

7.5.3.13 LStyleT1

```
:ANY-LAYOUT-STYLE      {
--  This style is used for the constituent constraint Form.  --
REQ        Layout-object-class             {OBJECT_CLASS_ID_OF(FormArea)}
}
```

7.5.3.14 LStyleT2

```
ANY-LAYOUT-STYLE       {
--  This style is used for the constituent constraint EntryElement.   --

--  In the case of Form, the following attribute shall be specified.  --
    REQ   Layout-object-class              {OBJECT_CLASS_ID_OF(FormEntryArea)}
 |
--  In the case of Table, both of the following attributes shall be specified. --
    REQ   New-layout-object                {OBJECT_CLASS_ID_OF(Cell)      -- to layout object class --
                                           |ANY_STRING                    -- to layout category --},
    REQ   Indivisibility                   {OBJECT_CLASS_ID_OF(Cell)      -- to layout object class --
                                           |ANY_STRING                    -- to layout category --  }
}
```

7.5.3.15 LStyleT3

```
:ANY-LAYOUT-STYLE      {
--  This style is used for the constituent constraint EntryGroup.  --
REQ        Layout-object-class             {OBJECT_CLASS_ID_OF(EntryGroupArea)}
}
```

7.5.3.16 LStyleT4

```
:ANY-LAYOUT-STYLE     {
-- This style is used for the constituent constraint Table. --
REQ     New-layout-object              {OBJECT_CLASS_ID_OF(TableArea)},
PERM    Indivisibility                 {OBJECT_CLASS_ID_OF(TableArea)
                                                      -- to layout object class --
                                       |ANY_STRING    -- to layout category --
                                       |'page'        -- to layout object type --
                                       | 'null' },
PERM    Same-layout-object             {$SameLayoutObject}
}
```

7.5.3.17 LStyleT5

```
:ANY-LAYOUT-STYLE     {
-- This layout style is used for the constituent constraint Row. --
REQ     New-layout-object              {OBJECT_CLASS_ID_OF(RowArea)},
PERM    Indivisibility                 {OBJECT_CLASS_ID_OF(RowArea)
                                                      -- to layout object class --
                                       |ANY_STRING    -- to layout category --
                                       |'page'        -- to layout object type --
                                       |'null'}
}
```

7.5.3.18 LStyleT6

```
:ANY-LAYOUT-STYLE     {
-- This style is used for the constituent constraint TableComponent. --
REQ     Layout-object-class            {OBJECT_CLASS_ID_OF(SubRowGroup)}
}
```

7.5.3.19 LStyleT7

```
:ANY-LAYOUT-STYLE     {
-- This style is used for the constituent constraint RowComponent. --
REQ     New-layout-object              {OBJECT_CLASS_ID_OF(SubRow)},
PERM    Indivisibility                 {OBJECT_CLASS_ID_OF(SubRow)
                                                      -- to layout object class --
                                       |ANY_STRING    -- to layout category --
                                       |'null'}
}
```

7.5.3.20 LStyleT8

```
:ANY-LAYOUT-STYLE     {
-- This style is used for the constituent constraint Table. --
PERM    Indivisibility                 {OBJECT_CLASS_ID_OF(TableArea)
                                                      -- to layout object class --
                                       |ANY_STRING    -- to layout category --
                                       |'page'        -- to layout object type --
                                       |'null'},
PERM    Same-layout-object             {$SameLayoutObject}
}
```

7.5.3.21 LStyleT9

```
:ANY-LAYOUT-STYLE     {
-- This style is used for the constituent constraints EntryText, EntryRaster and EntryGeometric. --
PERM    Block-alignment                {ANY_VALUE},
PERM    Layout-category                {ANY_VALUE},
PERM    Offset                         {ANY_VALUE}
}
```

7.6 Presentation style constituent constraints

7.6.1 Macro definitions

-- *No macro definitions are applicable to this subclause.* --

7.6.2 Factor constraints

```
FACTOR ANY-PRESENTATION-STYLE

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

7.6.3 Constituent constraints

7.6.3.1 PStyle1

```
:ANY-PRESENTATION-STYLE  {
-- This style is used for the constituent constraints BodyText, Number, FootnoteNumber, FootnoteReference,
   FootnoteText, EntryText, ReferencedContent, GenericBlock and SpecificBlock. --
PERM    Presentation-attributes {
  PERM #character-attributes {
    PERM #alignment                           {ANY_VALUE},
    PERM #character-fonts                     {ANY_VALUE},
    PERM #character-orientation               {ANY_VALUE},
    PERM #character-path                      {ANY_VALUE},
    PERM #character-spacing                   {ANY_VALUE},
    PERM #code-extension-announcers           {$CDEXTEN},
    PERM #first-line-offset                   {ANY_VALUE},
    PERM #graphic-character-sets              {$PERMIT-GRCHAR},
    PERM #graphic-character-subrepertoire     {ANY_VALUE},
    PERM #graphic-rendition                   {$GRAPHICRENDITIONS},
    PERM #indentation                         {ANY_VALUE},
    PERM #itemization                         {ANY_VALUE},
    PERM #kerning-offset                      {ANY_VALUE},
    PERM #line-layout-table                   {ANY_VALUE},
    PERM #line-progression                    {ANY_VALUE},
    PERM #line-spacing                        {ANY_VALUE},
    PERM #orphan-size                         {ANY_VALUE},
    PERM #pairwise-kerning                    {ANY_VALUE},
    PERM #proportional-line-spacing           {ANY_VALUE},
    PERM #widow-size                          {ANY_VALUE},
    PERM #initial-offset                      {ANY_VALUE},
    PERM #formatting-indicator                {ANY_VALUE}
    }
}}
```

7.6.3.2 PStyle2

```
:ANY-PRESENTATION-STYLE      {
-- This style is used for the constituent constraints BodyGeometric, CommonGeometric, EntryGeometric, GenericBlock
   and SpecificBlock. --
PERM    Presentation-attributes           {
    PERM #geometric-graphics-attributes {ANY_VALUE}}
}
```

7.6.3.3 PStyle3

```
:ANY-PRESENTATION-STYLE      {
-- This style is used for the constituent constraints BodyRaster, CommonRaster, EntryRaster, GenericBlock and
   SpecificBlock. --
PERM       Presentation-attributes {
  PERM #raster-graphics-attributes {
    PERM #pel-path                       {ANY_VALUE},
    PERM #line-progression               {ANY_VALUE},
    PERM #pel-spacing                    {ANY_VALUE},
    PERM #spacing-ratio                  {ANY_VALUE},
    PERM #clipping                       {ANY_VALUE},
    PERM #image-dimensions               {ANY_VALUE}
    }
}}
```

7.6.3.4 PStyle4

```
:ANY-PRESENTATION-STYLE  {
-- This style is used for the constituent constraints CommonText, PageNumber, TableNumber, CommonReference,
   CurrentInstance and SpecificBlock. --
PERM       Presentation-attributes {
  PERM #character-attributes {
    PERM #alignment                              {ANY_VALUE},
    PERM #character-fonts                        {ANY_VALUE},
    PERM #character-orientation                  {ANY_VALUE},
    PERM #character-path                         {ANY_VALUE},
    PERM #character-spacing                      {ANY_VALUE},
    PERM #code-extension-announcers              {$CDEXTEN},
    PERM #first-line-offset                      {ANY_VALUE},
    PERM #graphic-character-sets                 {$PERMIT-GRCHAR},
    PERM #graphic-character-subrepertoire        {ANY_VALUE},
    PERM #graphic-rendition                      {$GRAPHICRENDITIONS},
    PERM #indentation                            {ANY_VALUE},
    PERM #itemization                            {ANY_VALUE},
    PERM #kerning-offset                         {ANY_VALUE},
    PERM #line-layout-table                      {ANY_VALUE},
    PERM #line-progression                       {ANY_VALUE},
    PERM #line-spacing                           {ANY_VALUE},
    PERM #pairwise-kerning                       {ANY_VALUE},
    PERM #proportional-line-spacing              {ANY_VALUE},
    PERM #initial-offset                         {ANY_VALUE},
    PERM #formatting-indicator                   {ANY_VALUE}
    }
}}
```

7.7 Content portion constituent constraints

7.7.1 Macro definitions

```
DEFINE(T6,        "ASN.1 {2 8 3 7 0}")

DEFINE(T41D,      "ASN.1 {2 8 3 7 1}")

DEFINE(T42D,      "ASN.1 {2 8 3 7 2}")

DEFINE(Bitmap,    "ASN.1 {2 8 3 7 3}")
```

7.7.2 Factor constraints

```
FACTOR ANY-CONTENT   {
CASE $DAC OF {
 $FDA :
        REQ       Content-identifier-layout       {ANY_VALUE}

 $PDA :
        REQ       Content-identifier-logical      {ANY_VALUE}
                  -- This attribute is specified, if the content portion is associated with a basic logical object or a basic
                     logical object class. --
        |REQ      Content-identifier-layout      {ANY_VALUE}
                  -- This attribute is specified, if the content portion is associated with a basic layout object class. --
  $FPDA :
        REQ       Content-identifier-layout      {ANY_VALUE},
        REQ       Content-identifier-logical     {ANY_VALUE}
                  -- Both attributes are specified, if the content portion is associated with a basic logical object and a
                     basic layout object. --
        |REQ      Content-identifier-layout      {ANY_VALUE}
                  -- This attribute is specified, if the content portion is associated with a basic layout object class. --
        |REQ      Content-identifier-logical     {ANY_VALUE}
                  -- This attribute is specified, if the content portion is associated with a basic logical object class. --
        },
PERM    Alternative-representation               {ANY_STRING}
}
```

7.7.3 Constituent constraints

7.7.3.1 Character-content-portion

```
:ANY-CONTENT {
PERM    Type-of-coding                           {ASN.1{2 8 3 6 0}},
PERM    Content-information                      {CHARACTER
    {

--  Shared Control Functions --
    #CR
    | #GCC                                       {ANY_VALUE}
    | #IGS                                       {ANY_VALUE}
    | #LF
    | #PLD
    | #PLU
    | #SCS                                       {ANY_VALUE}
    | #SGR                                       {$GRAPHICRENDITIONS}
    | #SHS                                       {ANY_VALUE}
    | #SLS                                       {ANY_VALUE}
    | #SRS                                       {ANY_VALUE}
    | #STAB                                      {ANY_VALUE}
    | #SUB
    | #SVS                                       {ANY_VALUE}
    | #VPB                                       {ANY_VALUE}
    | #VPR                                       {ANY_VALUE}

--  Layout Control Functions --
    | #HPB                                       {ANY_VALUE}
    | #HPR                                       {ANY_VALUE}
    | #JFY                                       {0}
    | #SACS                                      {ANY_VALUE}
    | #SRCS                                      {ANY_VALUE}
    | #SSW                                       {ANY_VALUE}

--  Logical Control Functions --
    | #BPH
    | #NBH
    | #PTX                                       {ANY_VALUE}

--  Delimiter Functions --
    | #SOS
    | #ST

--  Space --
    | #SP

--  Code extension control functions  --
    | #LS0
    | #LS1R
    | #LS2R
    | #LS3R
```

```
     | #SS2
     | #SS3
     | #ESC {$DEG-CORE-G0}
     | #ESC {$DEG-646-G0}
     | #ESC {$DEG-ANY-G1}
     | #ESC {$DEG-ANY-G2}
     | #ESC {$DEG-ANY-G3}
     | #ESC {$DEG-EMPTY-G1}
     }...}
}
```

7.7.3.2 Raster-graphics-content-portion

```
:ANY-CONTENT {
PERM     Type-of-coding                        {$T6|$T41D|$T42D|$Bitmap},
PERM     Coding-attributes                     {
 PERM    #raster-graphics-coding-attributes          {
     PERM #compression                         {ANY_VALUE},
     PERM #number-of-lines                     {>0},
     REQ  #number-of-pels-per-line             {>=0}
     }
 },
PERM     Content-information                   {RASTER}
}
```

7.7.3.3 Geometric-graphics-content-portion

```
:ANY-CONTENT {
PERM     Type-of-coding                        {ASN.1 {2 8 3 8 0}},
PERM     Content-information                   {GEOMETRIC}
}
```

8 Interchange format

For conformance to this Recommendation, the interchange format class A shall be used when applying Open Document interchange Format (ODIF).

NOTE – Interchange format SDIF applies to the international Standardized Profile only.

8.1. Interchange format class A

8.1.1 Interchange format

The value of the document profile attribute “interchange format” for this interchange format is “if-a”. This form of ODIF is defined in CCITT Rec. T.415 | ISO 8613-5.

8.1.2 Document application profile object identifier

The value for the document profile attribute “document application profile” for this interchange format is represented by the following object identifier:

ASN.1 { 2 8 4 0 36 0 }

8.1.3 Encoding of application comments

The encoding of the attribute “application comments” is defined in this encoding as an octet string as specified in CCITT Rec. T.415 | ISO 8613-5. This document application profile requires that the encoding within that octet string be in accordance with the ASN.1 syntax specified in the following module definition:

```
FOD_DAPSpecification
DEFINITIONS    ::= BEGIN
EXPORTS  Appl-Comm-Encoding;

Appl-Comm-Encoding      ::= SEQUENCE {
       constraint-namE      [0] IMPLICIT PrintableString OPTIONAL,
       external-data        [1] IMPLICIT OCTET STRING OPTIONAL }
END
```

8.1.4 Data lengths

The maximum length of data values of the type OCTET STRING, as defined in CCITT Rec. X.208 | ISOÍndice 8824, in data streams which may be encoded in accordance with this document application profile is 32 767 octets. If it is required to encode an octet string of greater length than this, constructed type encoding shall be used. That is, data values greater than 32 767 in length shall be split into a sequence of strings shorter than 32 767, each of which is encoded using a primitive type.

Annex A

Amendments and technical corrigenda

(This annex forms an integral part of this Recommendation.)

A.1 Amendments

A.1.1 Amendments to the base standard

The amendments applicable to this Recommendation include text to be included in CCITT Rec. T.411 | ISO 8613-1 as the following annexes:

- *Annex E*: Use of ISOÍndice 10021 (MOTIS) to interchange documents conforming to CCITT Recs. T.410-Series | ISO 8613. Published as First extension to the CCITT Recs. T.410-Series (1988), Addendum II | ISO 8613-1 (1989), Amendment 1;
- *Annex F*: Document Application Profile proforma and notation. Published as CCITT Recs. T.411 – Annex F (1991) | ISO 8613-1 (1989), Addendum 1;
- *Annex G*: Conformance testing methodology. Published as ISO 8613-1 (1989), Amendment 2;
- *Annex H*: Recording of documents conforming to CCITT Recs. T.410-Series | ISO 8613 on flexible disk cartridges conforming to ISO 9293 – Currently Draft Amendment to ISO 8613-1 (1989), Annex H.

This Recommendation does not include the following features of the amendment:

- Addendum on security. Published as First extension to the CCITT Recs. T.410-Series (1988), Addendum IV | ISO 8613 (1989), Addendum 4;
- Addendum on styles. Published as First extension to the CCITT Recs. T.410-Series (1988), Addendum IV | ISO 8613 (1989), Addendum 6;
- Addendum on alternative representation. Published as First extension to the CCITT Recs. T.410-Series (1988), Addendum III | ISO 8613 (1989), Addendum 3;
- Addendum on colour. Published as Revision of the CCITT Recs. T.410-Series contained in COM VIII – R30-E Addendum 2 | ISO 8613 (1989), Amendment 2;
- Addendum on tiled raster graphics. Published as First extension to the CCITT Recs. T.410-Series (1988) Addendum I | ISO 8613 (1989), Addendum 1;
- Addendum on streams. Published as ISO 8613 (1989), Amendment 5.

In addition, this amendment addresses the inclusion of the CCITT Recs. T.410-Series (1988) | ISO 8613 (1988) Technical Corrigenda 1, 2 and 3.

A.1.2 Proposed changes to standards due to defects

This amendment addresses the inclusion of the CCITT Rec. T.410-Series (1988) | ISO 8613 (1988) Technical Corrigenda 1, 2 and 3.

A.2 Technical corrigenda

A.2.1 Technical corrigenda to this Recommendation

There is no technical corrigendum specific to this Recommendation.

A.3 Versions of the ODA Recommendation

The version of the ODA defined in A.1 is known as “version 1.1”. The associated date is 1992-01-01. The version of the ODA including all features of the amendments, addenda and technical corrigenda mentioned in A.1 is known as “version 1.2”. The associated date is 1992-07-01.

The values for the attribute “ODA version” for this document application profile is:

- *Standard or Recommendation*:

  CCITT Recs. T.410-Series (1988) | ISO 8613 (1989); version 1.1

- *Publication date*:

  1992-01-01

Annex B

Recommended practices

(This annex does not form an integral part of this Recommendation.)

B.1 Transfer methods for ODA

B.1.1 Conveyance of ODA over CCITT X.400 (1984)

This recommended practice describes how ODA body parts are to be encoded for transmission over a CCITT X.400 (1984) service.

An ODA body part is encoded as OdaBodyPart in the definition given below:

```
OdaBodyPart ::= SEQUENCE { OdaBodyPartParameters, OdaData }
OdaBodyPartParameters ::= SET {
    document-application-profile
        [0] IMPLICIT OBJECT IDENTIFER,
    document-architecture-class
        [1] IMPLICIT INTEGER {
            formatted (0),
            processable (1),
            formatted-processable (2) }
OdaData ::=   SEQUENCE OF Interchange-Data-Element
```

NOTE – It is recommended to transfer an ODA document as a single body part with tag 12:

```
Oda [12] IMPLICIT OCTETSTRING
```

The content of the octet string is encoded as OdaBodyPart, defined above. However, this is out of the scope of this Recommendation.

B.1.2 Conveyance of ODA over FTAM

This recommended practice describes the FTAM Document Type to be used for minimal storage and transfer capabilities of ODA data streams. It is recognized that enhanced capabilities may at some point be added.

When using FTAM to transfer an ODA file, the FTAM-3, “ISO FTAM Unstructured Binary”, document type shall be specified.

However, since files that do not contain ODA data streams can have the same document type, it is left up to the user of application programs that remotely access files using FTAM to know that a given file contains an ODA data stream.

B.1.3 Conveyance of ODA over DTAM

This recommended practice provides for information concerning the interchange of ODA based documents with DTAM protocols.

DTAM (Document Transfer and Manipulation) is defined in the T.430-Series Recommendations, and is, like ODA, an integral part of the T.400-Series Recommendations.

The T.520-Series Recommendations contain Communication Application Profiles (CAP). Recommendation T.522 describes the Communication Application Profile BT1 for document bulk transfer. Recommendation T.522 is applicable for the Open Document Format Profile (FOD) published in this Recommendation.

NOTE – The use of BT1 within the end-to-end oriented Telematic Services Telefax 4 and Teletex is described in 7.1/T.561 and 7.1/T.562.

B.1.4 Conveyance of ODA over flexible disks

The recommended practice for interchanging ODA documents between systems by the exchange of magnetically recorded Flexible Disk Cartridges is given in Annex H of CCITT Rec. T.411 | ISO 8613-1. This annex provides for recording each ODA document as a separate file as defined in ISO 9293.

B.2 Font reference

The recommended practice for specifying a font reference is to base it on ISO 9541.

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

The minimum font properties and values from ISO 9541 that are to be specified in a Font-Attribute-Set are those specified by the following document application profile notation.

```
Font-Attribute-Set      {

PERM     Font-Name                      {ANY_VALUE},
PERM     Standard-Version               {ANY_VALUE},
PERM     Data-source                    {ANY_VALUE},
PERM     Design-source                  {ANY_VALUE},
PERM     Font-Family-Name               {ANY_VALUE},

PERM     Posture                        {'upright' | 'italic-forward'},
PERM     Weight                         {'light' | 'medium' | 'bold'},
PERM     Proportionate-Width            {ANY_VALUE},
PERM     Glyph-Complement       {
         PERM #Included-Glyph-Collections
                                        {ANY_VALUE},
         PERM #Excluded-Glyph-Collections
                                        {ANY_VALUE},
         PERM #Included-Glyphs          {ANY_VALUE},
         PERM #Excluded-Glyphs          {ANY_VALUE}
                                },
PERM     Design-Size                    {ANY_VALUE},
PERM     Min-Size               {
         PERM #Numerator                {100 .. 1200},
         PERM #Denominator              {1}
                                },
PERM     Max-Size               {
         PERM #Numerator                {100 .. 1200},
         PERM #Denominator              {1}
                                },
    -- BMU [Units equivalent to range of 6..72 point sizes]--

PERM     Design-Group           {
         PERM #Class                    {ANY_VALUE},
         PERM #Subclass                 {ANY_VALUE},
         PERM #Specific-Group           {ANY_VALUE}
                },
PERM     Structure                      {ANY_VALUE},
PERM     Writing-Modes          {
         MUL                    {
         REQ  #Writing-Mode-Name
                                        {ANY_VALUE},
         PERM #Nominal-Escapement-Direction
                                        {ANY_VALUE},
PERM #Escapement-Class
                                        {ANY_VALUE},
PERM #Average-Escapement-X
                                        {ANY_VALUE},
PERM #Average-Escapement-Y
                                        {ANY_VALUE}
                             }
                   }
      }
```

B.3 ISO 8632 (CGM) constraints for this document application profile

It is a recommended practice that geometric graphics content information contain only those elements listed in this portion of the document, in addition to the constraints imposed by CCITT Rec. T.418 | ISO 8613-8. It is believed that this subset of the CGM is sufficiently implemented to enable interworking of geometric graphics for application conforming to this document application profile.

Where an element has parameters, constraints on the values that are part of this recommended practice are given. The "--" symbol indicates that there is no constraint given as part of this recommended practice.

Requirements in ISO 8632 and CCITT Rec. T.418 | ISO 8613-8 concerning mandatory elements, parameters shall be fulfilled.

B.3.1 Delimiter elements

No-Op	An arbitrary sequence of *n* octets. Where *n* = 0, 1, .., 32 767. The sequence of zero or more octets is for padding purposes.
Begin Metafile	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets.
End Metafile	
Begin Picture	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets.
Begin Picture Body	
End Picture	

B.3.2 Metafile descriptor elements

Metafile Version	1
Metafile Description	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets. The METAFILE DESCRIPTION string parameter will be used to include the sub-string "ITU-T Rec. PM-36" to label the content information as conforming to this Recommendation. In addition, generators of content are encouraged to append a sub-string that identifies the company and product that produced the CGM.
VDC Type	--
Integer Precision	16, 32
Real Precision	(0, 9, 23), (1, 16, 16)
Index Precision	16
Colour Precision	8, 16
Colour Index Precision	8, 16

Maximum Colour Index	0..63
Colour Value Extent	--
Metafile Element List	--
Metafile Defaults Replacement	--
Font List	All fonts referenced in the metafile shall be defined. Font referencing in FONT LISTS using ISO 9541 names is preferred, but font names may be specified using proprietary font names.
Character Set List	All character sets referenced in the metafile shall be defined in CHARACTER SET LIST. Permissible character sets are the same as for character content architecture.
Character Coding Announcer	--

B.3.3 Picture descriptor elements

Scaling Mode	The Scale Factor parameter of SCALING MODE element is always a 32-bit floating point value, even when the REAL PRECISION has selected fixed point for other real numbers. It is not apparent in ISO 8632 what the precision of this floating point value is when fixed point has been selected. Its precision shall be (0, 9, 23).
Colour Selection Mode	--
Line Width Specification Mode	--
Marker Size Specification Mode	--
Edge Width Specification Mode	--
VDC Extent	--
Background Colour	--

B.3.4 Control elements

VDC Integer Precision	16, 32
VDC Real Precision	(0, 9, 23), (1, 16, 16)
Auxiliary Colour	--
Transparency	Transparent
Clip Rectangle	--
Clip Indicator	--

B.3.5 Graphical primitive elements

Polyline	The minimum support for the length of point lists is 1024 vertices.
Disjoint Polyline	The minimum support for the length of point lists is 1024 vertices.
Polymarker	The minimum support for the length of point lists is 1024 vertices.
Text	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets. Format effector control characters are disallowed in the string parameter.

Restricted Text	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets. Format effector control characters are disallowed in the string parameter.
Append Text	Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets. Format effector control characters are disallowed in the string parameter.
Polygon	The minimum support for the length of point lists is 1024 vertices.
Polygon Set	The minimum support for the length of point lists is 1024 vertices.
Cell Array	The minimum support for the length of colour lists parameter for the CELL ARRAY element is 1 048 576. This will support a 1024 × 1024 image.
Rectangle	--
Circle	--
Circular Arc 3 Point	--
Circular Arc 3 Point Close	--
Circular Arc Centre	--
Circular Arc Centre Close	--
Ellipse	--
Elliptical Arc	--
Elliptical Arc Close	--

B.3.6 Attribute elements

Line Bundle Index	--
Line Type	Negative values are prohibited.
Line Width	--
Line Colour	--
Marker Bundle Index	--
Marker Type	Negative values are prohibited.
Marker Size	--
Marker Colour	--
Text Bundle Index	--
Text Font Index	All fonts referenced (indexed by TEXT FONT INDEX) in the metafile shall be defined in FONT LIST either in presentation parameters of CCITT Recs. T.410-Series | ISO 8613 or in ISO 8632.
Text Precision	--
Character Expansion Factor	--

Character Spacing	--
Text Colour	--
Character Height	--
Character Orientation	--
Text Path	--
Text Alignment	--
Character Set Index	All character sets referenced in the metafile (indexed by CHARACTER SET INDEX) shall be defined in CHARACTER SET LIST. The only character sets which may be designated in G0 are ISO 646 IRV or versions of ISO 646. Other character sets shall be designated in G1, G2 or G3.
Alternate Character Set Index	All character sets referenced in the metafile (indexed by ALTERNATE CHARACTER SET INDEX) shall be defined in CHARACTER SET LIST.
Fill Bundle Index	--
Interior Style	--
Fill Colour	--
Hatch Index	Negative values are prohibited.
Pattern Index	1 .. 8
Edge Bundle Index	--
Edge Type	Negative values are prohibited.
Edge Width	--
Edge Colour	--
Edge Visibility	--
Fill Reference Point	--
Pattern Table	The PATTERN TABLE element has an unspecified effect when it appears in a picture subsequent to any graphical primitives. The PATTERN TABLE element shall appear prior to any graphical primitive element to assure that interpreting systems without dynamic pattern update can render the intended effect. The minimum support for the length of the Colour Array parameter for the PATTERN TABLE element is 2048. This will support 8 patterns of 16 × 16, 2 patterns of 32 × 32 or 1 pattern of 32 × 64. All indexes which are used in the metafile shall be defined.
Pattern Size	--
Colour Table Specification	The COLOUR TABLE element has an unspecified effect when it appears in a picture subsequent to any graphical primitives. The COLOUR TABLE element shall appear prior to any graphical primitive elements to assure that interpreting systems without dynamic colour update can render the intended effect. The minimum support for the length of the Colour List parameter in the COLOUR TABLE element is 63. This will support a 64 (0..63) entry colour table. All indexes which are used in the metafile shall be defined.
Aspect Source Flags	--

B.3.7 External elements

Message

The presentation of message string may not be appropriate for all applications. No requirement for formatted presentation of the message string has been placed on the Interpreter. Only the No Action flag needs to be supported. Support for string lengths up to 254.

Application Data

Support will be provided for strings with a length up to 254 octets, except for data records which will support strings with a length up to 32 767 octets.

Annex C

Bibliography

(This annex does not form an integral part of this Recommendation.)

- CCITT Recommendation T.50 (1992), *International Reference Alphabet*.

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

- ITU-T Recommendation T.52 (1993), *Non-Latin coded character sets for telematic services*.

- CCITT Recommendation T.502 (1991), *Document Application Profile PM-11 for the Interchange of Character Content Documents in Processable and Formatted Forms*.

- CCITT Recommendation T.505 (1991), *Document Application Profile PM-26 for the Interchange of Mixed Content Documents in Processable and Formatted Forms*.

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

- ISO/IEC 9070:1991, *Information technology – SGML support facilities – Registration procedures for public text owner identifiers.*

- ISO/IEC TR 9573:1988, *Information technology – SGML support facilities – Techniques for using SGML*.

- ISO/IEC 10021:1990, *Information technology – Text Communication – Message-Oriented Text Interchange System*.